Project 1 Fish McDowell

Data is everywhere. It's power is immeasurable with finding patterns, modeling relationships, and driving decisions. In order to be able to do that, data must be handled appropriately. In this report, we will go through the motions of loading in and preprocessing some data so that it's true power can be used as discussed above.

Question 1: Selecting Columns

First, we will load in the appropriate data set and select only `Area_name`, `STCOU`, and any columns that end with the letter D, as this is the only information we need. We will also lower case the `Area_name` variable.

```
sec1 <- read_csv("./data/EDU01a.csv", col_names = TRUE)
```

```
Rows: 3198 Columns: 42
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (22): Area_name, STCOU, EDU010187N1, EDU010187N2, EDU010188N1, EDU010188...
dbl (20): EDU010187F, EDU010187D, EDU010188F, EDU010188D, EDU010189F, EDU010...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sec1_new <- sec1 |>
  select(area_name = Area_name,
         STCOU,
         ends_with("D"))

head(sec1_new, n = 5)
```

```
# A tibble: 5 x 12
  area_name     STCOU EDU010187D EDU010188D EDU010189D EDU010190D EDU010191D
  <chr>         <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
1 UNITED STATES 00000   40024299   39967624   40317775   40737600   41385442
2 ALABAMA       01000     733735     728234     730048     728252     725541
3 Autauga, AL   01001       6829       6900       6920       6847       7008
4 Baldwin, AL   01003      16417      16465      16799      17054      17479
5 Barbour, AL   01005       5071       5098       5068       5156       5173
# i 5 more variables: EDU010192D <dbl>, EDU010193D <dbl>, EDU010194D <dbl>,
#   EDU010195D <dbl>, EDU010196D <dbl>
```

The selected columns look to be what we hoped, with all that aren't `area_name` and `STCOU` end with "D".

Question 2: Long Formatted Data

Next, we will convert this data into long format with only one row per enrollment value for that area name. We will put the column names into a separate new variable to keep that information.

```
sec1_long <- sec1_new |>
  pivot_longer(cols = 3:12,
               names_to = "survey_type",
               values_to = "vals")

head(sec1_long, n = 5)
```

```
# A tibble: 5 x 4
  area_name     STCOU survey_type     vals
  <chr>         <chr> <chr>          <dbl>
1 UNITED STATES 00000 EDU010187D  40024299
2 UNITED STATES 00000 EDU010188D  39967624
3 UNITED STATES 00000 EDU010189D  40317775
4 UNITED STATES 00000 EDU010190D  40737600
5 UNITED STATES 00000 EDU010191D  41385442
```

This looks to match the pivot that we hoped to make.

Question 3: Further Splitting Data

As above, we notice that one of the new columns (labeled survey_type) corresponds to the old column names that end with "D". We know that the information in this column represents multiple pieces of information. Namely, the first 3 characters represent the survey, the next 4 represent the value type, and the last 2 digits represent the year of measurement. Knowing this information, we will now parse through those strings and create a new variable with the numeric date represented as YYYY. We will also do that with the first 3 and remaining 4 characters in the string.

```
long_updated <- sec1_long |>
  mutate(
    year = as.numeric(paste0("19", substr(sec1_long$survey_type, 8, 9))),
    survey = substr(sec1_long$survey_type, 1, 3),
    val_type = substr(sec1_long$survey_type, 4, 7)
  )

head(long_updated, n = 5)
```

```
# A tibble: 5 x 7
  area_name     STCOU survey_type     vals  year survey val_type
  <chr>         <chr> <chr>          <dbl> <dbl> <chr>  <chr>
1 UNITED STATES 00000 EDU010187D  40024299  1987 EDU    0101
2 UNITED STATES 00000 EDU010188D  39967624  1988 EDU    0101
3 UNITED STATES 00000 EDU010189D  40317775  1989 EDU    0101
4 UNITED STATES 00000 EDU010190D  40737600  1990 EDU    0101
5 UNITED STATES 00000 EDU010191D  41385442  1991 EDU    0101
```

Looking at the head of this data set, we have split the `survey_type` variable into the 3 separate pieces of information that is represents.

Question 4: Splitting Into County and Non-County Data