More on Data Wrangling

Contents


```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ---------------------------------- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

0.1 Data Wrangling

- Previous verbs:
  - `select`
  - `filter`
  - `mutate`
  - `rename`
  - `arrange`
  - `summarise`
  - `group_by`
  - `count`
  - `slice`
- Many, many others exist, but we will not cover them in detail – explore on your own!
- However, we will cover a very important pair of verbs used to reshape/pivot data:
  - Long-to-wide with `pivot_wider()`
  - Wide-to-long with `pivot_longer()`

0.2 Pivoting: wide-to-long

> "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

http://www.jstatsoft.org/v59/i10/paper

A dataset in the "long" format is tidy if it follows three interrelated rules:

- Each variable must have its own column.
- Each observation must have its own row.
- Each value must have its own cell.

0.2.1 Small example going from wide to long format (and back):

Number of bald eagle nesting sites for several US regions and years:

```
# Source: US Fish and Wildlife Service and https://dcl-wrangle.stanford.edu/pivot-basic.html
eagles <- tibble(region = c("Pacific", "Southwest", "Rocky Mountains and Plains"),
                 `2007` = c(1037, 51, 200),
                 `2009` = c(2587, 176, 338))
eagles
```

```
## # A tibble: 3 x 3
##   region                     `2007` `2009`
##   <chr>                       <dbl>  <dbl>
## 1 Pacific                      1037   2587
## 2 Southwest                      51    176
## 3 Rocky Mountains and Plains    200    338
```

```
eagles_long <- eagles |>
  pivot_longer(c(`2007`, `2009`), names_to = "year", values_to = "num_nests")
eagles_long
```

```
## # A tibble: 6 x 3
##   region                     year  num_nests
##   <chr>                      <chr>     <dbl>
## 1 Pacific                    2007       1037
## 2 Pacific                    2009       2587
## 3 Southwest                  2007         51
## 4 Southwest                  2009        176
## 5 Rocky Mountains and Plains 2007        200
## 6 Rocky Mountains and Plains 2009        338
```

```
eagles_long |> pivot_wider(names_from = "year", values_from = "num_nests")
```

```
## # A tibble: 3 x 3
##   region                     `2007` `2009`
##   <chr>                       <dbl>  <dbl>
## 1 Pacific                      1037   2587
## 2 Southwest                      51    176
## 3 Rocky Mountains and Plains    200    338
```

0.2.2 Bigger data example

We use the `billboard` dataset which is part of the `tidyr` package of the `tidyverse`:

```
billboard
```

```
## # A tibble: 317 x 79
##    artist   track date.entered   wk1   wk2   wk3   wk4   wk5   wk6   wk7   wk8   wk9  wk10
##    <chr>    <chr> <date>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2 Pac    Baby~ 2000-02-26      87    82    72    77    87    94    99    NA    NA    NA
##  2 2Ge+her  The ~ 2000-09-02      91    87    92    NA    NA    NA    NA    NA    NA    NA
```

```
##  3 3 Doors~ Kryp~ 2000-04-08      81    70    68    67    66    57    54    53    51    51
##  4 3 Doors~ Loser 2000-10-21      76    76    72    69    67    65    55    59    62    61
##  5 504 Boyz Wobb~ 2000-04-15      57    34    25    17    17    31    36    49    53    57
##  6 98^0     Give~ 2000-08-19      51    39    34    26    26    19     2     2     3     6
##  7 A*Teens  Danc~ 2000-07-08      97    97    96    95   100    NA    NA    NA    NA    NA
##  8 Aaliyah  I Do~ 2000-01-29      84    62    51    41    38    35    35    38    38    36
##  9 Aaliyah  Try ~ 2000-03-18      59    53    38    28    21    18    16    14    12    10
## 10 Adams, ~ Open~ 2000-08-26      76    76    74    69    68    67    61    58    57    59
## # i 307 more rows
## # i 66 more variables: wk11 <dbl>, wk12 <dbl>, wk13 <dbl>, wk14 <dbl>, wk15 <dbl>,
## #   wk16 <dbl>, wk17 <dbl>, wk18 <dbl>, wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>,
## #   wk23 <dbl>, wk24 <dbl>, wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>,
## #   wk30 <dbl>, wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,
## #   wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>, wk43 <dbl>,
## #   wk44 <dbl>, wk45 <dbl>, wk46 <dbl>, wk47 <dbl>, wk48 <dbl>, wk49 <dbl>, ...
```

