Project 1

Define libraries needed

```
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
```

Data Processing

First steps: Read in the data

```
census_data <- read_csv("EDU01a.csv")
```

Question 1: Select only the columns: Area_name, STCOU, and any column that ends in "D". Only display the first 5 rows

```
census_data_2 <- census_data |>
  select(Area_name, STCOU, ends_with("D")) |>
  rename(area_name = Area_name)

head(census_data_2, 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>
```

Question 2: Convert out data into long format where each row has only one enrollment value for that Area_name.

```
census_data_3 <- census_data_2 |>
  pivot_longer(cols = ends_with("D"), names_to = "code_year",
  values_to = "total_enrolled"
  )

head(census_data_3, 5)
```

```
# A tibble: 5 x 4
  area_name     STCOU code_year  total_enrolled
  <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
```

Question 3: Parse the string to pull out the year and convert the year into a numeric value. Grab the first three characters and following four digits to create a new variable representing which measurement was grabbed.

```
long_updated <- census_data_3 |>
  mutate(year = 1900 + as.numeric(substr(code_year, 8, 9)),
         new_measure = substr(code_year, 1, 7))

head(long_updated, 5)
```

```
# A tibble: 5 x 6
  area_name     STCOU code_year  total_enrolled  year new_measure
```

```
  <chr>         <chr> <chr>               <dbl> <dbl> <chr>
1 UNITED STATES 00000 EDU010187D       40024299  1987 EDU0101
2 UNITED STATES 00000 EDU010188D       39967624  1988 EDU0101
3 UNITED STATES 00000 EDU010189D       40317775  1989 EDU0101
4 UNITED STATES 00000 EDU010190D       40737600  1990 EDU0101
5 UNITED STATES 00000 EDU010191D       41385442  1991 EDU0101
```

Question 4: Create two data sets: one containing only non-count data, one containing only county level data

```
split_data <- grep(pattern = ", \\w\\w", long_updated$area_name)

county_tibble <- long_updated |>
  slice(split_data)
class(county_tibble) <- c("county", class(county_tibble))

state_tibble <- long_updated |>
  slice(-split_data)
class(state_tibble) <- c("state", class(state_tibble))

head(county_tibble, 10)
```

```
# A tibble: 10 x 6
   area_name   STCOU code_year  total_enrolled  year new_measure
   <chr>       <chr> <chr>               <dbl> <dbl> <chr>
 1 Autauga, AL 01001 EDU010187D           6829  1987 EDU0101
 2 Autauga, AL 01001 EDU010188D           6900  1988 EDU0101
 3 Autauga, AL 01001 EDU010189D           6920  1989 EDU0101
 4 Autauga, AL 01001 EDU010190D           6847  1990 EDU0101
 5 Autauga, AL 01001 EDU010191D           7008  1991 EDU0101
 6 Autauga, AL 01001 EDU010192D           7137  1992 EDU0101
 7 Autauga, AL 01001 EDU010193D           7152  1993 EDU0101
 8 Autauga, AL 01001 EDU010194D           7381  1994 EDU0101
 9 Autauga, AL 01001 EDU010195D           7568  1995 EDU0101
10 Autauga, AL 01001 EDU010196D           7834  1996 EDU0101
```

```
head(state_tibble, 10)
```

```
# A tibble: 10 x 6
   area_name     STCOU code_year  total_enrolled  year new_measure
```

```
   <chr>         <chr> <chr>               <dbl> <dbl> <chr>
 1 UNITED STATES 00000 EDU010187D       40024299  1987 EDU0101
 2 UNITED STATES 00000 EDU010188D       39967624  1988 EDU0101
 3 UNITED STATES 00000 EDU010189D       40317775  1989 EDU0101
 4 UNITED STATES 00000 EDU010190D       40737600  1990 EDU0101
 5 UNITED STATES 00000 EDU010191D       41385442  1991 EDU0101
 6 UNITED STATES 00000 EDU010192D       42088151  1992 EDU0101
 7 UNITED STATES 00000 EDU010193D       42724710  1993 EDU0101
 8 UNITED STATES 00000 EDU010194D       43369917  1994 EDU0101
 9 UNITED STATES 00000 EDU010195D       43993459  1995 EDU0101
10 UNITED STATES 00000 EDU010196D       44715737  1996 EDU0101
```

