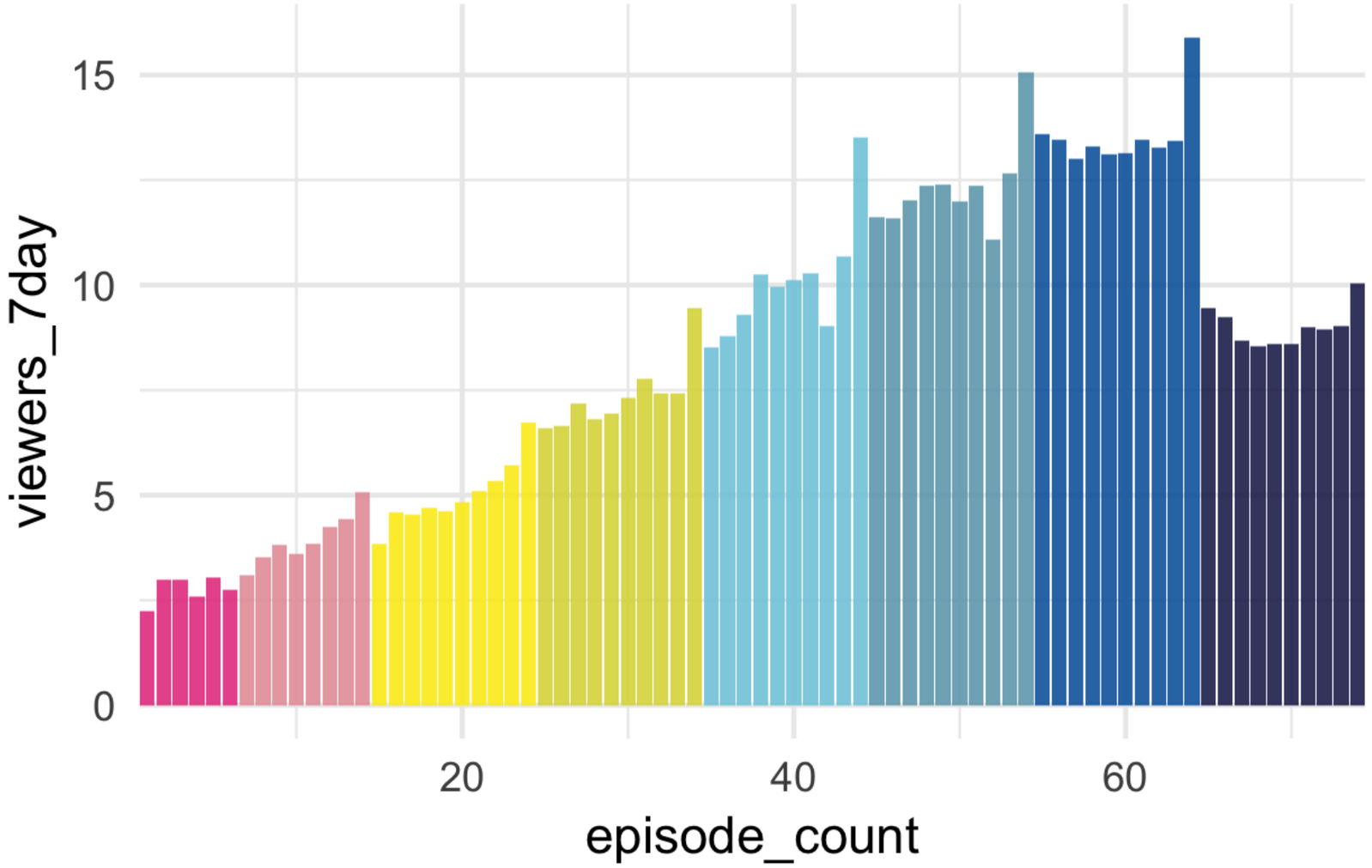
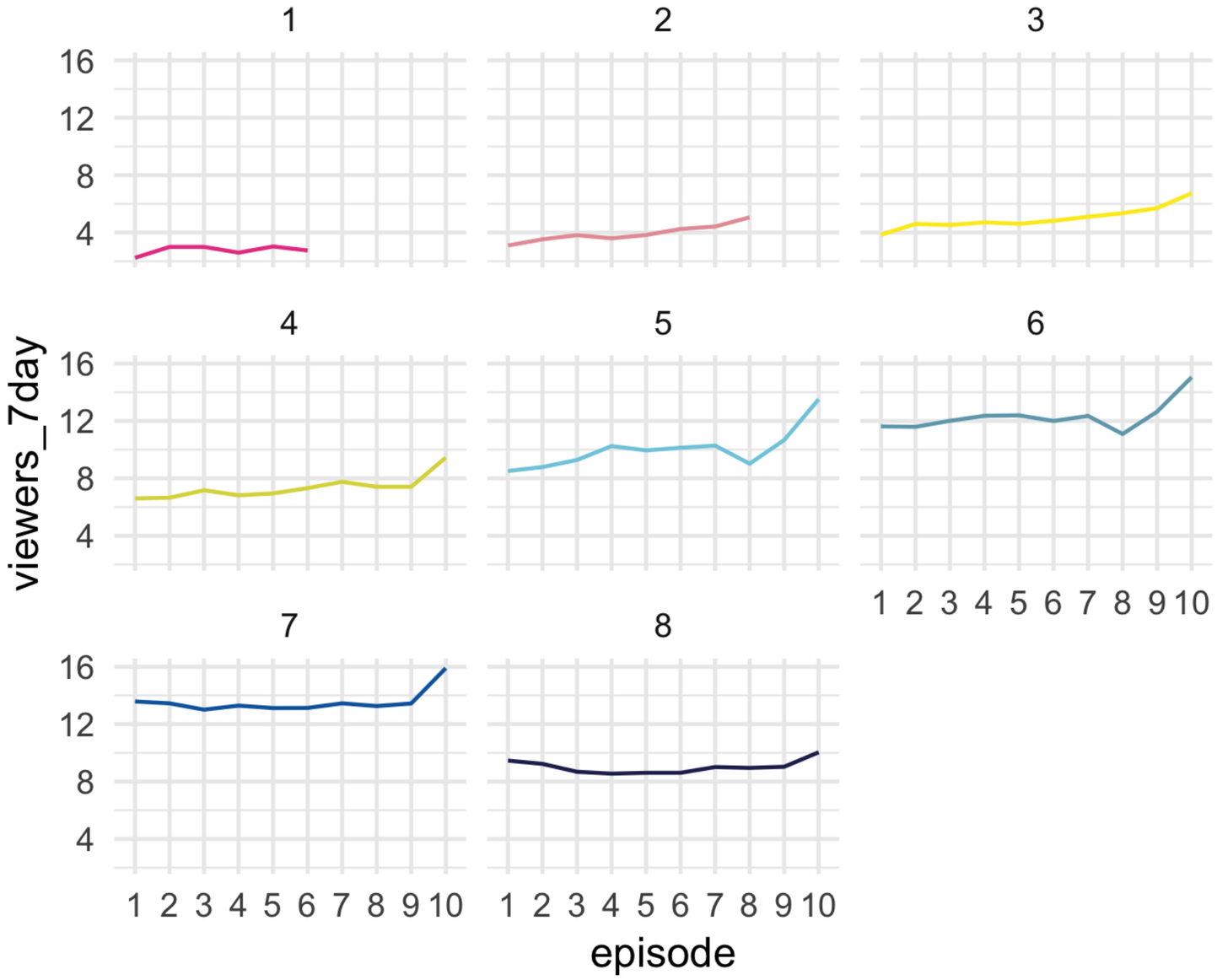
Introduction to Tidy Data