```
billboard |>
  pivot_longer(
    cols = starts_with("wk"),
    names_to = "week",
    values_to = "rank"
  )
```

```
## # A tibble: 24,092 x 5
##    artist track                   date.entered week   rank
##    <chr>  <chr>                   <date>       <chr> <dbl>
##  1 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk1      87
##  2 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk2      82
##  3 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk3      72
##  4 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk4      77
##  5 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk5      87
##  6 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk6      94
##  7 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk7      99
##  8 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk8      NA
##  9 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk9      NA
## 10 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk10     NA
## # i 24,082 more rows
```

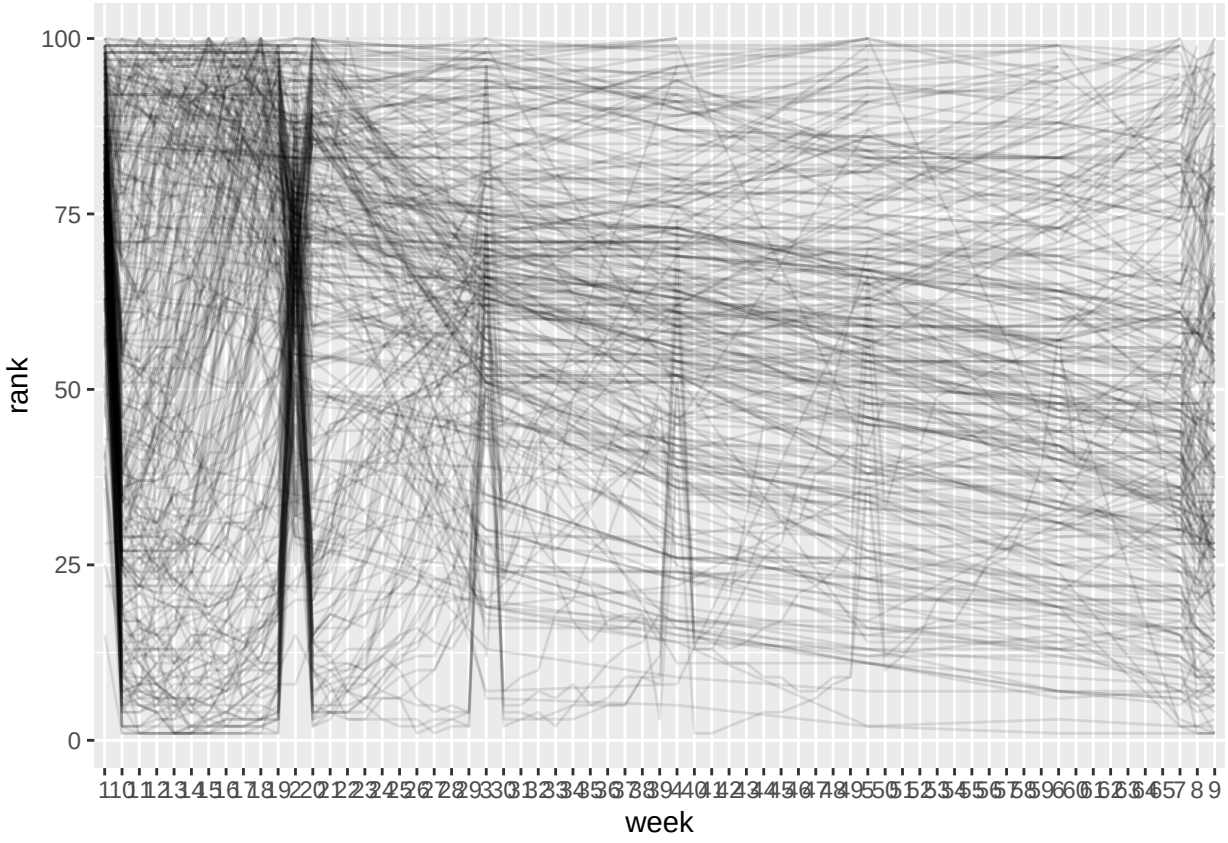
```
billboard_long <- billboard |>
  pivot_longer(
    cols = starts_with("wk"),
    names_to = "week",
    names_prefix = "wk",
    values_to = "rank",
    values_drop_na = TRUE
  )
billboard_long
```