Question 5: Creating a new variable for the county level tibble that describes which state one of the county measurements corresponds to

```
new_county_tibble <- county_tibble |>
  mutate(State = substr(area_name, start = nchar(area_name) -1 ,
                        stop = nchar(area_name)))

head(new_county_tibble, 5)
```

```
# A tibble: 5 x 7
  area_name   STCOU code_year  total_enrolled  year new_measure State
  <chr>       <chr> <chr>               <dbl> <dbl> <chr>       <chr>
1 Autauga, AL 01001 EDU010187D           6829  1987 EDU0101     AL
2 Autauga, AL 01001 EDU010188D           6900  1988 EDU0101     AL
3 Autauga, AL 01001 EDU010189D           6920  1989 EDU0101     AL
4 Autauga, AL 01001 EDU010190D           6847  1990 EDU0101     AL
5 Autauga, AL 01001 EDU010191D           7008  1991 EDU0101     AL
```

Question 6: Creating a new variable called "divison" for the non-county tibble that corresponds to the state's classification of division

```
noncounty_tibble_new <- state_tibble |>
  mutate(division = case_when (
    area_name %in% c("CONNECTICUT", "MAINE", "MASSACHUSETTS",
                     "NEW HAMPSHIRE", "RHODE ISLAND", "VERMONT")
    ~ "New England",
```

```
      area_name %in% c("NEW JERSEY", "NEW YORK", "PENNSYLVANIA")
    ~ "Mid-Atlantic",
      area_name %in% c("ILLINOIS", "INDIANA", "MICHIGAN", "OHIO",
                       "WISCONSIN") ~ "East North Central",
      area_name %in% c("IOWA", "KANSAS", "MINNESOTA", "MISSOURI",
                       "NEBRASKA", "NORTH DAKOTA", "SOUTH DAKOTA")
    ~ "West North Central",
      area_name %in% c("DELAWARE", "FLORIDA", "GEORGIA", "MARYLAND",
                       "NORTH CAROLINA", "SOUTH CAROLINA", "VIRGINIA",
                       "WEST VIRGINIA", "DISTRICT OF COLUMBIA")
    ~ "South Atlantic",
      area_name %in% c("ALABAMA", "KENTUCKY", "MISSISSIPPI",
                       "TENNESSEE") ~ "East South Central",
      area_name %in% c("ARKANSAS", "LOUISIANA", "OKLAHOMA", "TEXAS")
    ~ "West South Central",
      area_name %in% c("ARIZONA", "COLORADO", "IDAHO", "MONTANA",
                       "NEVADA", "NEW MEXICO", "UTAH", "WYOMING") ~ "Mountain",
      area_name %in% c("ALASKA", "CALIFORNIA", "HAWAII", "OREGON", "WASHINGTON")
    ~ "Pacific",
      TRUE ~ "ERROR"))

head(noncounty_tibble_new, 5)
```

```
# A tibble: 5 x 7
  area_name     STCOU code_year  total_enrolled  year new_measure division
  <chr>         <chr> <chr>               <dbl> <dbl> <chr>       <chr>
1 UNITED STATES 00000 EDU010187D       40024299  1987 EDU0101     ERROR
2 UNITED STATES 00000 EDU010188D       39967624  1988 EDU0101     ERROR
3 UNITED STATES 00000 EDU010189D       40317775  1989 EDU0101     ERROR
4 UNITED STATES 00000 EDU010190D       40737600  1990 EDU0101     ERROR
5 UNITED STATES 00000 EDU010191D       41385442  1991 EDU0101     ERROR
```

Requirements

Reading in data

```
census_data_b <- read_csv("EDU01b.csv")
```

Function 1: Steps 1 and 2

```
func_1 <- function(df, value_name = "students_enrolled") {
  df_1 <- df |>
    select(
      area_name = Area_name,
      STCOU,
      ends_with("D")
    )
  df_long <- df_1 |>
    pivot_longer(
      cols = ends_with("D"),
      names_to = "code_year",
      values_to = "total_enrolled"
    )
  return(df_long)
}
```

Function 2: Step 3

```
func_2 <- function(df_long){
  long_updated <- df_long |>
    mutate(year = 1900 + as.numeric(substr(code_year, 8, 9)),
         new_measure = substr(code_year, 1, 7))
  return(long_updated)
}
```

