CREATING TABLES USING `mmtable2` PACKAGE

HUMPHREY KINOTI

`mmtable2` Tutorial

To install, run: `remotes::install_github("ianmoran11/mmtable2")`
Let's get up and running with `mmtable2` so we can make a killer table that impresses your bosses and helps you make reports that get you promoted.

Goal for our table

Our goal is to analyze the `mpg` dataset (fuel economy of vehicles by important vehicle attributes like manufacturer, number of cylinders, etc). The table we put to into our report:

- Summarizes the average fuel economy (City and Highway)
- By two categories: Car Manufacturer and Number of Engine Cylinders

Step 1: Load the Libraries and Data

First, we need to:

- Load Libraries: Load `mmtable2` , `gt`, and `tidyverse`.
- Import Data: We're using the `mpg` dataset that comes with `ggplot2`.

```
library(mmtable2)
library(tidyverse)
library(gt)

data(mpg)
```

Step 2: Tidy the Data

Our next step is to use `dplyr` and `tidyr` to get the data into the right format for the table. We'll use 4 important data wrangling operations:

- `group_by()`: Groups by our grouping columns: Manufacturer and Number of Engine Cylinders.
- `summarise()`: We'll calculate the average fuel economy for both City and Highway. We combine with the across() function which makes it easy to summarize multiple columns. We use the mean() function to calculate the averages by group.
- `ungroup()`: Ungrouping is needed to remove any leftover groups.
- `pivot_longer()`: Used to convert from a "wide" to a "long" data frame, which stacks the City and Highway average fuel economy on top of each other. If you're familiar with ggplot2 the "long" format is critical to plotting.

Step 3: Make the basic table

With the mpg data summarized and in the long format, we can now use `mmtable2` to make a table, just like we would use `ggplot2` to make a plot. We perform 3 actions:

1. Setup the `mmtable()`: This is just like `ggplot()` function in `ggplot2`.
2. Specify the headers locations: This tells the location for each header needed to organize the table.
3. Format the header and table cells: This adds the lines that help to differentiate groups in our data.

BONUS: Customize the table with `gt`

The magic of `mmtable2` is that it actually uses another awesome package called `gt`, which is what allows `mmtable2` to produce awesome-looking tables.

So, if we know how to use gt, we can customize our basic table!

Let's give it a go by adding some gt headers, which give the table a title and subtitle.