Data is made up of a variable and an observation.

A **variable** is a quantity, quality, or property that you can measure. An **observation** is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object).

An observation will contain several values, each associated with a different variable. This is also sometimes referred to as a data point. A **value** is the state of a variable when you measure it. The value of a variable typically changes from observation to observation.

Tabular form

Tabular data is a set of values, each associated with a variable and an observation. Tabular data is tidy if each value is placed in its own 'cell', with each variable in its own column, and each observation in its own row.

Let's examine the tuberculosis (TB) case notifications data set. This data is in tidy tabular **long** form (meaning there are more observations than variables)

```
tb
```

```
## # A tibble: 47,866 x 6
##    country     iso3   year count sex   age_group
##    <chr>       <chr> <dbl> <dbl> <fct> <fct>
##  1 Afghanistan AFG    1997    10 M     15-24
##  2 Afghanistan AFG    1998   129 M     15-24
##  3 Afghanistan AFG    1999    55 M     15-24
##  4 Afghanistan AFG    2000   228 M     15-24
##  5 Afghanistan AFG    2001   379 M     15-24
##  6 Afghanistan AFG    2002   476 M     15-24
##  7 Afghanistan AFG    2003   511 M     15-24
##  8 Afghanistan AFG    2004   537 M     15-24
##  9 Afghanistan AFG    2005   606 M     15-24
## 10 Afghanistan AFG    2006   837 M     15-24
## # ... with 47,856 more rows
```

The difference between 'messy' and 'tidy'

Messy data is messy in its own way. You can make unique solutions, but then another data set comes along, and you have to again make a unique solution.

The original form of the TB case notifactions data is shown in the following code chunk:

```
tb_messy <- read_csv("data/TB_notifications.csv")
tb_messy
```

```
## # A tibble: 7,891 x 23
##    country iso3   year new_sp_m04 new_sp_m514 new_sp_m014 new_sp_m1524
##    <chr>   <chr> <dbl>      <dbl>       <dbl>       <dbl>        <dbl>
##  1 Afghan~ AFG    1980         NA          NA          NA           NA
##  2 Afghan~ AFG    1981         NA          NA          NA           NA
##  3 Afghan~ AFG    1982         NA          NA          NA           NA
##  4 Afghan~ AFG    1983         NA          NA          NA           NA
##  5 Afghan~ AFG    1984         NA          NA          NA           NA
##  6 Afghan~ AFG    1985         NA          NA          NA           NA
##  7 Afghan~ AFG    1986         NA          NA          NA           NA
##  8 Afghan~ AFG    1987         NA          NA          NA           NA
##  9 Afghan~ AFG    1988         NA          NA          NA           NA
## 10 Afghan~ AFG    1989         NA          NA          NA           NA
## # ... with 7,881 more rows, and 16 more variables: new_sp_m2534 <dbl>,
## #   new_sp_m3544 <dbl>, new_sp_m4554 <dbl>, new_sp_m5564 <dbl>,
## #   new_sp_m65 <dbl>, new_sp_mu <dbl>, new_sp_f04 <dbl>, new_sp_f514 <db
l>,
## #   new_sp_f014 <dbl>, new_sp_f1524 <dbl>, new_sp_f2534 <dbl>,
## #   new_sp_f3544 <dbl>, new_sp_f4554 <dbl>, new_sp_f5564 <dbl>,
## #   new_sp_f65 <dbl>, new_sp_fu <dbl>
```

Why keep it 'tidy'?

In a messy format, it is unclear what some of the columns are measuring, and the column names confound several variables: age group, sex and the technique used to measure TB case. In this form, the data is in **wide** tabular form, because there are multiple columns containing different columns.

Tidy data can be thought of as Lego bricks. Once you have this form, you can put it together in so many different ways, to make different analyses.

In the following steps, you'll learn how to change your data from **wide** to **long** (and the other way around), and how to **separate** columns into multiple variables.

Verbs for tidying

Throughout this course, you will use the **`tidyverse`** collection of packages to perform data tidying, wrangling and visualisation.

To perform operations on your data, you will use functions from these packages that are named after verbs. As you work your way through the course and learn the R language, you will learn how to compose these verbs together to form 'data analysis' sentences.

The verbs for tidying data are:

- **gather:** take a data set from wide to long

- **spread:** go from long to wide
- **separate:** split variables in one column to multiple columns.

Over the next steps, you will learn what these verbs do, and how you can use them to transform the TB data from messy to a tidy form.