WORKING WITH DATA IN THE TIDYVERSE

Alison Hill
Professor & Data Scientist


15
10
5
0
viewers_7day
20
40
60
episode_count


1
2
3
4
5
6
7
8
16
12
8
4
viewers_7day
1 2 3 4 5 6 7 8 9 10
episode

The Great British Bake Off Series 8

Prue Leith
@PrueLeith

Follow

I am so sorry to the fans of the show for my mistake this morning, I am in a different time zone and mortified by my error #GBBO.

4:53 AM - 31 Oct 2017

tame data ≠ tidy data

Tame but un-tidy

```
juniors_untidy
```

```
# A tibble: 4 x 4
  baker  cinnamon_1 cardamom_2 nutmeg_3
  <chr>       <int>      <int>    <int>
1 Emma            1          0        1
2 Harry           1          1        1
3 Ruby            1          0        1
4 Zainab          0         NA        0
```

Tidy data

```
juniors_tidy
```

```
# A tibble: 12 x 4
   baker  spice    order correct
   <chr>  <chr>    <int>   <int>
 1 Emma   cinnamon     1       1
 2 Harry  cinnamon     1       1
 3 Ruby   cinnamon     1       1
 4 Zainab cinnamon     1       0
 5 Emma   cardamom     2       0
 6 Harry  cardamom     2       1
 7 Ruby   cardamom     2       0
 8 Zainab cardamom     2      NA
 9 Emma   nutmeg       3       1
10 Harry  nutmeg       3       1
11 Ruby   nutmeg       3       1
12 Zainab nutmeg       3       0
```

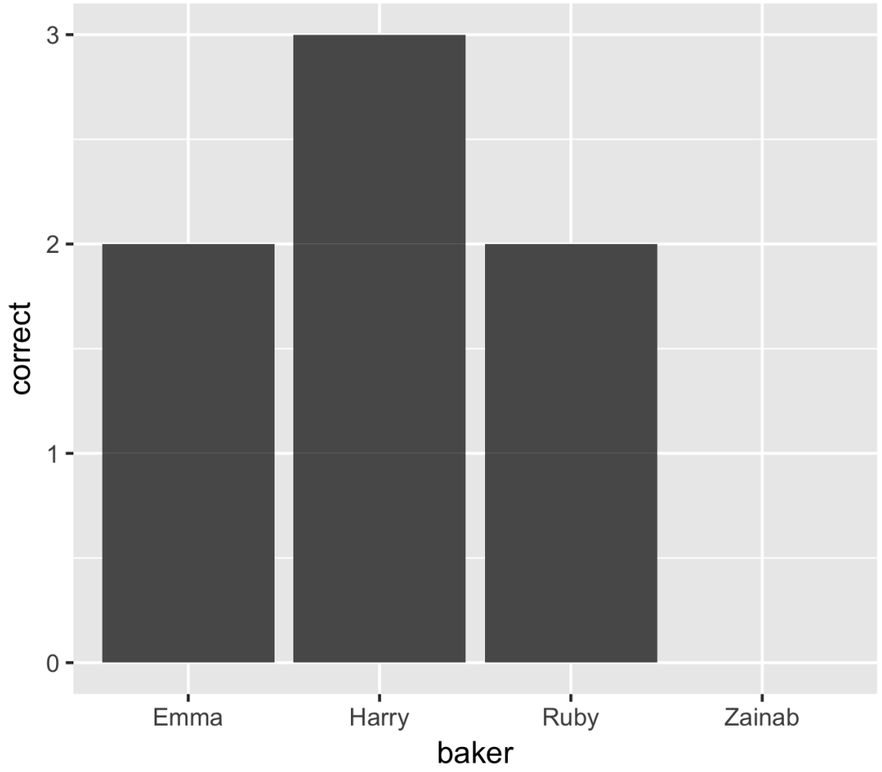
Who won? Count it!

```
juniors_tidy %>%
  count(baker, wt = correct)
```

```
# A tibble: 4 x 2
  baker      n
  <chr>  <int>
1 Emma       2
2 Harry      3
3 Ruby       2
4 Zainab     0
```