Function 3: Step 5

```
func_3 <- function(county_tibble){
  county_tibble |>
    mutate(State = substr(area_name, start = nchar(area_name) -1 ,
                          stop = nchar(area_name)))
}
```

Function 4: Step 6

```
func_4 <- function(state_tibble){
   state_tibble |>
    mutate(division = case_when (
    area_name %in% c("CONNECTICUT", "MAINE", "MASSACHUSETTS", "NEW HAMPSHIRE",
                     "RHODE ISLAND", "VERMONT") ~ "New England",
      area_name %in% c("NEW JERSEY", "NEW YORK", "PENNSYLVANIA") ~
      "Mid-Atlantic",
      area_name %in% c("ILLINOIS", "INDIANA", "MICHIGAN", "OHIO", "WISCONSIN")
      ~"East North Central",
      area_name %in% c("IOWA", "KANSAS", "MINNESOTA", "MISSOURI", "NEBRASKA",
                       "NORTH DAKOTA", "SOUTH DAKOTA") ~ "West North Central",
      area_name %in% c("DELAWARE", "FLORIDA", "GEORGIA", "MARYLAND",
                       "NORTH CAROLINA", "SOUTH CAROLINA", "VIRGINIA",
                       "WEST VIRGINIA", "DISTRICT OF COLUMBIA")
      ~ "South Atlantic",
      area_name %in% c("ALABAMA", "KENTUCKY", "MISSISSIPPI", "TENNESSEE")
      ~ "East South Central",
      area_name %in% c("ARKANSAS", "LOUISIANA", "OKLAHOMA", "TEXAS")
      ~ "West South Central",
      area_name %in% c("ARIZONA", "COLORADO", "IDAHO", "MONTANA", "NEVADA",
                       "NEW MEXICO", "UTAH", "WYOMING") ~ "Mountain",
      area_name %in% c("ALASKA", "CALIFORNIA", "HAWAII", "OREGON", "WASHINGTON")
      ~ "Pacific",
      TRUE ~ "ERROR"))
}
```

Function 5

```
func_5 <- function(long_updated){
  split_data <- grep(pattern = ", \\w\\w", long_updated$area_name)

  county_tibble <- long_updated |>
  slice(split_data) |>
    func_3()
class(county_tibble) <- c("county", class(county_tibble))

state_tibble <- long_updated |>
  slice(-split_data) |>
```

```
  func_4()
class(state_tibble) <- c("state", class(state_tibble))

return(list(county_tibble, state_tibble))
}
```

Wrapper Function

```
my_wrapper <- function(url, default_var_name = "students_enrolled"){
df <-  read_csv(url)

df_long <- func_1(df)

long_updated <- func_2(df_long)

result <- func_5(long_updated)

return(result)
}
```

Use wrapper function for both data files

```
census_a <- my_wrapper("EDU01a.csv")

census_b <- my_wrapper("EDU01b.csv")
```

Function to combine tibbles

```
combine_function <- function(wrapper_a, wrapper_b){

  combined_county <- bind_rows(wrapper_a[[1]], wrapper_b[[1]])

  combined_state <- bind_rows(wrapper_a[[2]], wrapper_b[[2]])

return(list(county = combined_county, state = combined_state))
}
```

```
combine_function(census_a, census_b)
```

```
$county
# A tibble: 62,900 x 7
   area_name   STCOU code_year  total_enrolled  year new_measure State
   <chr>       <chr> <chr>               <dbl> <dbl> <chr>       <chr>
 1 Autauga, AL 01001 EDU010187D           6829  1987 EDU0101     AL
 2 Autauga, AL 01001 EDU010188D           6900  1988 EDU0101     AL
 3 Autauga, AL 01001 EDU010189D           6920  1989 EDU0101     AL
 4 Autauga, AL 01001 EDU010190D           6847  1990 EDU0101     AL
 5 Autauga, AL 01001 EDU010191D           7008  1991 EDU0101     AL
 6 Autauga, AL 01001 EDU010192D           7137  1992 EDU0101     AL
 7 Autauga, AL 01001 EDU010193D           7152  1993 EDU0101     AL
 8 Autauga, AL 01001 EDU010194D           7381  1994 EDU0101     AL
 9 Autauga, AL 01001 EDU010195D           7568  1995 EDU0101     AL
10 Autauga, AL 01001 EDU010196D           7834  1996 EDU0101     AL
# i 62,890 more rows

$state
# A tibble: 1,060 x 7
   area_name     STCOU code_year  total_enrolled  year new_measure division
   <chr>         <chr> <chr>               <dbl> <dbl> <chr>       <chr>
 1 UNITED STATES 00000 EDU010187D       40024299  1987 EDU0101     ERROR
 2 UNITED STATES 00000 EDU010188D       39967624  1988 EDU0101     ERROR
 3 UNITED STATES 00000 EDU010189D       40317775  1989 EDU0101     ERROR
 4 UNITED STATES 00000 EDU010190D       40737600  1990 EDU0101     ERROR
 5 UNITED STATES 00000 EDU010191D       41385442  1991 EDU0101     ERROR
 6 UNITED STATES 00000 EDU010192D       42088151  1992 EDU0101     ERROR
 7 UNITED STATES 00000 EDU010193D       42724710  1993 EDU0101     ERROR
 8 UNITED STATES 00000 EDU010194D       43369917  1994 EDU0101     ERROR
 9 UNITED STATES 00000 EDU010195D       43993459  1995 EDU0101     ERROR
10 UNITED STATES 00000 EDU010196D       44715737  1996 EDU0101     ERROR
# i 1,050 more rows
```

