There’s a lot going on in the development version of {tidyr}. New functions for pivoting data  
frames, pivot\_wide() and pivot\_long() are coming, and will replace the current functions,  
spread() and gather(). spread() and gather() will remain in the package though:

You may have heard a rumour that gather/spread are going away. This is simply not true (they’ll stay around forever) but I am working on better replacements which you can learn about at <https://t.co/sU2GzWeBaf>. Now is a great time for feedback! [#rstats](https://twitter.com/hashtag/rstats?src=hash&ref_src=twsrc%5Etfw)

— Hadley Wickham (@hadleywickham) [March 19, 2019](https://twitter.com/hadleywickham/status/1108107722128613377?ref_src=twsrc%5Etfw)

If you want to try out these new functions, you need to install the development version of {tidyr}:

devtools::install\_github("tidyverse/tidyr")

and you can read the vignette [here](https://tidyr.tidyverse.org/dev/articles/pivot.html#many-variables-in-column-names).  
Because these functions are still being developed, some more changes might be introduced, but I guess  
that the main functionality will not change much.

Let’s play around with these functions and the mtcars data set. First let’s load the packages and  
the data:

library(tidyverse)

data(mtcars)

First, let’s create a wide dataset, by *spreading* the levels of the “am” column to two new columns:

mtcars\_wide1 <- mtcars %>%

pivot\_wide(names\_from = "am", values\_from = "mpg")

mtcars\_wide1 %>%

select(`0`, `1`, everything())

## # A tibble: 32 x 11

## `0` `1` cyl disp hp drat wt qsec vs gear carb

##

## 1 NA 21 6 160 110 3.9 2.62 16.5 0 4 4

## 2 NA 21 6 160 110 3.9 2.88 17.0 0 4 4

## 3 NA 22.8 4 108 93 3.85 2.32 18.6 1 4 1

## 4 21.4 NA 6 258 110 3.08 3.22 19.4 1 3 1

## 5 18.7 NA 8 360 175 3.15 3.44 17.0 0 3 2

## 6 18.1 NA 6 225 105 2.76 3.46 20.2 1 3 1

## 7 14.3 NA 8 360 245 3.21 3.57 15.8 0 3 4

## 8 24.4 NA 4 147. 62 3.69 3.19 20 1 4 2

## 9 22.8 NA 4 141. 95 3.92 3.15 22.9 1 4 2

## 10 19.2 NA 6 168. 123 3.92 3.44 18.3 1 4 4

## # … with 22 more rows

pivot\_wide()’s arguments are quite explicit: names\_from = is where you specify the column that  
will be spread across the data frame, meaning, the levels of this column will become new columns.  
values\_from = is where you specify the column that will fill in the values of the new columns.

“0” and “1” are the new columns (“am” had two levels, 0 and 1), which contain the miles per  
gallon for manual and automatic cars respectively. Let’s also take a look at the data frame itself:

mtcars\_wide1 %>%

select(`0`, `1`, everything())

## # A tibble: 32 x 11

## `0` `1` cyl disp hp drat wt qsec vs gear carb

##

## 1 NA 21 6 160 110 3.9 2.62 16.5 0 4 4

## 2 NA 21 6 160 110 3.9 2.88 17.0 0 4 4

## 3 NA 22.8 4 108 93 3.85 2.32 18.6 1 4 1

## 4 21.4 NA 6 258 110 3.08 3.22 19.4 1 3 1

## 5 18.7 NA 8 360 175 3.15 3.44 17.0 0 3 2

## 6 18.1 NA 6 225 105 2.76 3.46 20.2 1 3 1

## 7 14.3 NA 8 360 245 3.21 3.57 15.8 0 3 4

## 8 24.4 NA 4 147. 62 3.69 3.19 20 1 4 2

## 9 22.8 NA 4 141. 95 3.92 3.15 22.9 1 4 2

## 10 19.2 NA 6 168. 123 3.92 3.44 18.3 1 4 4

## # … with 22 more rows

Now suppose that we want to spread the values of “am” times “cyl”, and filling the data with the  
values of “mpg”:

mtcars\_wide2 <- mtcars %>%

pivot\_wide(names\_from = c("am", "cyl"), values\_from = "mpg")

mtcars\_wide2 %>%

select(matches("^0|1"), everything())