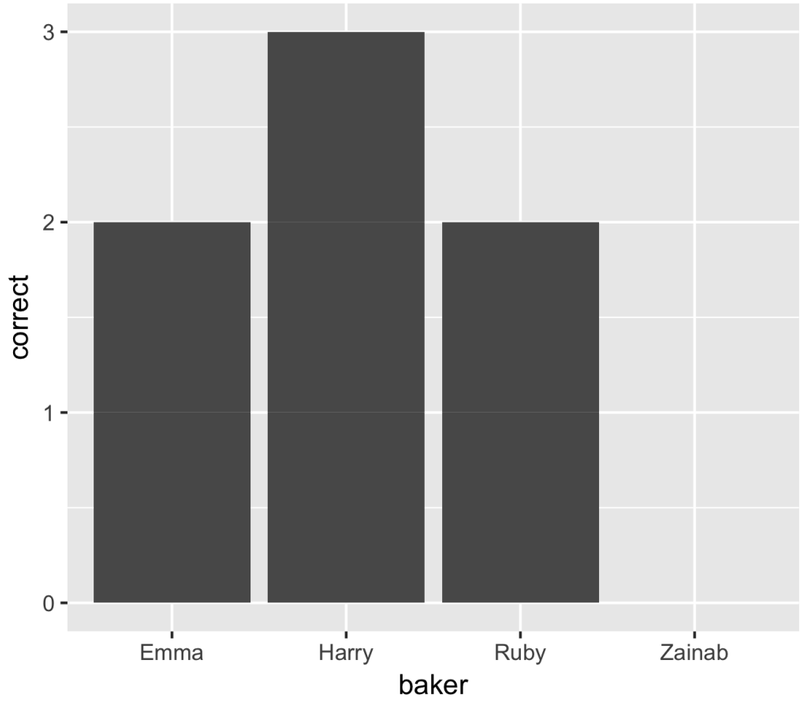
Who won? Plot it!

```
ggplot(juniors_tidy, aes(baker, correct)) +
  geom_col()
```



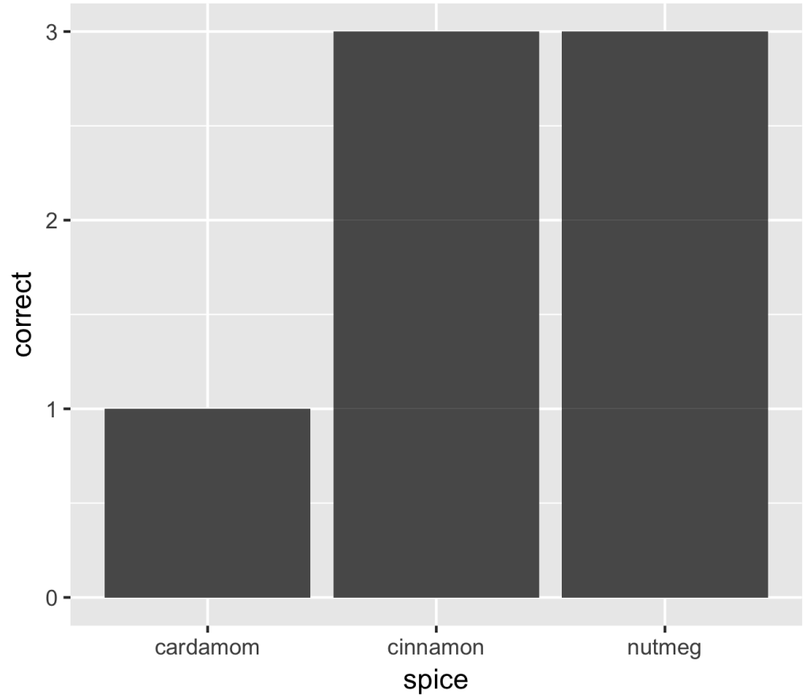
Which spice was the hardest to guess? Count it!

```
ggplot(juniors_tidy, aes(baker, correct)) +
  geom_col()
```



Which spice was the hardest to guess? Plot it!

```
ggplot(juniors_tidy, aes(spice, correct)) +
  geom_col()
```



Insert title here...

tame data

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

trial	Emma	Harry	Ruby	Zainab
cinnamon_1	1	1	1	0
cardamom_2	0	1	0	NA
nutmeg_3	1	1	1	0

tidy data

baker	spice	order	correct
Emma	Cinnamon	1	1
Harry	Cinnamon	1	1
Ruby	Cinnamon	1	1
Zainab	Cinnamon	1	0
Emma	Cardamom	2	0
Harry	Cardamom	2	1
Ruby	Cardamom	2	0
Zainab	Cardamom	2	NA
Emma	Nutmeg	3	1
Harry	Nutmeg	3	1
Ruby	Nutmeg	3	1
Zainab	Nutmeg	3	0

Let's get to work!

