**Tidy data** is a way to organize tabular data in a consistent data structure across packages. A table is tidy if:

- Each variable is in its own column
- Each observation, or case, is in its own row

### Tibbles

**AN ENHANCED DATA FRAME**

Tibbles are a table format provided by the `tibble` package. They inherit the data frame class, but have improved behaviors:

- **Subset** a new tibble with `[`, a vector with `[[` and `$`.
- **No partial matching** when subsetting columns.
- **Display** concise views of the data on one screen.

#### Options

- `options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf)` Control default display settings.

#### View() or glimpse()

View the entire data set.

#### Construct a Tibble

`tibble()` function constructs by columns.

`tibble(x = 1:3, y = c("a", "b", "c"))`

###split_cells(x, ...)` Convert a data frame to a tibble.

`enframe(x, name = "name", value = "value")` Convert a named vector to a tibble. Also `deframe()`.

`is_tibble(x)` Test whether `x` is a tibble.

### Tables

#### Reshape Data

- **Pivot data to reorganize values into a new layout.**

```
<table>
<thead>
<tr>
<th>country</th>
<th>year</th>
<th>cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1999</td>
<td>0.7K</td>
</tr>
<tr>
<td>B</td>
<td>2000</td>
<td>2K</td>
</tr>
<tr>
<td>C</td>
<td>2000</td>
<td>172M</td>
</tr>
</tbody>
</table>
```

#### Expand Tables

- **Create new combinations of variables or identify implicit missing values (combinations of variables not present in the data).**

#### Handle Missing Values

- **Drop or replace explicit missing values (NA).**

### Functions

- `is_tibble()`: Check if an object is a tibble.
- `tibble()`: Create a new tibble.
- `as_tibble()`: Convert a data frame to a tibble.
- `enframe()`: Convert a named vector to a tibble.
- `deframe()`: Convert a tibble to a data frame.
- `is_tibble()`: Check if an object is a tibble.
- `tibble()`: Create a new tibble.
- `as_tibble()`: Convert a data frame to a tibble.
- `enframe()`: Convert a named vector to a tibble.
- `deframe()`: Convert a tibble to a data frame.
- `is_tibble()`: Check if an object is a tibble.
- `tibble()`: Create a new tibble.
- `as_tibble()`: Convert a data frame to a tibble.
- `enframe()`: Convert a named vector to a tibble.
- `deframe()`: Convert a tibble to a data frame.
- `is_tibble()`: Check if an object is a tibble.
- `tibble()`: Create a new tibble.
- `as_tibble()`: Convert a data frame to a tibble.
- `enframe()`: Convert a named vector to a tibble.
- `deframe()`: Convert a tibble to a data frame.
- `is_tibble()`: Check if an object is a tibble.
A nested data frame stores individual tables as a list-column of data frames within a larger organizing data frame. List-columns can also be lists of vectors or lists of varying data types.

**CREATE NESTED DATA**

```r
tibble::tribble( ... ) Makes list-columns when needed. tribble(max, -max, -seq, 3, 1:3, 4, 1:4, 5, 1:5)
tibble(max = c(3, 4, 5), seq = list(1:3, 1:4, 1:5))
tibble(x, name = "name", value = "value") Converts multi-level list to a tibble with list-cols.
enframe(list(3 = 1:3, 4 = 1:4, 5 = 1:5), ‘max’, ‘seq’)
```

**OUTPUT LIST-COLUMNS FROM OTHER FUNCTIONS**

```r
mutate(), transmute(), and summarise() will output list-columns if they return a list.
```