Lecture 25: Restructuring data

Taylor Arnold

gathering data

In some cases we have datasets where multiple columns could be treated as individual observations. What does that mean? Think of the cancer dataset we used earlier in the semester, taking off just the cancer incidence rates:

```
cancer <- read_csv("https://statsmaths.github.io/stat_data/cance
cancer <- select(cancer, breast, colorectal, prostate, lung,
                 melanoma)
```

gathering data

```
cancer
```

```
## # A tibble: 1,961 x 5
##   breast colorectal prostate  lung melanoma
##    <dbl>      <dbl>    <dbl> <dbl>    <dbl>
## 1  127.8       40.9    113.0  61.5     14.5
## 2  133.2       39.2     82.1  58.1     13.1
## 3  101.6       36.9    117.9  35.1     13.7
## 4  127.4       41.0     96.5  64.9     12.3
## 5  119.8       37.5    121.4  69.7     15.0
## 6  150.8       46.2    110.6  74.9     25.4
## # ... with 1,955 more rows
```

gathering data

A common question with this data is: how we could plot all of the cancer types on the same plot. The canonical way of doing this would be to make a new dataset where each row, instead of being a single county, is then just one incidence rate. In other words each county will have five rows associated with it. To do this we use the `gather` function. It is found in the package **tidyr**.

gathering data

```r
library(tidyr)
gather(cancer)
```

```
## # A tibble: 9,805 x 2
##      key value
##    <chr> <dbl>
## 1 breast 127.8
## 2 breast 133.2
## 3 breast 101.6
## 4 breast 127.4
## 5 breast 119.8
## 6 breast 150.8
## # ... with 9,799 more rows
```

gathering data

If we want to give the name of the `key' andvalue', those are given as the next two parameters to gather:

```
gather(cancer, type, incidence)
```

```
## # A tibble: 9,805 x 2
##     type incidence
##    <chr>     <dbl>
## 1 breast     127.8
## 2 breast     133.2
## 3 breast     101.6
## 4 breast     127.4
## 5 breast     119.8
## 6 breast     150.8
## # ... with 9,799 more rows
```

gathering data

In many cases, there are other variables that we want to be duplicated along with the other keys. For example, take the speed skating dataset, selecting off a few variables to make it a bit more tractable:

```
speed <- read_csv("https://statsmaths.github.io/stat_data/speed_
speed <- select(speed, num_skater, nationality, time_lap1,
                time_lap2, time_lap3,
                time_lap4, time_lap5)
```

gathering data

We can indicate the variables that should not be gathered by including them after the key and value terms with minus signs:

```
gather(speed, skater, value, -num_skater, -nationality)
```

```
## # A tibble: 23,520 x 4
##   num_skater nationality    skater value
##        <int>       <chr>     <chr> <dbl>
## 1        175         RUS time_lap1  6.91
## 2        100         CAN time_lap1  6.69
## 3        138         ITA time_lap1  6.87
## 4        182         TUR time_lap1  7.30
## 5         92         AUS time_lap1  7.00
## 6        190         USA time_lap1  6.69
## # ... with 2.351e+04 more rows
```

gathering data

Or, we can not use the minus sign and include on those variables that should be gathered:

```r
gather(speed, skater, value, time_lap1, time_lap2,
        time_lap3, time_lap4, time_lap5)
```

```
## # A tibble: 23,520 x 4
##   num_skater nationality    skater value
##        <int>       <chr>     <chr> <dbl>
## 1        175         RUS time_lap1  6.91
## 2        100         CAN time_lap1  6.69
## 3        138         ITA time_lap1  6.87
## 4        182         TUR time_lap1  7.30
## 5         92         AUS time_lap1  7.00
## 6        190         USA time_lap1  6.69
## # ... with 2.351e+04 more rows
```

gathering data

The results are the same; in most cases one or the other will lead to less typing. There is also a complement to gathering called spreading, available through the function `spread`. You should not need that in this class because every dataset is maximally spread already. It is also a lot more difficult to use because you have to be careful about implicit missing values and what to do with them.