```
## # A tibble: 5,307 x 5
##    artist  track                   date.entered week   rank
##    <chr>   <chr>                   <date>       <chr> <dbl>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26   1        87
```

```
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26   2        82
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26   3        72
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26   4        77
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26   5        87
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26   6        94
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26   7        99
##  8 2Ge+her The Hardest Part Of ... 2000-09-02   1        91
##  9 2Ge+her The Hardest Part Of ... 2000-09-02   2        87
## 10 2Ge+her The Hardest Part Of ... 2000-09-02   3        92
## # i 5,297 more rows
```

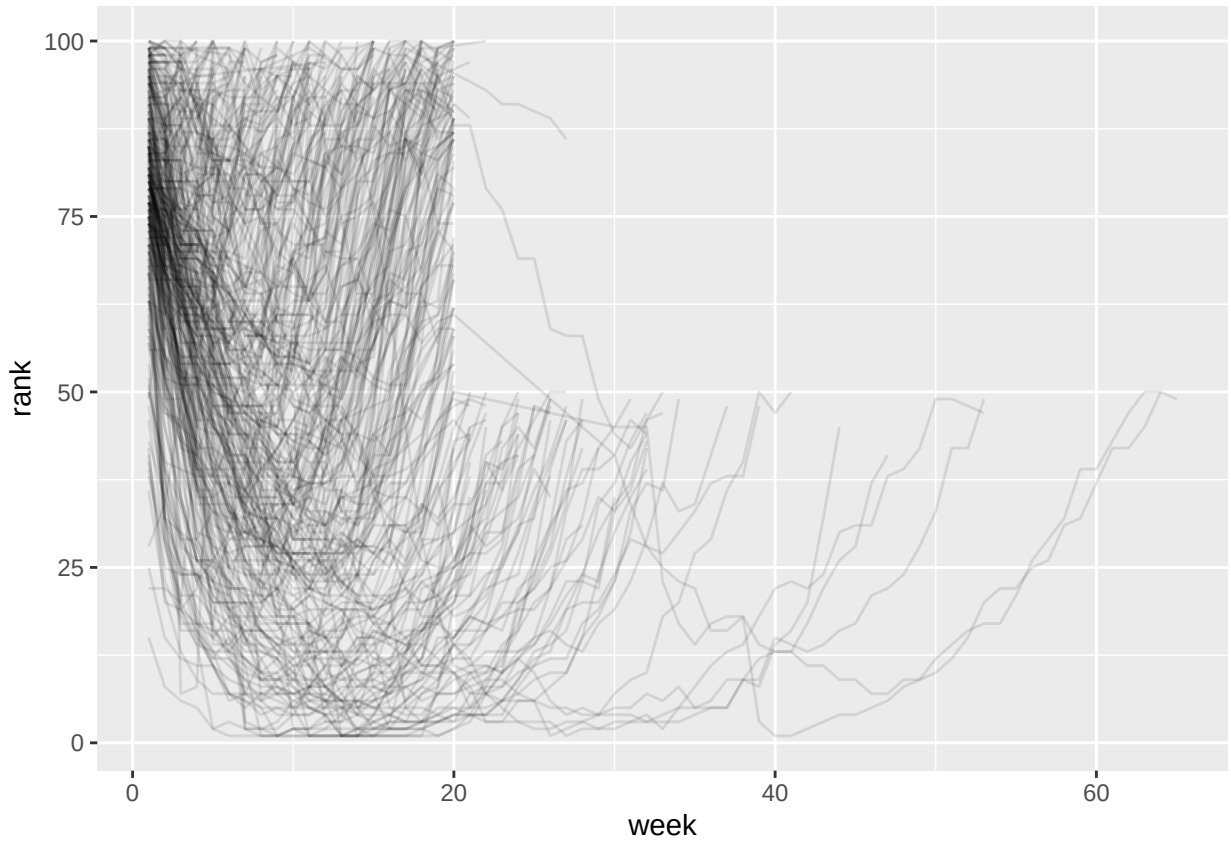
What's wrong here?

