Getting / cleaning data 2

More with `dplyr`

dplyr

So far, you've used several dplyr functions:

- rename
- filter
- select
- mutate
- group_by
- summarize

Some other useful dplyr functions to add to your toolbox are:

- separate and unite
- mutate and other dplyr functions with group_by
- anti_join and semi_join

separate

Sometimes, you want to take one column and split it into two columns. For example, you may have information for two variables in one column:

```
ebola
```

```
## # A tibble: 4 x 1
##   ebola_key
##   <chr>
## 1 Liberia_Cases
## 2 Liberia_Deaths
## 3 Spain_Cases
## 4 Spain_Deaths
```

separate

If you have a consistent "split" character, you can use the `separate` function to split one column into two:

```
ebola %>%
  separate(col = ebola_key, into = c("country", "outcome"),
           sep = "_")
```

```
## # A tibble: 4 x 2
##   country outcome
##   <chr>   <chr>
## 1 Liberia Cases
## 2 Liberia Deaths
## 3 Spain   Cases
## 4 Spain   Deaths
```

separate

Here is the generic code for separate:

```
## Generic code
separate([dataframe],
         col = [name of the single column you want to split],
         into = [vector of the names of the columns
                 you want to create],
         sep = [the character that designates where
                you want to split])
```

separate

The default is to drop the original column and only keep the columns into which it was split. However, you can use the argument `remove = FALSE` to keep the first column, as well:

```
ebola %>%
  separate(col = ebola_key, into = c("country", "outcome"),
           sep = "_", remove = FALSE)
```

```
## # A tibble: 4 x 3
##   ebola_key      country outcome
##   <chr>          <chr>   <chr>
## 1 Liberia_Cases  Liberia Cases
## 2 Liberia_Deaths Liberia Deaths
## 3 Spain_Cases    Spain   Cases
## 4 Spain_Deaths   Spain   Deaths
```

separate

You can use the `fill` argument (`fill = "right"` or `fill = "left"`) to control what happens when there are some observations that do not have the split character.

For example, say your original column looked like this:

```
## # A tibble: 4 x 1
##   ebola_key
##   <chr>
## 1 Liberia_Cases
## 2 Liberia
## 3 Spain_Cases
## 4 Spain_Deaths
```

separate

You can use `fill = "right"` to set how to split observations like the second one, where there is no separator character ("_"):

```
ebola %>%
  separate(col = ebola_key, into = c("country", "outcome"),
           sep = "_", fill = "right")
```

```
## # A tibble: 4 x 2
##   country outcome
##   <chr>   <chr>
## 1 Liberia Cases
## 2 Liberia <NA>
## 3 Spain   Cases
## 4 Spain   Deaths
```