## # A tibble: 32 x 14

## `1\_6` `1\_4` `0\_6` `0\_8` `0\_4` `1\_8` disp hp drat wt qsec vs

##

## 1 21 NA NA NA NA NA 160 110 3.9 2.62 16.5 0

## 2 21 NA NA NA NA NA 160 110 3.9 2.88 17.0 0

## 3 NA 22.8 NA NA NA NA 108 93 3.85 2.32 18.6 1

## 4 NA NA 21.4 NA NA NA 258 110 3.08 3.22 19.4 1

## 5 NA NA NA 18.7 NA NA 360 175 3.15 3.44 17.0 0

## 6 NA NA 18.1 NA NA NA 225 105 2.76 3.46 20.2 1

## 7 NA NA NA 14.3 NA NA 360 245 3.21 3.57 15.8 0

## 8 NA NA NA NA 24.4 NA 147. 62 3.69 3.19 20 1

## 9 NA NA NA NA 22.8 NA 141. 95 3.92 3.15 22.9 1

## 10 NA NA 19.2 NA NA NA 168. 123 3.92 3.44 18.3 1

## # … with 22 more rows, and 2 more variables: gear , carb

As you can see, this is easily achieved by simply providing more columns to names\_from =.

Finally, it is also possible to use an optional data set which contains the specifications of the  
new columns:

mtcars\_spec <- mtcars %>%

expand(am, cyl, .value = "mpg") %>%

unite(".name", am, cyl, remove = FALSE)

mtcars\_spec

## # A tibble: 6 x 4

## .name am cyl .value

##

## 1 0\_4 0 4 mpg

## 2 0\_6 0 6 mpg

## 3 0\_8 0 8 mpg

## 4 1\_4 1 4 mpg

## 5 1\_6 1 6 mpg

## 6 1\_8 1 8 mpg

This optional data set defines how the columns “0\_4”, “0\_6” etc are constructed, and also the  
value that shall be used to fill in the values. “am” and “cyl” will be used to create the “.name”  
and the “mpg” column will be used for the “.value”:

mtcars %>%

pivot\_wide(spec = mtcars\_spec) %>%

select(matches("^0|1"), everything())

## # A tibble: 32 x 14

## `0\_4` `0\_6` `0\_8` `1\_4` `1\_6` `1\_8` disp hp drat wt qsec vs

##

## 1 NA NA NA NA 21 NA 160 110 3.9 2.62 16.5 0

## 2 NA NA NA NA 21 NA 160 110 3.9 2.88 17.0 0

## 3 NA NA NA 22.8 NA NA 108 93 3.85 2.32 18.6 1

## 4 NA 21.4 NA NA NA NA 258 110 3.08 3.22 19.4 1

## 5 NA NA 18.7 NA NA NA 360 175 3.15 3.44 17.0 0

## 6 NA 18.1 NA NA NA NA 225 105 2.76 3.46 20.2 1

## 7 NA NA 14.3 NA NA NA 360 245 3.21 3.57 15.8 0

## 8 24.4 NA NA NA NA NA 147. 62 3.69 3.19 20 1

## 9 22.8 NA NA NA NA NA 141. 95 3.92 3.15 22.9 1

## 10 NA 19.2 NA NA NA NA 168. 123 3.92 3.44 18.3 1

## # … with 22 more rows, and 2 more variables: gear , carb

Using a spec is especially useful if you need to make new levels that are not in the data.  
For instance, suppose that there are actually 10-cylinder cars too, but they do not appear in our  
sample. We would like to make the fact that they’re missing explicit:

mtcars\_spec2 <- mtcars %>%

expand(am, "cyl" = c(cyl, 10), .value = "mpg") %>%

unite(".name", am, cyl, remove = FALSE)

