cm009 Exercises: tidy data

```
suppressPackageStartupMessages(library(tidyverse))
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
## Warning: package 'tidyr' was built under R version 3.4.4
## Warning: package 'dplyr' was built under R version 3.4.4
```

Reading and Writing Data: Exercises

Make a tibble of letters, their order in the alphabet, and then a pasting of the two columns together.

Make a tibble of three names and commute times.

Write the `iris` data frame as a `csv`.

Write the `iris` data frame to a file delimited by a dollar sign.

Read the dollar-delimited `iris` data to a tibble.

Read these three LOTR csv's, saving them to `lotr1`, `lotr2`, and `lotr3`:

- https://raw.githubusercontent.com/jennybc/lotr-tidy/master/data/The_Fellowship_Of_The_Ring.csv
- https://raw.githubusercontent.com/jennybc/lotr-tidy/master/data/The_Two_Towers.csv
- https://github.com/jennybc/lotr-tidy/blob/master/data/The_Return_Of_The_King.csv

`gather()`

(Exercises largely based off of Jenny Bryan's gather tutorial)

This function is useful for making untidy data tidy (so that computers can more easily crunch the numbers).

1. Combine the three LOTR untidy tables (`lotr1`, `lotr2`, `lotr3`) to a single untidy table by stacking them.
2. Convert to tidy. Also try this by specifying columns as a range, and with the `contains()` function.
3. Try again (bind and tidy the three untidy data frames), but without knowing how many tables there are originally.
    - The additional work here does not require any additional tools from the tidyverse, but instead uses a `do.call` from base R – a useful tool in data analysis when the number of "items" is variable/unknown, or quite large.

`spread()`

(Exercises largely based off of Jenny Bryan's spread tutorial)

This function is useful for making tidy data untidy (to be more pleasing to the eye).

Read in the tidy LOTR data (despite having just made it):

```
lotr_tidy <- read_csv("https://raw.githubusercontent.com/jennybc/lotr-tidy/master/data/lotr_tidy.csv")
```

```
## Parsed with column specification:
## cols(
##   Film = col_character(),
##   Race = col_character(),
##   Gender = col_character(),
##   Words = col_integer()
## )
```

Get word counts across "Race". Then try "Gender".

Now try combining race and gender. Use `unite()` from `tidyr` instead of `paste()`.

Other `tidyr` goodies

Check out the Examples in the documentation to explore the following.

`expand` vs `complete` (trim vs keep everything). Together with `nesting`. Check out the Examples in the `expand` documentation.

`separate_rows`: useful when you have a variable number of entries in a "cell".

`unite` and `separate`.

`uncount` (as the opposite of `dplyr::count()`)

`drop_na` and `replace_na`

`fill`

`full_seq`

Time remaining?

Time permitting, do this exercise to practice tidying data.