Getting / cleaning data 2

Longer and wider data

pivot_longer / pivot_wider

There are two functions from the `tidyr` package (another member of the tidyverse) that you can use to change between wide and long data: `pivot_longer` and `pivot_wider`.

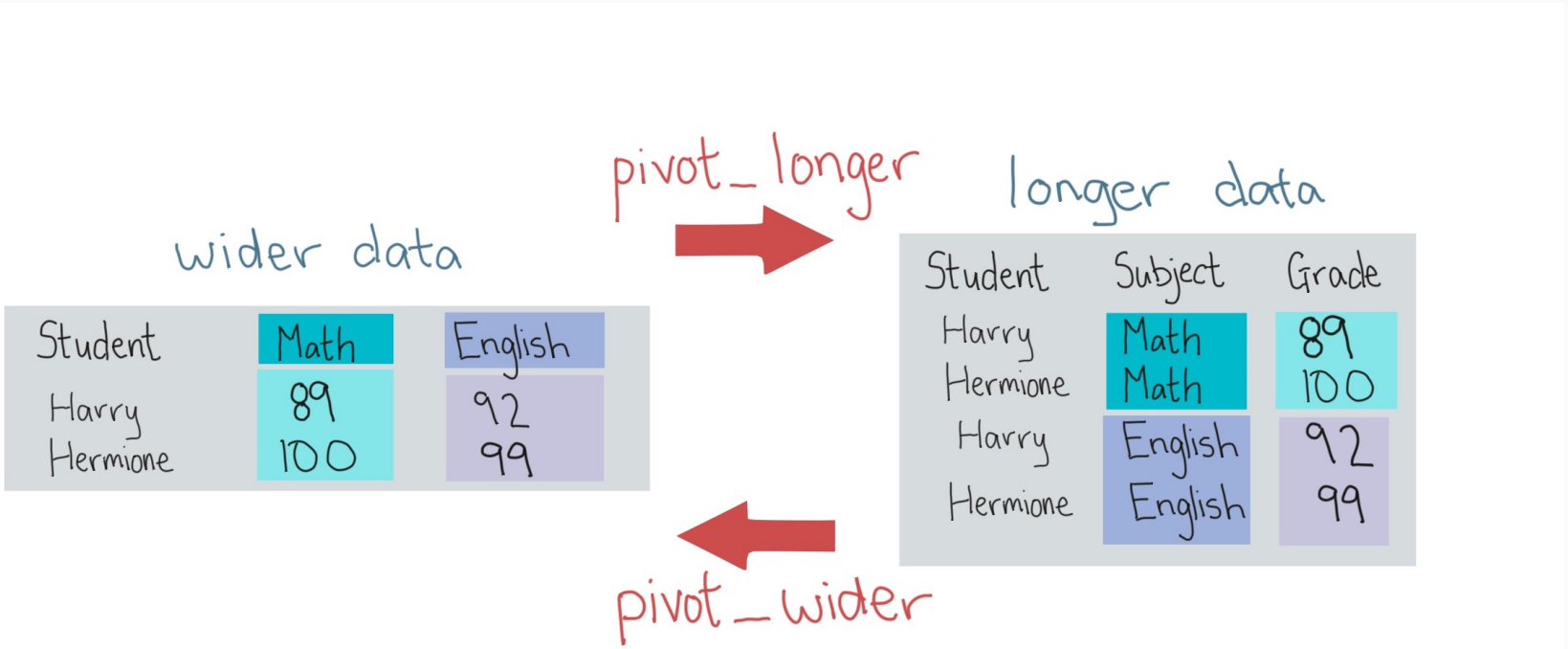
These are somewhat new, and they replace the older `gather` and `spread` functions.

pivot_longer / pivot_wider

Here is a description of these two functions:

- `pivot_longer`: Takes several columns and pivots them down into two columns. One of the new columns contains the former column names and the other contains the former cell values.
- `pivot_wider`: Takes two columns and pivots them up into multiple columns. Column names for the new columns will come from one column and the cell values from the other.

pivot_longer / pivot_wider



pivot_longer / pivot_wider

The following examples show the effects of making a dataset longer or wider.

Here is some example wide data:

```
hogwarts_wide
```

```
## # A tibble: 2 x 4
##   student   math english science
##   <chr>    <dbl>   <dbl>   <dbl>
## 1 Harry       89      92      93
## 2 Hermione   100      99      98
```

pivot_longer / pivot_wider

In the `hogwarts_wide` dataset, there are separate columns for three different courses (`math`, `english`, and `science`). Each cell gives the value for a certain stock on a certain day.

```
hogwarts_wide
```

```
## # A tibble: 2 x 4
##   student   math english science
##   <chr>    <dbl>   <dbl>   <dbl>
## 1 Harry       89      92      93
## 2 Hermione   100      99      98
```

This data isn't "tidy", because the identify of the course (`math`, `english`, or `science`) is a variable, and you'll probably want to include it as a variable in modeling.

