The `arrange` verb orders a dataset by a given variable. It's useful for the display of results, primarily.

In RStudio on your computer, run the following code chunk to arrange the `five_name` data set by the variable `max`:

```
by_country <- group_by(tb_long, country)
five_num <- summarise(by_country,
                      min = min(count, na.rm = TRUE),
                      first_quartile = quantile(count, 0.25, na.rm = TRUE),
                      median = median(count, na.rm = TRUE),
                      third_quartile = quantile(count, 0.75, na.rm = TRUE),
                      max = max(count, na.rm = TRUE)
)

arrange(five_num, max)
```

```
## # A tibble: 219 x 6
##    country                       min first_quartile median third_quartile   max
##    <chr>                       <dbl>          <dbl>  <dbl>          <dbl> <dbl>
##  1 Bonaire, Saint Eustatius an~     0              0      0              0     0
##  2 Tokelau                          0              0      0              0     0
##  3 Anguilla                         0              0      0              0     1
##  4 Bermuda                          0              0      0              0     1
##  5 British Virgin Islands           0              0      0              0     1
##  6 Monaco                           0              0      0              0     1
##  7 Montserrat                       0              0      0              0     1
##  8 Niue                             0              0      0              0     1
##  9 San Marino                       0              0      0              0     1
## 10 US Virgin Islands                0              0      0              1     1
## # ... with 209 more rows
```

Results are sorted in ascending order by default. To change the sorting order, wrap the **`desc()`** function around the variable you wish to sort your data set by.

```
arrange(five_num, desc(max))
```

```
## # A tibble: 219 x 6
##    country                       min first_quartile median third_quartile   max
##    <chr>                       <dbl>          <dbl>  <dbl>          <dbl> <dbl>
##  1 India                          11          1949. 14451          46034. 90830
##  2 China                           0          6390. 17800          27372  70376
##  3 Indonesia                       0           828   4020.         13502  25460
##  4 South Africa                    0          1174.  2355          11185. 22620
##  5 Pakistan                        0           317   1917           6162  15445
##  6 Philippines                     0           110.  3206.          5700. 14474
##  7 Bangladesh                      0           609.  3048           6448. 13378
##  8 Brazil                          0           382.  2192           4400. 11906
##  9 Democratic Republic of the ~  137          1220.  2449           5885. 10412
## 10 Nigeria                         0           505.  1198.          3107  10382
## # ... with 209 more rows
```

# Give it a go!

Continue to develop your wrangling skills by working through the following exercise - arranging the tuberculosis (TB) data. If you haven't already, make sure you download the tidy long form TB data (https://github.com/datascienceprogram/ids_course_data/blob/master/tb_long.rds) and store it in your project data folder e.g. **first_project/data/tb_long.rds**. Once you have done this, load the collection of `tidyverse` packages, read the TB long form data into your R session, then:

- **sort the results in descending order, for example, alphabetically**
- **arrange the dataset by the first 10 countries**.

Within the **Comments**, share with other learners your results from the exercise.

---