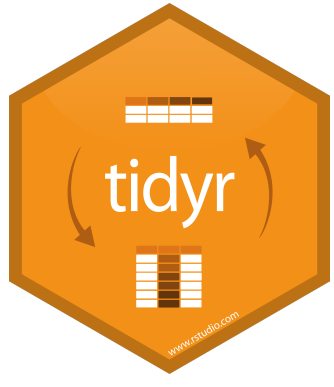
Tidy Messy Data with



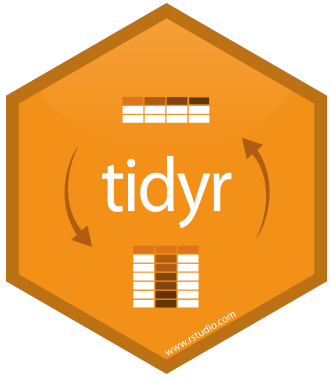
"Happy families are all alike, every
unhappy family is unhappy in its own way."

– Leo Tolstoy

"Tidy datasets are all alike, but every messy dataset is messy in its own way."

– Hadley Wickham

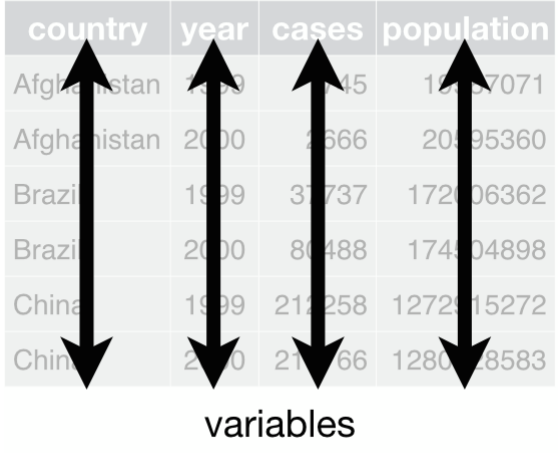
Introduction to tidyr

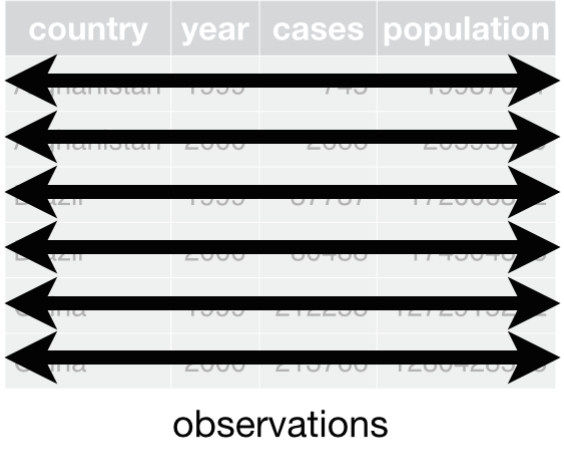


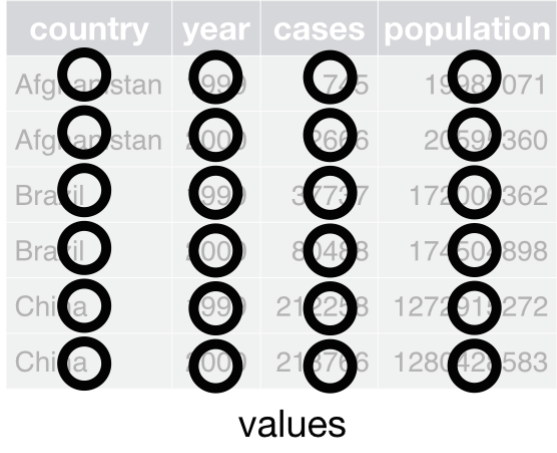
A package that reshapes the layout of tabular data.