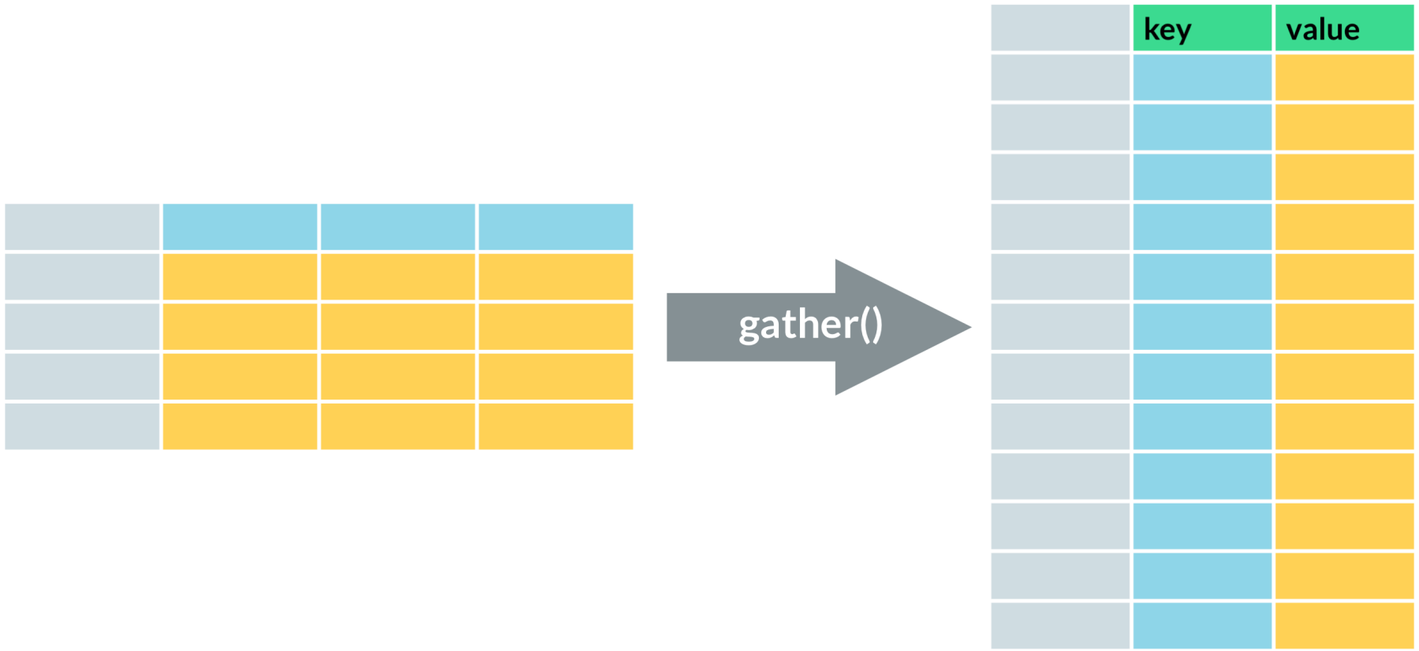
WORKING WITH DATA IN THE TIDYVERSE

Gather

WORKING WITH DATA IN THE TIDYVERSE

Alison Hill
Professor & Data Scientist

The `tidyr` package



[1] http://tidyr.tidyverse.org ## Title ```yaml type: FullSlide key: e6e5223c49 hide_title: true ```

Gather: usage

```
?gather
```

Gather Columns Into Key-Value Pairs.

Gather takes multiple columns and collapses into key-value pairs, duplicating all other columns as needed. You use `gather()` when you notice that you have columns that are not variables.

Usage

```
gather(data, key = "key", value = "value", ..., na.rm = FALSE,
  convert = FALSE, factor_key = FALSE)
```

Gather: arguments

```
?gather
```

Arguments

data	A data frame.
key, value	Names of new key and value columns, as strings or symbols. This argument is passed by expression and supports quasiquotation (you can unquote strings and symbols). The name is captured from the expression with `rlang::quo_name()` (note that this kind of interface where symbols do not represent actual objects is now discouraged in the tidyverse; we support it here for backward compatibility).
...	A selection of columns. If empty, all variables are selected. You can supply bare variable names, select all variables between x and z with `x:z`, exclude y with `-y`. For more options, see the `dplyr::select()` documentation. See also the section on selection rules below.

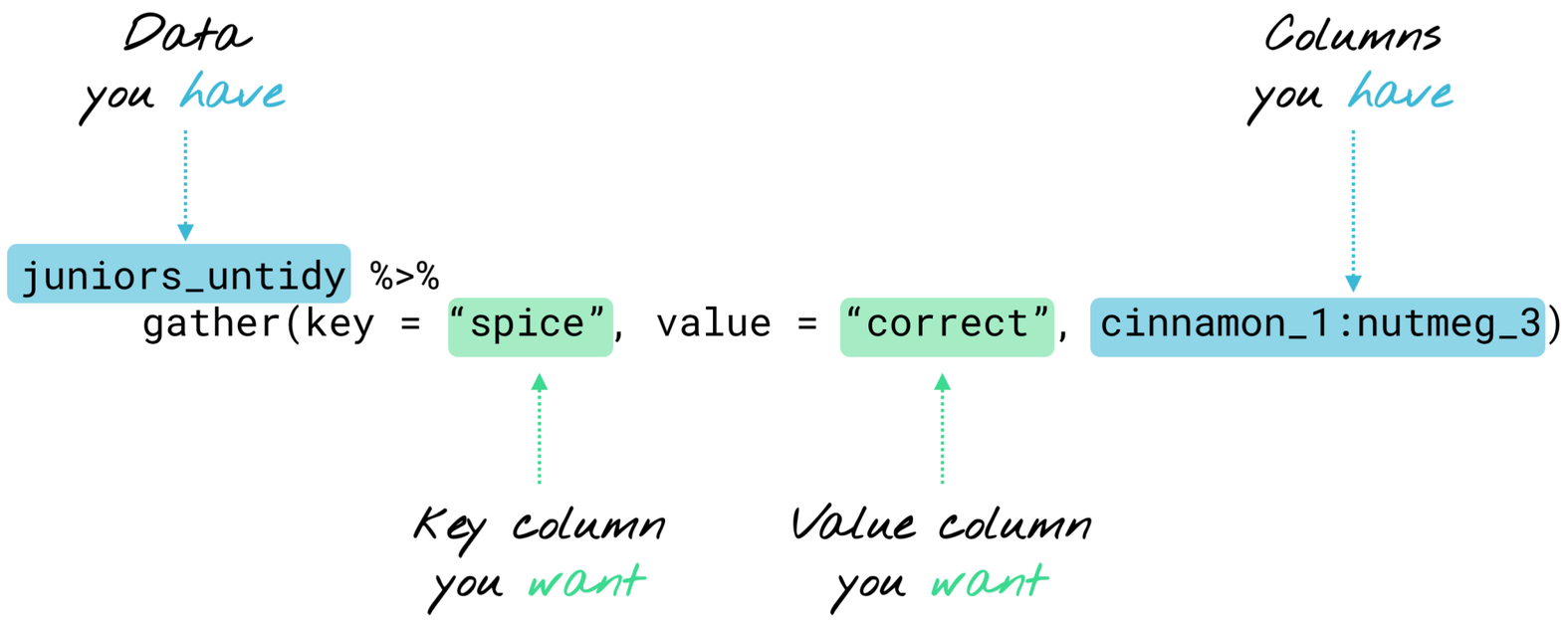
Gathering juniors

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

Gathering what you have into what you want



The key column

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

The key column

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

The key column

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

The value column

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

The value column

```
gather(key = “spice”, value = “correct”, cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

The value column

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

A little trick

```
gather(key = "key", value = "value", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	key	value
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

Let's get to work!

WORKING WITH DATA IN THE TIDYVERSE

Separate

WORKING WITH DATA IN THE TIDYVERSE

Alison Hill
Professor & Data Scientist

Gathering the juniors data

```
gather(key = "spice", value = "correct", cinnamon_1:nutmeg_3)
```

baker	cinnamon_1	cardamom_2	nutmeg_3
Emma	1	0	1
Harry	1	1	1
Ruby	1	0	1
Zainab	0	NA	0

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

Separate: usage

```
?separate
```

Separate One Column Into Multiple Columns.

Given either regular expression or a vector of character positions, `separate()` turns a single character column into multiple columns.

