

## Wrangling

verb	action
arrange	<b>arrange</b> the <i>rows</i> according to some <i>column</i>
filter	<b>filter</b> out or obtain a subset of the <i>rows</i>
select	<b>select</b> a subset of <i>columns</i>
mutate	<b>mutate</b> or create a <i>column</i>
summarize	calculate a numerical <b>summary</b> of a <i>column</i>
group_by	<b>group</b> the <i>rows</i> by a specified <i>column</i>

## Reshaping

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K



country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

**pivot\_longer**(data, cols, names\_to = "name",  
values\_to = "value", values\_drop\_na = FALSE)

"Lengthen" data by collapsing several columns  
into two. Column names move to a new  
names\_to column and values to a new values\_to  
column.

```
pivot_longer(table4a, cols = 2:3, names_to = "year",  
values_to = "cases")
```

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T



country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

**pivot\_wider**(data, names\_from = "name",  
values\_from = "value")

The inverse of pivot\_longer(). "Widen" data by  
expanding two columns into several. One column  
provides the new column names, the other the  
values.

```
pivot_wider(table2, names_from = type,  
values_from = count)
```

## Joining Data

The most common mutating joins are:

- **left\_join()** Keeps *all* observations from the left, but discards any observations in the right that do not have a match in the left.<sup>1</sup>
- **inner\_join()** Keeps *only* the observations from the left with a match in the right.

- `full_join()` Keeps *all* observations from the left *and* the right. (This is less common than `left_join()` and `inner_join()`).
- `semi_join()` Discards any observations in the left table that *do not* have a match in the right table. If there are multiple matches of right cases to a left case, it keeps just one copy of the left case.
- `anti_join()` Discards any observations in the left table that *do* have a match in the right table.

## Factors

- functions for changing the **order** of factor levels
  - `fct_relevel()` = *manually* reorder levels
  - `fct_reorder()` = reorder levels according to values of another *variable*
  - `fct_infreq()` = order levels from highest to lowest frequency
  - `fct_rev()` = reverse the current order
- functions for changing the **labels** or values of factor levels
  - `fct_recode()` = *manually* change levels
  - `fct_lump()` = *group together* least common levels

## Strings

function	arguments	returns
<code>str_replace()</code>	<code>x, pattern, replacement</code>	a modified string
<code>str_replace_all()</code>	<code>x, pattern, replacement</code>	a modified string
<code>str_to_lower()</code>	<code>x</code>	a modified string
<code>str_sub()</code>	<code>x, start, end</code>	a modified string
<code>str_length()</code>	<code>x</code>	a number
<code>str_detect()</code>	<code>x, pattern</code>	TRUE/FALSE