Tidy Data Principle

1. Each **variable** is in its own **column**
2. Each **observation** is in its own **row**
3. Each **value** is in its own **cell**

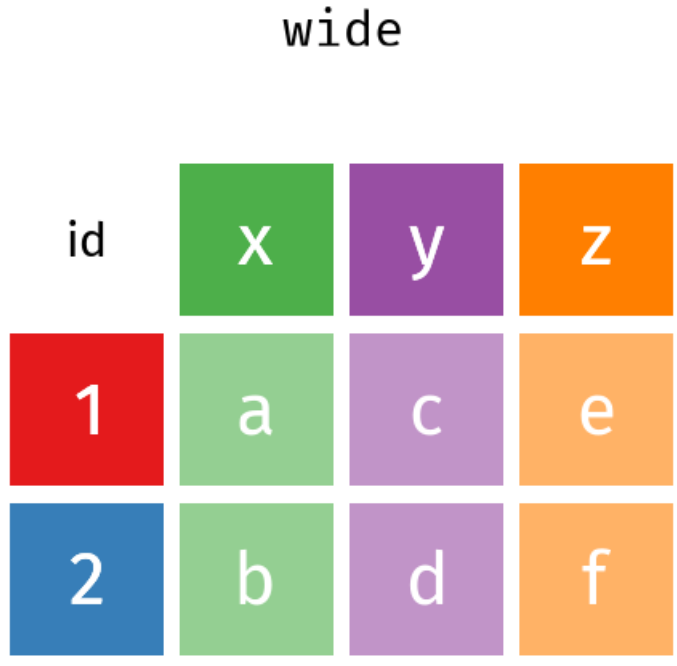






Tidyr - Functions

Function	Description
pivot_wider()	widen the columns
pivot_longer()	lengthen the rows



Pivot in Tidyr

```
pivot_longer(data,
             cols = "columns",
             names_to = "name",
             values_to = "value")
```

```
pivot_wider(data,
            names_from = name,
            values_from = value)
```

Here,

- **data** - name of data frame
- **cols** - select columns to lengthen
- **names_to** - store column names
- **values_to** - store values
- **names_from** - widen column names
- **values_from** - widen value names

Creating a Data Frame

```r
# data frame with 4 cols, 3 rows

data <- data.frame(id = c(1, 2),
                   x = c("a", "b"),
                   y = c("c", "d"),
                   z =  c("e", "f"))
```

id	x	y	z
1	a	c	e
2	b	d	f

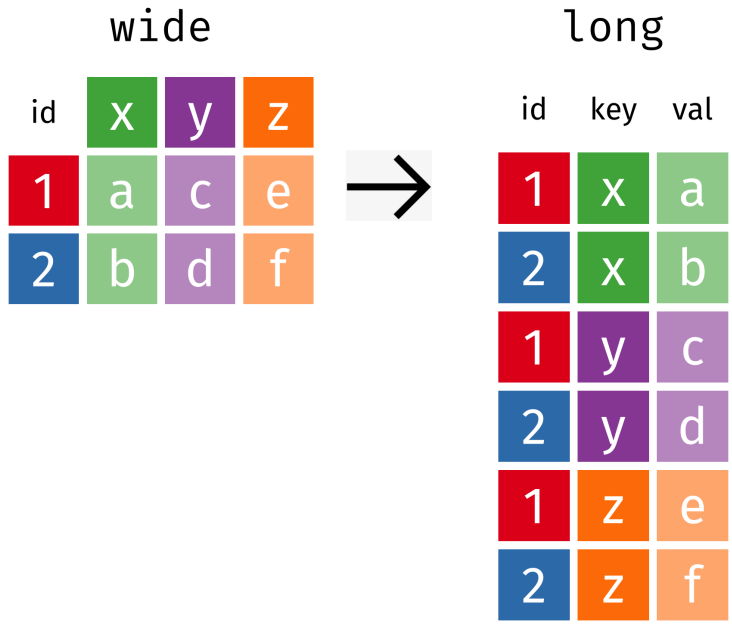
Pivoting - Wide to Long

```r
library(tidyr)

long <- pivot_longer(data,
  cols = 2:4,
  names_to = "key",
  values_to = "val"
)
```

Here,

- cols - select 2nd to 4th colums
- names_to - store names to "key"
- values_to - store values to "val"