mtcars\_spec2

## # A tibble: 8 x 4

## .name am cyl .value

##

## 1 0\_4 0 4 mpg

## 2 0\_6 0 6 mpg

## 3 0\_8 0 8 mpg

## 4 0\_10 0 10 mpg

## 5 1\_4 1 4 mpg

## 6 1\_6 1 6 mpg

## 7 1\_8 1 8 mpg

## 8 1\_10 1 10 mpg

mtcars %>%

pivot\_wide(spec = mtcars\_spec2) %>%

select(matches("^0|1"), everything())

## # A tibble: 32 x 16

## `0\_4` `0\_6` `0\_8` `0\_10` `1\_4` `1\_6` `1\_8` `1\_10` disp hp drat

##

## 1 NA NA NA NA NA 21 NA NA 160 110 3.9

## 2 NA NA NA NA NA 21 NA NA 160 110 3.9

## 3 NA NA NA NA 22.8 NA NA NA 108 93 3.85

## 4 NA 21.4 NA NA NA NA NA NA 258 110 3.08

## 5 NA NA 18.7 NA NA NA NA NA 360 175 3.15

## 6 NA 18.1 NA NA NA NA NA NA 225 105 2.76

## 7 NA NA 14.3 NA NA NA NA NA 360 245 3.21

## 8 24.4 NA NA NA NA NA NA NA 147. 62 3.69

## 9 22.8 NA NA NA NA NA NA NA 141. 95 3.92

## 10 NA 19.2 NA NA NA NA NA NA 168. 123 3.92

## # … with 22 more rows, and 5 more variables: wt , qsec ,

## # vs , gear , carb

As you can see, we now have two more columns have been added, and they are full of NA’s.

Now, let’s try to go from wide to long data sets, using pivot\_long():

mtcars\_wide1 %>%

pivot\_long(cols = c(`1`, `0`), names\_to = "am", values\_to = "mpg") %>%

select(am, mpg, everything())

## # A tibble: 64 x 11

## am mpg cyl disp hp drat wt qsec vs gear carb

##

## 1 1 21 6 160 110 3.9 2.62 16.5 0 4 4

## 2 0 NA 6 160 110 3.9 2.62 16.5 0 4 4

## 3 1 21 6 160 110 3.9 2.88 17.0 0 4 4

## 4 0 NA 6 160 110 3.9 2.88 17.0 0 4 4

## 5 1 22.8 4 108 93 3.85 2.32 18.6 1 4 1

## 6 0 NA 4 108 93 3.85 2.32 18.6 1 4 1

## 7 1 NA 6 258 110 3.08 3.22 19.4 1 3 1

## 8 0 21.4 6 258 110 3.08 3.22 19.4 1 3 1

## 9 1 NA 8 360 175 3.15 3.44 17.0 0 3 2

## 10 0 18.7 8 360 175 3.15 3.44 17.0 0 3 2

## # … with 54 more rows

The arguments of pivot\_long() are quite explicit too, and similar to the ones in pivot\_wide().  
cols = is where the user specifies the columns that need to be pivoted. names\_to = is where  
the user can specify the name of the new columns, whose levels will be exactly the ones specified  
to cols =. values\_to = is where the user specifies the column name of the new column that  
will contain the values.

It is also possible to specify the columns that should not be transformed, by using -:

mtcars\_wide1 %>%

pivot\_long(cols = -matches("^[[:alpha:]]"), names\_to = "am", values\_to = "mpg") %>%

select(am, mpg, everything())