Usage

```
separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE,
  convert = FALSE, extra = "warn", fill = "warn", ...)
```

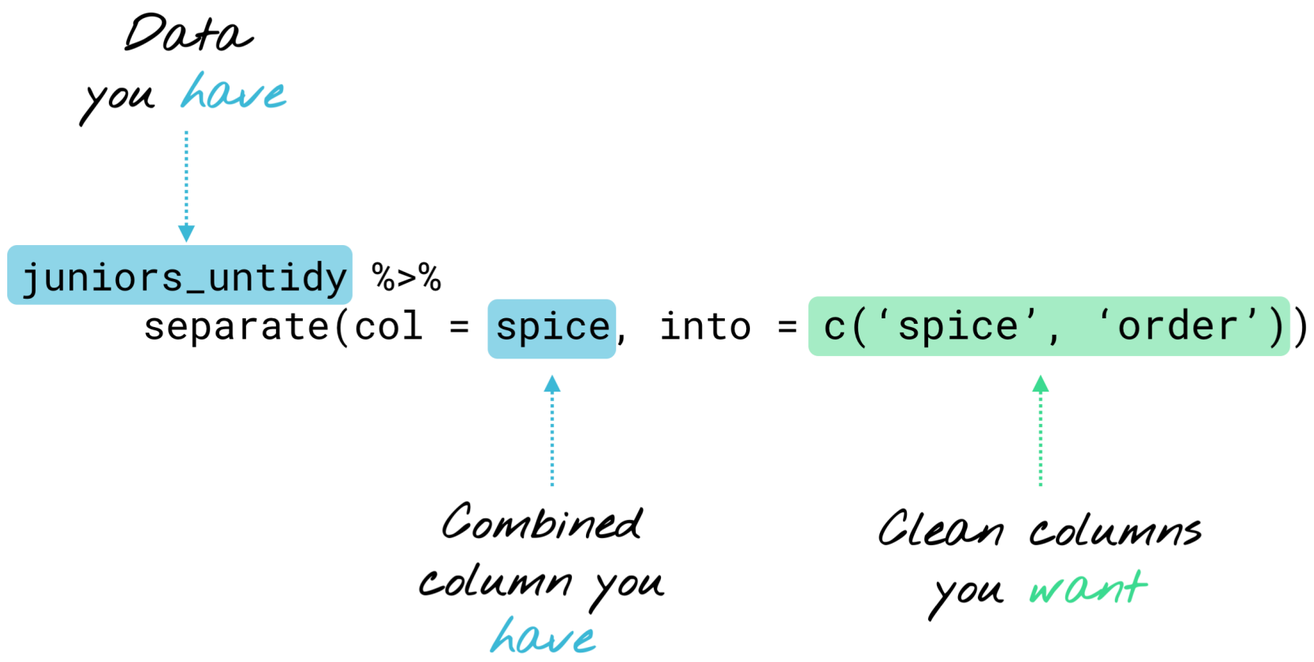
Separate: arguments

```
?separate
```

Arguments

data	A data frame.
col	Column name or position. This is passed to `tidyselect::vars_pull()`.
	This argument is passed by expression and supports quasiquotation (you can unquote column names or column positions).
into	Names of new variables to create as character vector.

Separating what you have into what you want



Separate `spice`

```
separate(col = spice, into = c('spice', 'order'))
```

baker	spice	correct
Emma	cinnamon_1	1
Harry	cinnamon_1	1
Ruby	cinnamon_1	1
Zainab	cinnamon_1	0
Emma	cardamom_2	0
Harry	cardamom_2	1
Ruby	cardamom_2	0
Zainab	cardamom_2	NA
Emma	nutmeg_3	1
Harry	nutmeg_3	1
Ruby	nutmeg_3	1
Zainab	nutmeg_3	0

Baker	spice	order	correct
Emma	cinnamon	1	1
Harry	cinnamon	1	1
Ruby	cinnamon	1	1
Zainab	cinnamon	1	0
Emma	cardamom	2	0
Harry	cardamom	2	1
Ruby	cardamom	2	0
Zainab	cardamom	2	NA
Emma	nutmeg	3	1
Harry	nutmeg	3	1
Ruby	nutmeg	3	1
Zainab	nutmeg	3	0

Reminder: pre-separate

```
juniors_untidy %>%
  gather(key = spice, value = correct, -baker)
```

```
# A tibble: 12 x 3
   baker  spice      correct
   <chr>  <chr>        <int>
 1 Emma   cinnamon_1      1
 2 Harry  cinnamon_1      1
 3 Ruby   cinnamon_1      1
 4 Zainab cinnamon_1      0
 5 Emma   cardamom_2      0
 6 Harry  cardamom_2      1
 7 Ruby   cardamom_2      0
 8 Zainab cardamom_2     NA
 9 Emma   nutmeg_3        1
10 Harry  nutmeg_3        1
11 Ruby   nutmeg_3        1
12 Zainab nutmeg_3        0
```

Gather and separate

```
juniors_untidy %>%
    gather(key = "spice", value = "correct", -baker) %>%
    separate(spice, into = c("spice", "order"))
```

```
# A tibble: 12 x 4
   baker  spice    order correct
   <chr>  <chr>    <chr>   <int>
 1 Emma   cinnamon 1           1
 2 Harry  cinnamon 1           1
 3 Ruby   cinnamon 1           1
 4 Zainab cinnamon 1           0
 5 Emma   cardamom 2           0
 6 Harry  cardamom 2           1
 7 Ruby   cardamom 2           0
 8 Zainab cardamom 2          NA
 9 Emma   nutmeg   3           1
10 Harry  nutmeg   3           1
11 Ruby   nutmeg   3           1
12 Zainab nutmeg   3           0
```

Gather, separate, and convert types

```
juniors_untidy %>%
    gather(key = "spice", value = "correct", -baker) %>%
    separate(spice, into = c("spice", "order"), convert = TRUE)
```

```
# A tibble: 12 x 4
   baker  spice    order correct
   <chr>  <chr>    <int>   <int>
 1 Emma   cinnamon     1       1
 2 Harry  cinnamon     1       1
 3 Ruby   cinnamon     1       1
 4 Zainab cinnamon     1       0
 5 Emma   cardamom     2       0
 6 Harry  cardamom     2       1
 7 Ruby   cardamom     2       0
 8 Zainab cardamom     2      NA
 9 Emma   nutmeg       3       1
10 Harry  nutmeg       3       1
11 Ruby   nutmeg       3       1
12 Zainab nutmeg       3       0
```