```
billboard_long |>
  ggplot(aes(x = week, y = rank, group = track)) +
  geom_line(alpha = .1)
```



The variable `week` was character and we fix it with an intermediate `mutate()` here (but it could also have been fixed by adding `names_transform = as.numeric` as an argument to `pivot_longer()` when we defined the long dataset).

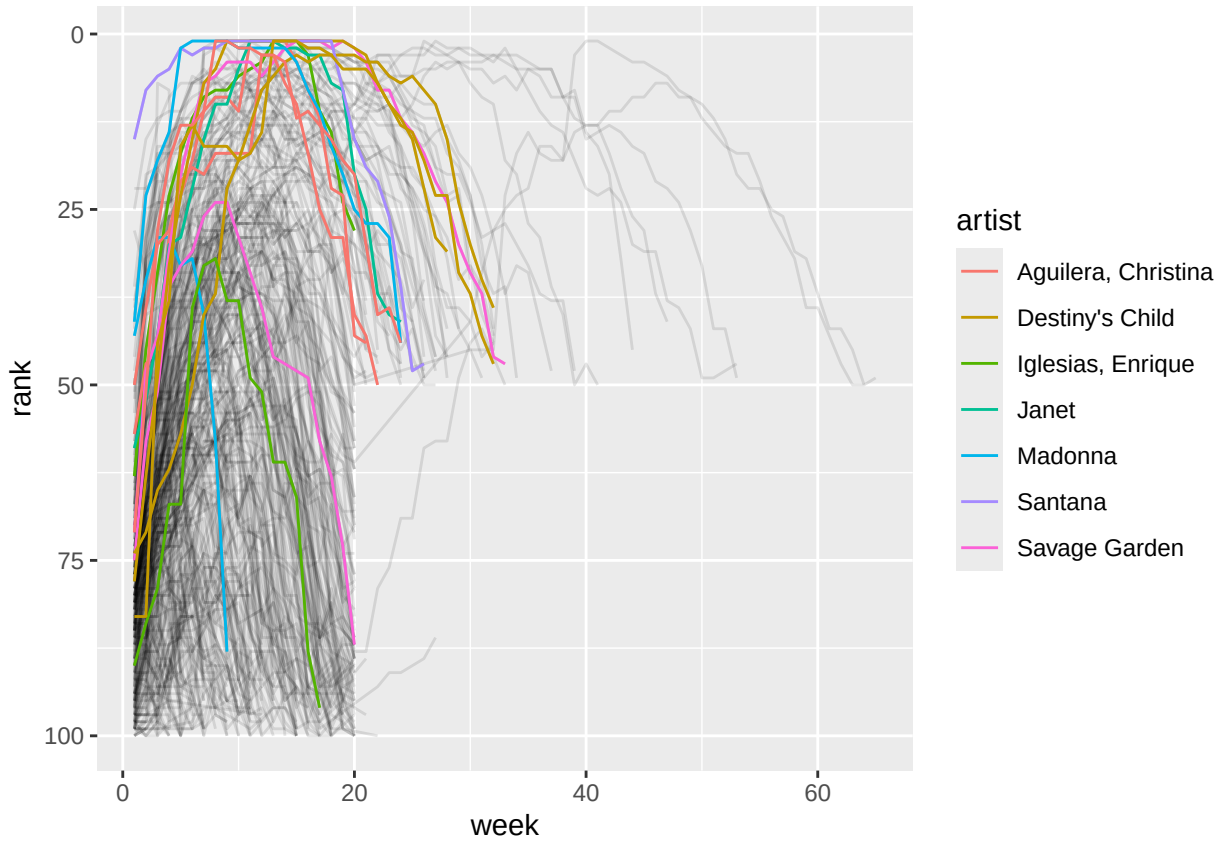
```
billboard_long <- billboard_long |> mutate(week = as.numeric(week))
billboard_long |>
  ggplot(aes(x = week, y = rank, group = track)) +
  geom_line(alpha = .1)
```