## # A tibble: 64 x 11

## am mpg cyl disp hp drat wt qsec vs gear carb

##

## 1 1 21 6 160 110 3.9 2.62 16.5 0 4 4

## 2 0 NA 6 160 110 3.9 2.62 16.5 0 4 4

## 3 1 21 6 160 110 3.9 2.88 17.0 0 4 4

## 4 0 NA 6 160 110 3.9 2.88 17.0 0 4 4

## 5 1 22.8 4 108 93 3.85 2.32 18.6 1 4 1

## 6 0 NA 4 108 93 3.85 2.32 18.6 1 4 1

## 7 1 NA 6 258 110 3.08 3.22 19.4 1 3 1

## 8 0 21.4 6 258 110 3.08 3.22 19.4 1 3 1

## 9 1 NA 8 360 175 3.15 3.44 17.0 0 3 2

## 10 0 18.7 8 360 175 3.15 3.44 17.0 0 3 2

## # … with 54 more rows

Here the columns that should not be modified are all those that start with a letter, hence the “[1](http://www.brodrigues.co/blog/2019-03-20-pivot/#fn1)”  
regular expression. It is also possible to remove all the NA’s from the data frame, with na.rm =.

mtcars\_wide1 %>%

pivot\_long(cols = c(`1`, `0`), names\_to = "am", values\_to = "mpg", na.rm = TRUE) %>%

select(am, mpg, everything())

## # A tibble: 32 x 11

## am mpg cyl disp hp drat wt qsec vs gear carb

##

## 1 1 21 6 160 110 3.9 2.62 16.5 0 4 4

## 2 1 21 6 160 110 3.9 2.88 17.0 0 4 4

## 3 1 22.8 4 108 93 3.85 2.32 18.6 1 4 1

## 4 0 21.4 6 258 110 3.08 3.22 19.4 1 3 1

## 5 0 18.7 8 360 175 3.15 3.44 17.0 0 3 2

## 6 0 18.1 6 225 105 2.76 3.46 20.2 1 3 1

## 7 0 14.3 8 360 245 3.21 3.57 15.8 0 3 4

## 8 0 24.4 4 147. 62 3.69 3.19 20 1 4 2

## 9 0 22.8 4 141. 95 3.92 3.15 22.9 1 4 2

## 10 0 19.2 6 168. 123 3.92 3.44 18.3 1 4 4

## # … with 22 more rows

We can also pivot data frames where the names of the columns are made of two or more variables,  
for example in our mtcars\_wide2 data frame:

mtcars\_wide2 %>%

select(matches("^0|1"), everything())

## # A tibble: 32 x 14

## `1\_6` `1\_4` `0\_6` `0\_8` `0\_4` `1\_8` disp hp drat wt qsec vs

##

## 1 21 NA NA NA NA NA 160 110 3.9 2.62 16.5 0

## 2 21 NA NA NA NA NA 160 110 3.9 2.88 17.0 0

## 3 NA 22.8 NA NA NA NA 108 93 3.85 2.32 18.6 1

## 4 NA NA 21.4 NA NA NA 258 110 3.08 3.22 19.4 1

## 5 NA NA NA 18.7 NA NA 360 175 3.15 3.44 17.0 0

## 6 NA NA 18.1 NA NA NA 225 105 2.76 3.46 20.2 1

## 7 NA NA NA 14.3 NA NA 360 245 3.21 3.57 15.8 0

## 8 NA NA NA NA 24.4 NA 147. 62 3.69 3.19 20 1

## 9 NA NA NA NA 22.8 NA 141. 95 3.92 3.15 22.9 1

## 10 NA NA 19.2 NA NA NA 168. 123 3.92 3.44 18.3 1

## # … with 22 more rows, and 2 more variables: gear , carb

All the columns that start with either “0” or “1” must be pivoted:

mtcars\_wide2 %>%

pivot\_long(cols = matches("0|1"), names\_to = "am\_cyl", values\_to = "mpg", na.rm = TRUE) %>%

select(am\_cyl, everything())