pivot_longer / pivot_wider

If you want to convert the dataframe to have all stock values in a single column, you can use `pivot_longer` to convert wide data to long data:

```
library("tidyr")
hogwarts_long <- pivot_longer(data = hogwarts_wide,
                              cols = math:science,
                              names_to = "subject",
                              values_to = "grade")
```

pivot_longer / pivot_wider

In this "longer" dataframe, there is now one column that gives the identify of the course (`subject`) and another column that gives the grade a student got for that course (`grade`):

```
hogwarts_long
```

```
## # A tibble: 6 x 3
##   student  subject grade
##   <chr>    <chr>   <dbl>
## 1 Harry    math       89
## 2 Harry    english    92
## 3 Harry    science    93
## 4 Hermione math      100
## 5 Hermione english    99
## 6 Hermione science    98
```

pivot_longer / pivot_wider

The format for a pivots_longer call is:

```
## Generic code
new_df <- pivot_longer(old_df,
                       cols = [name(s) of the columns you want
                               to make longer],
                       names_to = [name of new column to store
                                   the old column names],
                       values_to = [name of new column to store
                                    the old values])
```

pivot_longer / pivot_wider

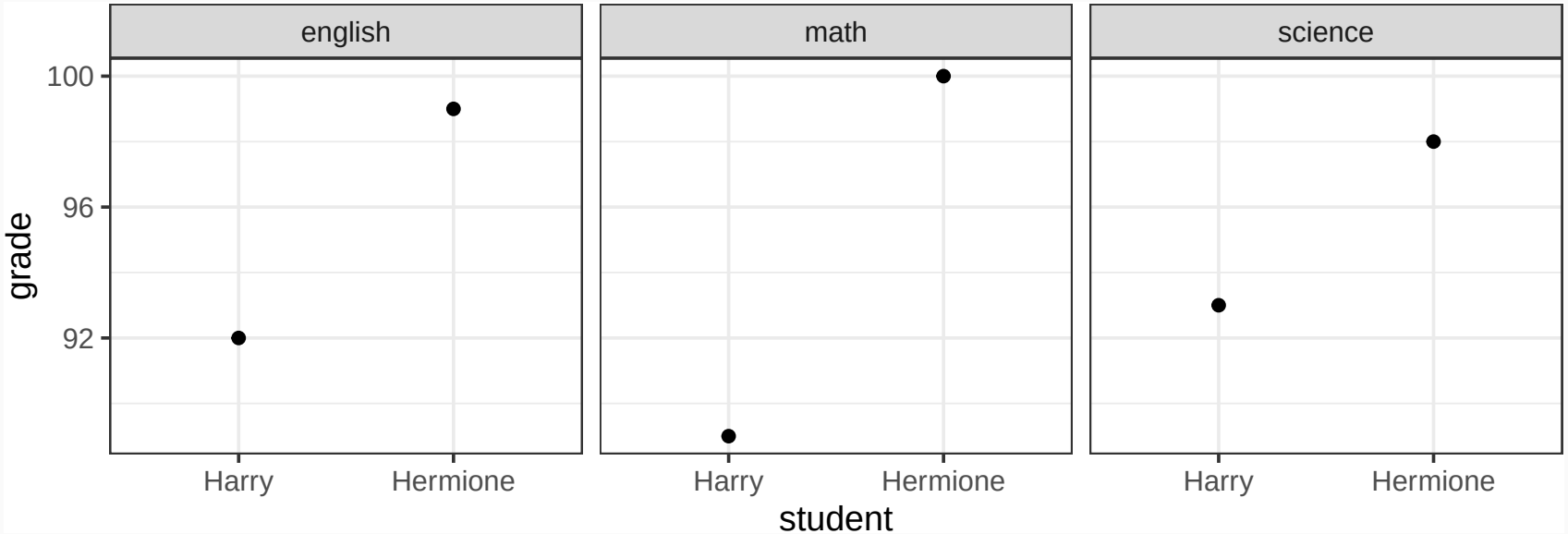
Three important notes:

- Everything is pivoted into one of two columns—one column with the old column names, and one column with the old cell values
- With the `names_to` and `values_to` arguments, you are just providing column names for the two columns that everything's pivoted into. When you are pivoting from "wide" to "long", you get to pick these names.
- If there is a column you don't want to include in the pivot (`date` in the example), use `-` to exclude it in the `cols` argument.

pivot_longer / pivot_wider

Notice how easy it is, now that the data is gathered, to use `subject` for aesthetics of faceting in a `ggplot2` call:

```
ggplot(hogwarts_long, aes(x = student, y = grade)) +
  geom_point() +
  facet_wrap(~ subject) +
  theme_bw()
```



pivot_longer / pivot_wider

If you have data in a "longer" format and would like to make it "wider", you can use `pivot_wider` to do that:

```
hogwarts <- pivot_wider(hogwarts_long,
                        names_from = "subject",
                        values_from = "grade")
```

Notice that this reverses the action of `pivot_longer`.

Further reading

Chapters 12 and 13 of "R for Data Science" are an excellent supplemental resource if you'd like extra practice on tidy data, pivoting, and joining different datasets.

Note: At this time "R for Data Science" uses the `gather` and `spread` instead of `pivot_*`. These are older functions, you should use `pivot_*`.