Before and after separate

```
# A tibble: 12 x 3
   baker  spice      correct
   <chr>  <chr>        <int>
 1 Emma   cinnamon_1       1
 2 Harry  cinnamon_1       1
 3 Ruby   cinnamon_1       1
 4 Zainab cinnamon_1       0
 5 Emma   cardamom_2       0
 6 Harry  cardamom_2       1
 7 Ruby   cardamom_2       0
 8 Zainab cardamom_2      NA
 9 Emma   nutmeg_3         1
10 Harry  nutmeg_3         1
11 Ruby   nutmeg_3         1
12 Zainab nutmeg_3         0
```

```
# A tibble: 12 x 4
   baker  spice    order correct
   <chr>  <chr>    <int>   <int>
 1 Emma   cinnamon     1       1
 2 Harry  cinnamon     1       1
 3 Ruby   cinnamon     1       1
 4 Zainab cinnamon     1       0
 5 Emma   cardamom     2       0
 6 Harry  cardamom     2       1
 7 Ruby   cardamom     2       0
 8 Zainab cardamom     2      NA
 9 Emma   nutmeg       3       1
10 Harry  nutmeg       3       1
11 Ruby   nutmeg       3       1
12 Zainab nutmeg       3       0
```

The `sep` argument

```
?separate
```

Arguments

data	A data frame.
col	Column name or position. This is passed to `tidyselect::vars_pull()`. This argument is passed by expression and supports quasiquotation (you can unquote column names or column positions).
into	Names of new variables to create as character vector.
sep	Separator between columns. If character, is interpreted as a regular expression. The default value is a regular expression that matches any sequence of non-alphanumeric values. If numeric, interpreted as positions to split at. Positive values start at 1 at the far-left of the string; negative value start at -1 at the far-right of the string. The length of `sep` should be one less than `into`.

Let's practice!