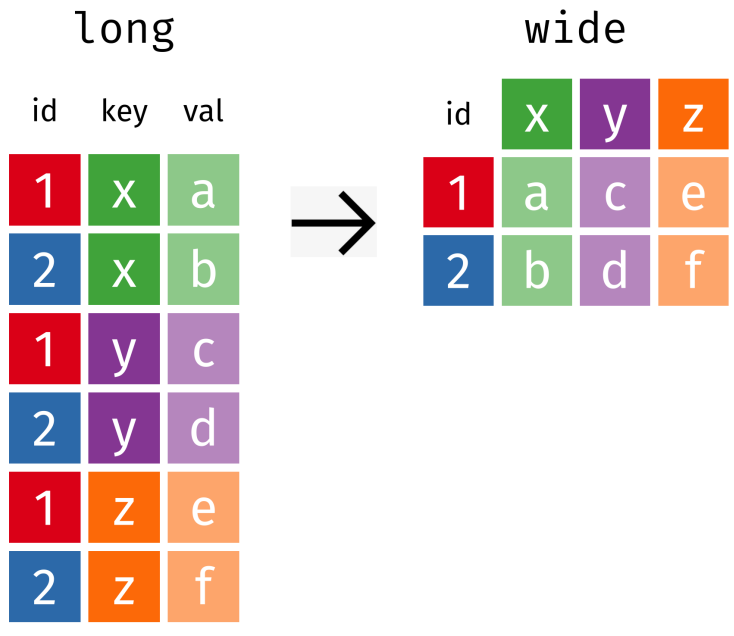


Pivoting - Long to Wide

```
wide <- pivot_wider(long,
  names_from = key,
  values_from = val
)
```

Here,

- **names_from** - widen "key" cols
- **values_from** - widen "val" cols



Untidy Data Format

table4a

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

Tidy Data Format

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

table4a - Reshaping Data

```
table_a <- table4a %>%
  pivot_longer(cols = 2:3,
               names_to = "year",
               values_to = "cases")
```

Here,

- column **2nd to 3rd** is selected
- lengthen variable to "**year**" and
- lengthen values to "**cases**"

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

02:00

Your Turn - 01

Answer:

```
table_b <- table4b %>%
  pivot_longer(cols = 2:3,
               names_to = "year",
               values_to = "population")
```

table4b - Reshape data into tidy format

country	year	population
Afghanistan	1999	19987071
Afghanistan	2000	20595360
Brazil	1999	172006362
Brazil	2000	174504898
China	1999	1272915272
China	2000	1280428583

Merging Data

```
df <- merge(table_a, table_b)
```

Here,

- **merge** - combines two data frame
- **table_a** - country, year, cases
- **table_b** - country, year, population

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

02:00

Your Turn - 02

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Find average rate of cases in year 2000.

Hints:

1. Filter value by year 2000
2. Remove year
3. Create variable - **rate** by **cases/population**
4. Summarise i.e average by **rate** using **mean()**

Filter by Value

country	year	cases	population
Afghanistan	2000	2666	20595360
Brazil	2000	80488	174504898
China	2000	213766	1280428583

```
filter(df, year == 2000)
```

Select by Variables

country	year	cases	population
Afghanistan	2000	2666	20595360
Brazil	2000	80488	174504898
China	2000	213766	1280428583

```
filter(df, year == 2000)
select(df, -year)
```

Mutate - Add New Variable

country	cases	population	rate
Afghanistan	2666	20595360	0.00013
Brazil	80488	174504898	0.00046
China	213766	1280428583	0.00017

```
filter(df, year == 2000)
select(df, -year)
mutate(df, rate = cases /population)
```

Summarize Data

country	cases	population	rate
Afghanistan	2666	20595360	0.00013
Brazil	80488	174504898	0.00046
China	213766	1280428583	0.00017

avg
0.00025

```
filter(df, year == 2000)
select(df, -year)
mutate(df, rate = cases/population)
summarise(df, avg = mean(rate))
```

Average rate of cases in year 2000

country	cases	population	rate
Afghanistan	2666	20595360	0.00013
Brazil	80488	174504898	0.00046
China	213766	1280428583	0.00017

avg
0.00025

```
df %>%
filter(year == 2000) %>%
select(-year) %>%
mutate(rate = cases/population) %>%
summarise(avg = mean(rate))
```