## # A tibble: 32 x 10

## am\_cyl disp hp drat wt qsec vs gear carb mpg

##

## 1 1\_6 160 110 3.9 2.62 16.5 0 4 4 21

## 2 1\_6 160 110 3.9 2.88 17.0 0 4 4 21

## 3 1\_4 108 93 3.85 2.32 18.6 1 4 1 22.8

## 4 0\_6 258 110 3.08 3.22 19.4 1 3 1 21.4

## 5 0\_8 360 175 3.15 3.44 17.0 0 3 2 18.7

## 6 0\_6 225 105 2.76 3.46 20.2 1 3 1 18.1

## 7 0\_8 360 245 3.21 3.57 15.8 0 3 4 14.3

## 8 0\_4 147. 62 3.69 3.19 20 1 4 2 24.4

## 9 0\_4 141. 95 3.92 3.15 22.9 1 4 2 22.8

## 10 0\_6 168. 123 3.92 3.44 18.3 1 4 4 19.2

## # … with 22 more rows

Now, there is one new column, “am\_cyl” which must still be transformed by separating “am\_cyl” into two new columns:

mtcars\_wide2 %>%

pivot\_long(cols = matches("0|1"), names\_to = "am\_cyl", values\_to = "mpg", na.rm = TRUE) %>%

separate(am\_cyl, into = c("am", "cyl"), sep = "\_") %>%

select(am, cyl, everything())

## # A tibble: 32 x 11

## am cyl disp hp drat wt qsec vs gear carb mpg

##

## 1 1 6 160 110 3.9 2.62 16.5 0 4 4 21

## 2 1 6 160 110 3.9 2.88 17.0 0 4 4 21

## 3 1 4 108 93 3.85 2.32 18.6 1 4 1 22.8

## 4 0 6 258 110 3.08 3.22 19.4 1 3 1 21.4

## 5 0 8 360 175 3.15 3.44 17.0 0 3 2 18.7

## 6 0 6 225 105 2.76 3.46 20.2 1 3 1 18.1

## 7 0 8 360 245 3.21 3.57 15.8 0 3 4 14.3

## 8 0 4 147. 62 3.69 3.19 20 1 4 2 24.4

## 9 0 4 141. 95 3.92 3.15 22.9 1 4 2 22.8

## 10 0 6 168. 123 3.92 3.44 18.3 1 4 4 19.2

## # … with 22 more rows

It is also possible to achieve this using a data frame with the specification of what you need:

mtcars\_spec\_long <- mtcars\_wide2 %>%

pivot\_long\_spec(matches("0|1"), values\_to = "mpg") %>%

separate(name, c("am", "cyl"), sep = "\_")

mtcars\_spec\_long

## # A tibble: 6 x 4

## .name .value am cyl

##

## 1 1\_6 mpg 1 6

## 2 1\_4 mpg 1 4

## 3 0\_6 mpg 0 6

## 4 0\_8 mpg 0 8

## 5 0\_4 mpg 0 4

## 6 1\_8 mpg 1 8

Providing this spec to pivot\_long() solves the issue:

mtcars\_wide2 %>%

pivot\_long(spec = mtcars\_spec\_long, na.rm = TRUE) %>%

select(am, cyl, everything())

## # A tibble: 32 x 11

## am cyl disp hp drat wt qsec vs gear carb mpg

##

## 1 1 6 160 110 3.9 2.62 16.5 0 4 4 21

## 2 1 6 160 110 3.9 2.88 17.0 0 4 4 21

## 3 1 4 108 93 3.85 2.32 18.6 1 4 1 22.8

## 4 0 6 258 110 3.08 3.22 19.4 1 3 1 21.4

## 5 0 8 360 175 3.15 3.44 17.0 0 3 2 18.7

## 6 0 6 225 105 2.76 3.46 20.2 1 3 1 18.1

## 7 0 8 360 245 3.21 3.57 15.8 0 3 4 14.3

## 8 0 4 147. 62 3.69 3.19 20 1 4 2 24.4

## 9 0 4 141. 95 3.92 3.15 22.9 1 4 2 22.8

## 10 0 6 168. 123 3.92 3.44 18.3 1 4 4 19.2

## # … with 22 more rows

Stay tuned to Hadley Wickham’s [twitter](https://twitter.com/hadleywickham) as there will definitely  
be announcements soon!