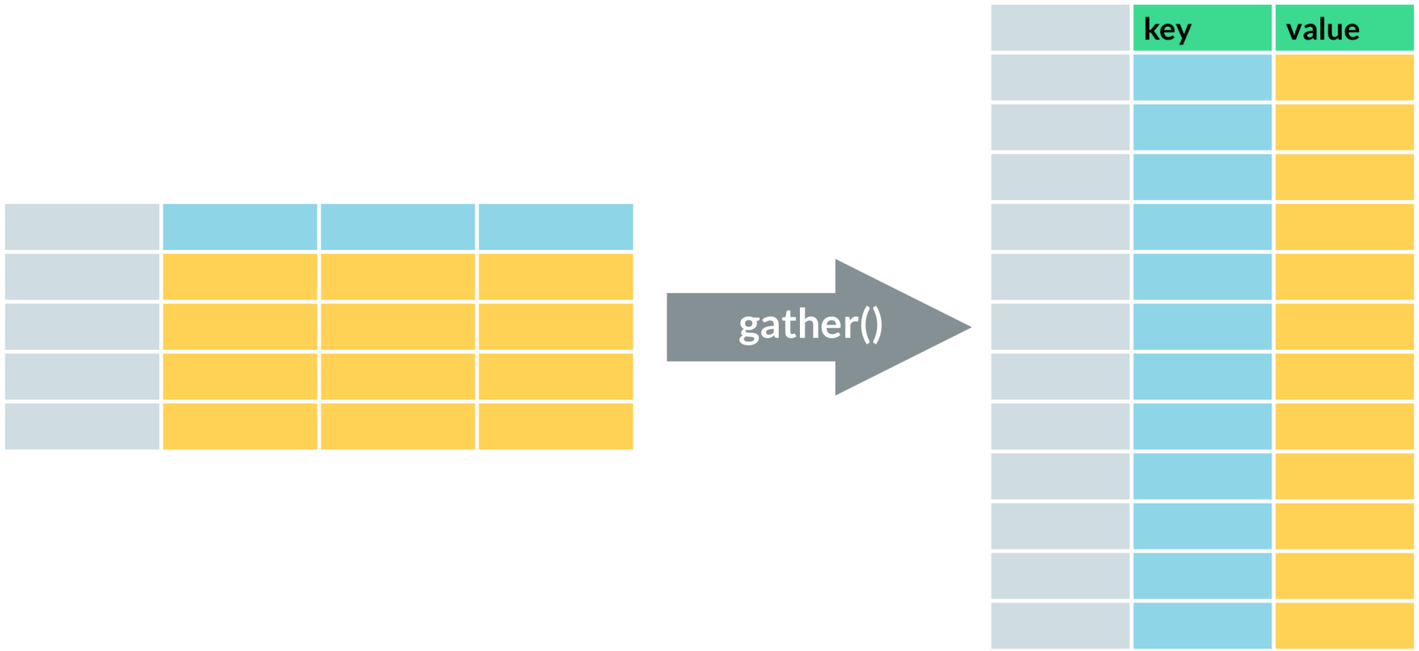
WORKING WITH DATA IN THE TIDYVERSE

Spread

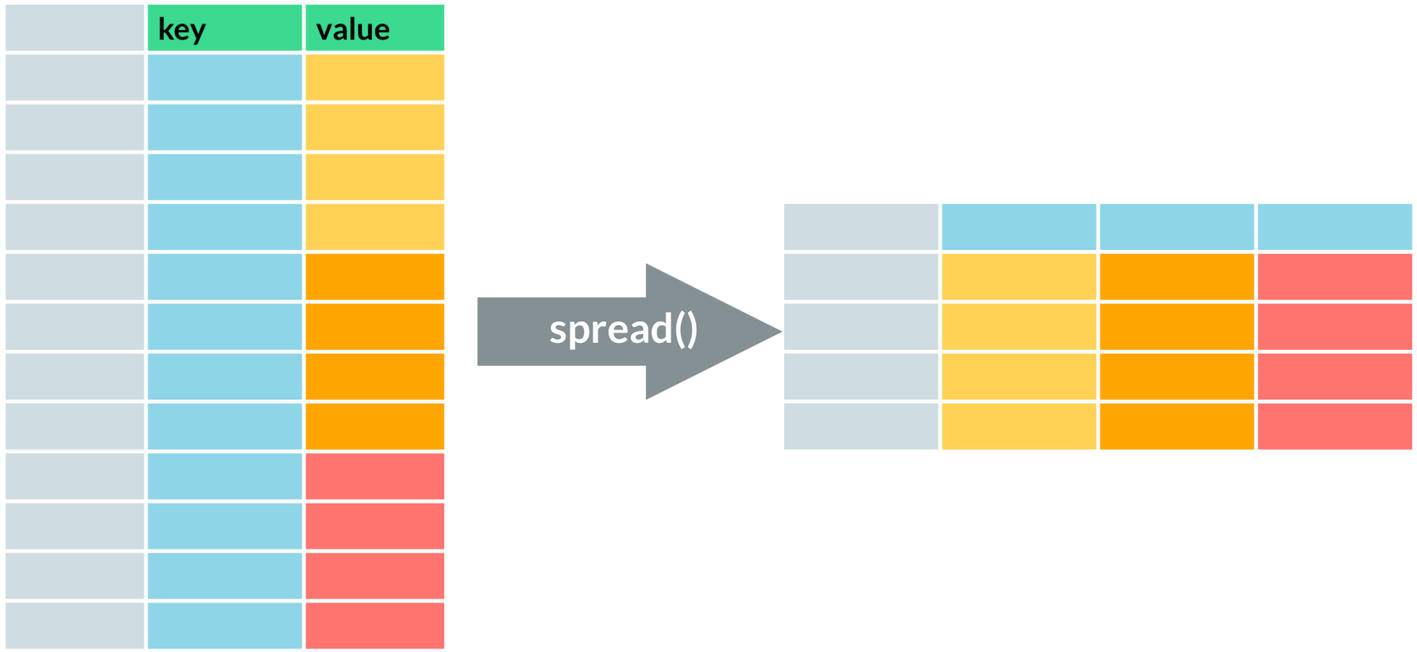
WORKING WITH DATA IN THE TIDYVERSE

Alison Hill
Professor & Data Scientist

Gather

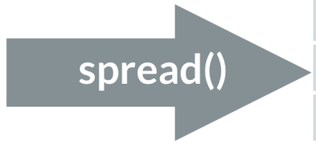


Spread



Spread

baker	key	value
Emma	age	11
Harry	age	10
Ruby	age	11
Zainab	age	10
Emma	outcome	finalist
Harry	outcome	winner
Ruby	outcome	finalist
Zainab	outcome	finalist
Emma	spices	2
Harry	spices	3
Ruby	spices	2
Zainab	spices	0



baker	age	outcome	spices
Emma	11	finalist	2
Harry	10	winner	3
Ruby	11	finalist	2
Zainab	10	finalist	0

Spread: usage

```
?spread
```

Spread A Key-Value Pair Across Multiple Columns.

Spread a key-value pair across multiple columns.

Usage

```
spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE,
  sep = NULL)
```

Spread: arguments

```
?spread
```

Arguments

data	A data frame.
key, value	Column names or positions. This is passed to `tidyselect::vars_pull()` . These arguments are passed by expression and support quasiquotation (you can unquote column names or column positions).

Using spread

```
juniors_jumbled
```

```
# A tibble: 12 x 3
   baker  key     value
   <chr>  <chr>   <chr>
 1 Emma   age     11
 2 Harry  age     10
 3 Ruby   age     11
 4 Zainab age     10
 5 Emma   outcome finalist
 6 Harry  outcome winner
 7 Ruby   outcome finalist
 8 Zainab outcome finalist
 9 Emma   spices  2
10 Harry  spices  3
11 Ruby   spices  2
12 Zainab spices  0
```

```
juniors_jumbled %>%
  spread(key = key, value = value)
```

```
# A tibble: 4 x 4
  baker  age   outcome  spices
  <chr>  <chr> <chr>    <chr>
1 Emma   11    finalist 2
2 Harry  10    winner   3
3 Ruby   11    finalist 2
4 Zainab 10    finalist 0
```

Spread and convert

```
juniors_jumbled
```

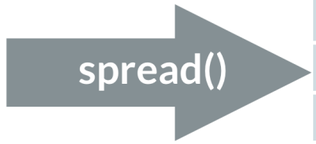
```
# A tibble: 12 x 3
   baker  key     value
   <chr>  <chr>   <chr>
 1 Emma   age     11
 2 Harry  age     10
 3 Ruby   age     11
 4 Zainab age     10
 5 Emma   outcome finalist
 6 Harry  outcome winner
 7 Ruby   outcome finalist
 8 Zainab outcome finalist
 9 Emma   spices  2
10 Harry  spices  3
11 Ruby   spices  2
12 Zainab spices  0
```

```
juniors_jumbled %>%
  spread(key = key, value = value,
         convert = TRUE)
```

```
# A tibble: 4 x 4
  baker    age outcome  spices
  <chr>  <int> <chr>     <int>
1 Emma      11 finalist      2
2 Harry     10 winner        3
3 Ruby      11 finalist      2
4 Zainab    10 finalist      0
```

Spread review

baker	key	value
Emma	age	11
Harry	age	10
Ruby	age	11
Zainab	age	10
Emma	outcome	finalist
Harry	outcome	winner
Ruby	outcome	finalist
Zainab	outcome	finalist
Emma	spices	2
Harry	spices	3
Ruby	spices	2
Zainab	spices	0



baker	age	outcome	spices
Emma	11	finalist	2
Harry	10	winner	3
Ruby	11	finalist	2
Zainab	10	finalist	0

Let's practice!

