Tidy Data with R

- The **tidyr** package by Hadley Wickham is designed to help you tidy your data.
- **tidyr** contains four functions that alter the layout of tabular data sets, while preserving the values and relationships contained in the data sets.
- The two most important functions in tidyr are `gather()` and `spread()`.
- Each relies on the idea of a key value pair.

Tidy Data with R

- A key value pair is a simple way to record information.
- A pair contains two parts: a **key** that explains what the information describes, and a **value** that contains the actual information.

```
Password: 0123456789
```

- *0123456789* is the **value**, and it is associated with the **key** *Password*.

Tidy Data with R

- You could decompose table1 into a group of key value pairs, but it would cease to be a useful data set because you no longer know which values belong to the same observation (next slides).
- In tidy data, each cell will contain a value and each column name will contain a key, but this doesnt need to be the case for untidy data.

Tidy Data with R

```
Country: Afghanistan
Country: Brazil
Country: China
Year: 1999
Year: 2000
Year: 2001
```

Tidy Data with R

```
Population:   19987071
Population:   20595360
Population:  172006362
Population:  174504898
Population: 1272915272
Population: 1280428583
Cases:    745
Cases:   2666
Cases:  37737
Cases:  80488
Cases: 212258
Cases: 213766
```

Tidy Data with R

```
## Source: local data frame [12 x 4]
##
##        country year        key      value
## 1  Afghanistan 1999      cases        745
## 2  Afghanistan 1999 population   19987071
## 3  Afghanistan 2000      cases       2666
## 4  Afghanistan 2000 population   20595360
## 5       Brazil 1999      cases      37737
## 6       Brazil 1999 population  172006362
## 7       Brazil 2000      cases      80488
## 8       Brazil 2000 population  174504898
## 9        China 1999      cases     212258
## 10       China 1999 population 1272915272
## 11       China 2000      cases     213766
## 12       China 2000 population 1280428583
```

Tidy Data with R

spead()

- In table2, the key column contains only keys (and not just because the column is labelled key).
- Conveniently, the value column contains the values associated with those keys.
- You can use the `spread()` function to tidy this layout.

Tidy Data with R

`spread()`

- `spread()` turns a pair of key:value columns into a set of tidy columns.
- To use `spread()`, pass it the name of a data frame, then the name of the key column in the data frame, and then the name of the value column.
- Pass the column names as they are; do not use quotes.
- To tidy table2, you would pass `spread()` the key column and then the value column.

Tidy Data with R

```
## Source: local data frame [12 x 4]
##
##        country year        key     value
## 1  Afghanistan 1999      cases       745
## 2  Afghanistan 1999 population  19987071
## 3  Afghanistan 2000      cases      2666
## 4  Afghanistan 2000 population  20595360
.....
```

Tidy Data with R

```
library(tidyr)
spread(table2, key, value)

## Source: local data frame [6 x 4]
##
##       country year  cases population
## 1 Afghanistan 1999    745   19987071
## 2 Afghanistan 2000   2666   20595360
## 3      Brazil 1999  37737  172006362
## 4      Brazil 2000  80488  174504898
## 5       China 1999 212258 1272915272
## 6       China 2000 213766 1280428583
```

Tidy Data with R

- `spread()` returns a copy of your data set that has had the **key** and **value** columns removed.
- In their place, `spread()` adds a new column for each unique value of the key column (i.e. new columns: cases and populations).
- These unique values will form the column names of the new columns.
- `spread()` distributes the cells of the former value column across the cells of the new columns and truncates any non-key, non-value columns in a way that prevents duplication.

Tidy Data with R

- `spread()` distributes a pair of key:value columns into a field of cells. The unique values of the key column become the column names of the field of cells.
- You can see that `spread()` maintains each of the relationships expressed in the original data set. The output contains the four original variables, country, year, population, and cases.
- And the values of these variables are grouped according to the orginal observations, but now the layout of these relationships is tidy.

Tidy Data with R

`spread()` takes three optional arguments in addition to data, key, and value:

- `fill`
- `convert`
- `drop`

Tidy Data with R

`fill`

- If the tidy structure creates combinations of variables that do not exist in the original data set, `spread()` will place an `NA` in the resulting cells.
- (`NA` is `R`s missing value symbol).
- You can change this behaviour by passing `fill` an alternative value to use.

Tidy Data with R

`convert`

- If a value column contains multiple types of data, its elements will be saved as a single type, usually character strings.
- As a result, the new columns created by `spread()` will also contain character strings.
- If you set `convert = TRUE`, `spread()` will run `type.convert()` on each new column, which will convert strings to *doubles (numerics)*, *integers*, *logicals*, *complexes*, or *factors*.

Tidy Data with R

drop

- The drop argument controls how spread() handles factors in the key column.
- If you set drop = FALSE, spread will keep factor levels that do not appear in the key column, filling in the missing combinations with the value of fill.