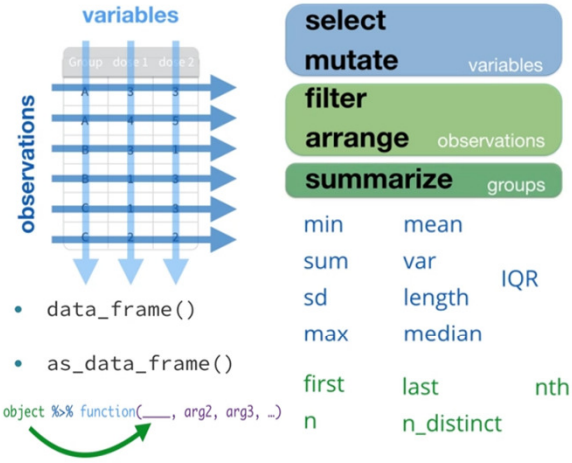
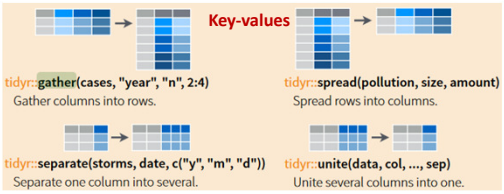


select variables
mutate
filter observations
arrange
summarize groups

min mean
sum var IQR
sd length
max median
first last nth
n n_distinct

- data_frame()
- as_data_frame()

```
object %>% function(____, arg2, arg3, …)
```



Use these helpers with select (),
e.g. select(iris, starts_with("Sepal"))

contains(match) **num_range**(prefix, range) :, e.g. mpg:cyl
ends_with(match) **one_of**(…) -, e.g, -Species
matches(match) **starts_with**(match)

You can also combine `group_by()` with `mutate()`. When you mutate grouped data, `mutate()` will calculate the new variables independently for each group. This is particularly useful when `mutate()` uses the `rank()` function, that calculates within-group rankings. `rank()` takes a group of values and calculates the rank of each value within the group, e.g.

summary function

COUNTS
dplyr::**n()** - number of values/rows
dplyr::**n_distinct()** - # of uniques
sum(!is.na()) - # of non-NA's

LOCATION
mean() - mean, also **mean(!is.na())**
median() - median

LOGICALS
mean() - Proportion of TRUE's
sum() - # of TRUE's

POSITION/ORDER
dplyr::**first()** - first value
dplyr::last() - last value
dplyr::nth() - value in nth location of vector

factors

```
# y is a factor
> y <- factor(c(5, 6, 7, 6))
> y
[1] 5 6 7 6
Levels: 5 6 7

> unclass(y)
[1] 1 2 3 2
attr(,"levels")
[1] "5" "6" "7"
```

```
> as.character(y)
[1] "5" "6" "7" "6"
> as.numeric(y)
[1] 1 2 3 2
> as.numeric(as.character(y))
[1] 5 6 7 6
```

- left_join()
- right_join()
- inner_join()
- full_join()
- semi_join()
- anti_join()
- bind_rows()
- bind_cols()

Mutating joins
- left_join(), right_join(), inner_join(), full_join()

Filtering joins
- semi_join(), anti_join()

Set operations
- union(), intersect(), setdiff()

Comparisons
- setequal()

```
> bind_rows(Beatles = band1, Stones = band2, .id = "band")
    band    name   surname
1 Beatles    John    Lennon
2 Beatles    Paul McCartney
3 Beatles  George  Harrison
4 Beatles   Ringo     Starr
5  Stones    Mick    Jagger
6  Stones   Keith  Richards
7  Stones Charlie     Watts
```

```
> left_join(members, plays, by = c("member" = "name"))
  member    band  plays
1   Mick  Stones  <NA>
2   John Beatles Guitar
3   Paul Beatles  Bass
```

```
> left_join(names2, plays2, by = c("name", "surname"))
```

vector of key column name(s)

```
> names
   name    band
1  Mick  Stones
2  John Beatles
3  Paul Beatles
```

```
> plays
   name  plays
1  John Guitar
2  Paul   Bass
3 Keith Guitar
```

```
> semi_join(names, plays, by = "name")
   name    band
1  John Beatles
2  Paul Beatles
```

```
> names
   name    band
1  Mick  Stones
2  John Beatles
3  Paul Beatles
```

```
> plays
   name  plays
1  John Guitar
2  Paul   Bass
3 Keith Guitar
```

```
> right_join(names, plays, by = "name")
   name    band  plays
1  John Beatles Guitar
2  Paul Beatles   Bass
3 Keith    <NA> Guitar
```

```
> names
   name    band
1  Mick  Stones
2  John Beatles
3  Paul Beatles
```

```
> more_names
   name    band
1 Keith  Stones
2  Mick  Stones
3  John Beatles
4  John Beatles
```

```
> setdiff(names, more_names)
   name    band
1  Paul Beatles
```