WORKING WITH DATA IN THE TIDYVERSE

Tidy multiple sets of columns

WORKING WITH DATA IN THE TIDYVERSE

Alison Hill
Professor & Data Scientist

Multiple sets to gather

```
students
```

```
# A tibble: 4 x 5
  student math_num chem_num math_let chem_let
  <chr>      <int>    <int> <chr>    <chr>
1 Emma          80       98 B-       A+
2 Harry         90       75 A-       C
3 Ruby          95       70 A        C-
4 Zainab        85       90 B        A-
```

```
patients
```

```
# A tibble: 4 x 5
  patient height_1 height_2 weight_1 weight_2
  <chr>      <int>    <int>    <int>    <int>
1 Emma          54       59       72       95
2 Harry         55       58       68       90
3 Ruby          55       60       70       94
4 Zainab        53       58       71       95
```

```
students
```

```
# A tibble: 4 x 5
  student math_num chem_num math_let chem_let
  <chr>      <int>    <int> <chr>    <chr>
1 Emma          80       98 B-       A+
2 Harry         90       75 A-       C
3 Ruby          95       70 A        C-
4 Zainab        85       90 B        A-

# A tibble: 8 x 4
  student subject let     num
  <chr>   <chr>   <chr> <int>
1 Emma    chem    A+       98
2 Emma    math    B-       80
3 Harry   chem    C        75
4 Harry   math    A-       90
5 Ruby    chem    C-       70
6 Ruby    math    A        95
7 Zainab  chem    A-       90
8 Zainab  math    B        85
```

```
patients
```

```
# A tibble: 4 x 5
  patient height_1 height_2 weight_1 weight_2
  <chr>      <int>    <int>    <int>    <int>
1 Emma          54       59       72       95
2 Harry         55       58       68       90
3 Ruby          55       60       70       94
4 Zainab        53       58       71       95
```

```
# A tibble: 8 x 4
  patient visit height weight
  <chr>   <chr>  <int>  <int>
1 Emma    1         54     72
2 Emma    2         59     95
3 Harry   1         55     68
4 Harry   2         58     90
5 Ruby    1         55     70
6 Ruby    2         60     94
7 Zainab  1         53     71
8 Zainab  2         58     95
```

Untidy vs. tidy

```
juniors_multi
```

```
# A tibble: 3 x 7
  baker  score_1 score_2 score_3 guess_1  guess_2  guess_3
  <chr>    <int>   <int>   <int> <chr>    <chr>    <chr>
1 Emma         1       0       1 cinnamon cloves   nutmeg
2 Harry        1       1       1 cinnamon cardamom nutmeg
3 Ruby         1       0       1 cinnamon cumin    nutmeg
```

```
juniors_tidy %>% slice(6)
```

```
# A tibble: 9 x 4
   baker  order guess    score
   <chr>  <int> <chr>    <chr>
 1 Emma       1 cinnamon 1
 2 Emma       2 cloves   0
 3 Emma       3 nutmeg   1
 4 Harry      1 cinnamon 1
 5 Harry      2 cardamom 1
 6 Harry      3 nutmeg   1
```

Step 1: `gather`

```
juniors_multi %>%
  gather(key = "key", value = "value", score_1:guess_3)
```

```
# A tibble: 24 x 3
   baker key     value
   <chr> <chr>   <chr>
 1 Emma  guess_1 cinnamon
 2 Emma  guess_2 cloves
 3 Emma  guess_3 nutmeg
 4 Emma  score_1 1
 5 Emma  score_2 0
 6 Emma  score_3 1
 7 Harry guess_1 cinnamon
 8 Harry guess_2 cardamom
 9 Harry guess_3 nutmeg
10 Harry score_1 1
# ... with 14 more rows
```

Step 2: `separate`

```
juniors_multi %>%
  gather(key = "key", value = "value", score_1:guess_3) %>%
  separate(key, into = c("var", "order"), convert = TRUE)
```

```
# A tibble: 24 x 4
   baker var   order value
   <chr> <chr> <int> <chr>
 1 Emma  guess     1 cinnamon
 2 Emma  guess     2 cloves
 3 Emma  guess     3 nutmeg
 4 Emma  score     1 1
 5 Emma  score     2 0
 6 Emma  score     3 1
 7 Harry guess     1 cinnamon
 8 Harry guess     2 cardamom
 9 Harry guess     3 nutmeg
10 Harry score     1 1
# ... with 14 more rows
```

Before and after spread

```
# A tibble: 24 x 4
   baker var   order value
   <chr> <chr> <int> <chr>
 1 Emma  guess     1 cinnamon
 2 Emma  guess     2 cloves
 3 Emma  guess     3 nutmeg
 4 Emma  score     1 1
 5 Emma  score     2 0
 6 Emma  score     3 1
 7 Harry guess     1 cinnamon
 8 Harry guess     2 cardamom
 9 Harry guess     3 nutmeg
10 Harry score     1 1
# ... with 14 more rows
```

```
# A tibble: 12 x 4
   baker  order guess    score
   <chr>  <int> <chr>    <chr>
 1 Emma       1 cinnamon 1
 2 Emma       2 cloves   0
 3 Emma       3 nutmeg   1
 4 Harry      1 cinnamon 1
 5 Harry      2 cardamom 1
 6 Harry      3 nutmeg   1
 7 Ruby       1 cinnamon 1
 8 Ruby       2 cumin    0
 9 Ruby       3 nutmeg   1
10 Zainab     1 cardamom 0
11 Zainab     2 NA       NA
12 Zainab     3 cinnamon 0
```

Step 3: `spread`

```
juniors_multi %>%
  gather(key = "key", value = "value", score_1:guess_3) %>%
  separate(key, into = c("var", "order"), convert = TRUE) %>%
  spread(var, value)
```

```
# A tibble: 12 x 4
   baker  order guess    score
   <chr>  <int> <chr>    <chr>
 1 Emma       1 cinnamon 1
 2 Emma       2 cloves   0
 3 Emma       3 nutmeg   1
 4 Harry      1 cinnamon 1
 5 Harry      2 cardamom 1
 6 Harry      3 nutmeg   1
 7 Ruby       1 cinnamon 1
 8 Ruby       2 cumin    0
 9 Ruby       3 nutmeg   1
10 Zainab     1 cardamom 0
11 Zainab     2 NA       NA
12 Zainab     3 cinnamon 0
```

Untidy to tidy

```
juniors_multi
```

```
# A tibble: 4 x 7
  baker  score_1 score_2 score_3 guess_1  guess_2  guess_3
  <chr>    <int>   <int>   <int> <chr>    <chr>    <chr>
1 Emma         1       0       1 cinnamon cloves   nutmeg
2 Harry        1       1       1 cinnamon cardamom nutmeg
3 Ruby         1       0       1 cinnamon cumin    nutmeg
```

```
juniors_tidy %>% slice(6)
```

```
# A tibble: 12 x 4
   baker  order guess    score
   <chr>  <int> <chr>    <chr>
 1 Emma       1 cinnamon 1
 2 Emma       2 cloves   0
 3 Emma       3 nutmeg   1
 4 Harry      1 cinnamon 1
 5 Harry      2 cardamom 1
 6 Harry      3 nutmeg   1
```

Let's practice!

WORKING WITH DATA IN THE TIDYVERSE