```
top_artists <- billboard_long |>
  filter(rank == 1) |>
  group_by(track,artist) |>
  count() |>
  filter(n>2) |>
  pull(artist) |>
  unique()
top_artists
```

```
## [1] "Iglesias, Enrique"   "Aguilera, Christina" "Janet"
## [4] "Savage Garden"       "Destiny's Child"     "Santana"
## [7] "Madonna"
```

```
billboard_long |>
  ggplot(aes(x = week, y = rank, group = track)) +
  geom_line(alpha = .1) +
  geom_line(aes(color = artist),
            data = billboard_long |> filter(artist %in% top_artists)) +
  scale_y_reverse()
```



0.3 Pivoting: long-to-wide

```
billboard_wide <- billboard_long |>
  pivot_wider(names_from = week, values_from = rank, names_prefix = "wk")
billboard_wide
```

```
## # A tibble: 317 x 68
##    artist   track date.entered   wk1   wk2   wk3   wk4   wk5   wk6   wk7   wk8   wk9  wk10
##    <chr>    <chr> <date>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2 Pac    Baby~ 2000-02-26      87    82    72    77    87    94    99    NA    NA    NA
##  2 2Ge+her  The ~ 2000-09-02      91    87    92    NA    NA    NA    NA    NA    NA    NA
##  3 3 Doors~ Kryp~ 2000-04-08      81    70    68    67    66    57    54    53    51    51
##  4 3 Doors~ Loser 2000-10-21      76    76    72    69    67    65    55    59    62    61
##  5 504 Boyz Wobb~ 2000-04-15      57    34    25    17    17    31    36    49    53    57
##  6 98^0     Give~ 2000-08-19      51    39    34    26    26    19     2     2     3     6
##  7 A*Teens  Danc~ 2000-07-08      97    97    96    95   100    NA    NA    NA    NA    NA
##  8 Aaliyah  I Do~ 2000-01-29      84    62    51    41    38    35    35    38    38    36
##  9 Aaliyah  Try ~ 2000-03-18      59    53    38    28    21    18    16    14    12    10
## 10 Adams, ~ Open~ 2000-08-26      76    76    74    69    68    67    61    58    57    59
## # i 307 more rows
## # i 55 more variables: wk11 <dbl>, wk12 <dbl>, wk13 <dbl>, wk14 <dbl>, wk15 <dbl>,
## #   wk16 <dbl>, wk17 <dbl>, wk18 <dbl>, wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>,
## #   wk23 <dbl>, wk24 <dbl>, wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>,
## #   wk30 <dbl>, wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,
## #   wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>, wk43 <dbl>,
## #   wk44 <dbl>, wk45 <dbl>, wk46 <dbl>, wk47 <dbl>, wk48 <dbl>, wk49 <dbl>, ...
```

0.4 Cheat sheet

https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-transformation.pdf:

See others at https://www.rstudio.com/resources/cheatsheets/ (or at https://github.com/rstudio/cheatsheets/).

0.5 Exercises

- Confront the qmd exercise on Moodle.