Summarizing Functions

Writing a Generic Function for Summarizing non county tibble

```r
plot.state <- function(df, var_name = "total_enrolled") {
  new_df <- df |>
    filter(division != "ERROR")

  mean_df <- new_df |>
    group_by(division, year) |>
    summarize(mean_stat = mean(get(var_name), na.rm = TRUE))
  ggplot(mean_df, aes(x = year, y = mean_stat, color = division)) +
    geom_line() + geom_point() +
    labs(title = "Mean of Enrollment by Division (Non-County)")}
```

Generic function for the county tibble

```r
plot.county <- function(df, var_name = "total_enrolled",
            state_of_interest = "NC", top_bottom = "top", num_top_bottom = 5) {

  df_state <- df |>
    filter(State == state_of_interest)

  county_mean <- df_state |>
    group_by(area_name) |>
    summarize(mean_val = mean(get(var_name), na.rm = TRUE))

  if(top_bottom == "top") {
    select_county <- county_mean |>
      arrange(desc(mean_val)) |>
      head(num_top_bottom) |>
      pull(area_name)
  } else if (top_bottom == "bottom") {
    select_county <- county_mean |>
      arrange(mean_val) |>
      head(num_top_bottom) |>
      pull(area_name)
  }
  plotting_data <- df_state |>
    filter(area_name %in% select_county)

  ggplot(plotting_data, aes(x = year, y = .data[[var_name]],
                            color = area_name)) + geom_line() + geom_point() +
```

```
    labs(title = "Enrollment Over Time")
}
```

Put It Together

Run your data processing function on the two enrollment URLs given previously, specifying an appropriate name for the enrollment data column.

```
census_a <- my_wrapper("EDU01a.csv")

census_b <- my_wrapper("EDU01b.csv")
```

Run your data combining function to put these into one object (with two data frames)

```
combine_function(census_a, census_b)[[1]]
```

```
# A tibble: 62,900 x 7
   area_name   STCOU code_year  total_enrolled  year new_measure State
   <chr>       <chr> <chr>               <dbl> <dbl> <chr>       <chr>
 1 Autauga, AL 01001 EDU010187D           6829  1987 EDU0101     AL
 2 Autauga, AL 01001 EDU010188D           6900  1988 EDU0101     AL
 3 Autauga, AL 01001 EDU010189D           6920  1989 EDU0101     AL
 4 Autauga, AL 01001 EDU010190D           6847  1990 EDU0101     AL
 5 Autauga, AL 01001 EDU010191D           7008  1991 EDU0101     AL
 6 Autauga, AL 01001 EDU010192D           7137  1992 EDU0101     AL
 7 Autauga, AL 01001 EDU010193D           7152  1993 EDU0101     AL
 8 Autauga, AL 01001 EDU010194D           7381  1994 EDU0101     AL
 9 Autauga, AL 01001 EDU010195D           7568  1995 EDU0101     AL
10 Autauga, AL 01001 EDU010196D           7834  1996 EDU0101     AL
# i 62,890 more rows
```

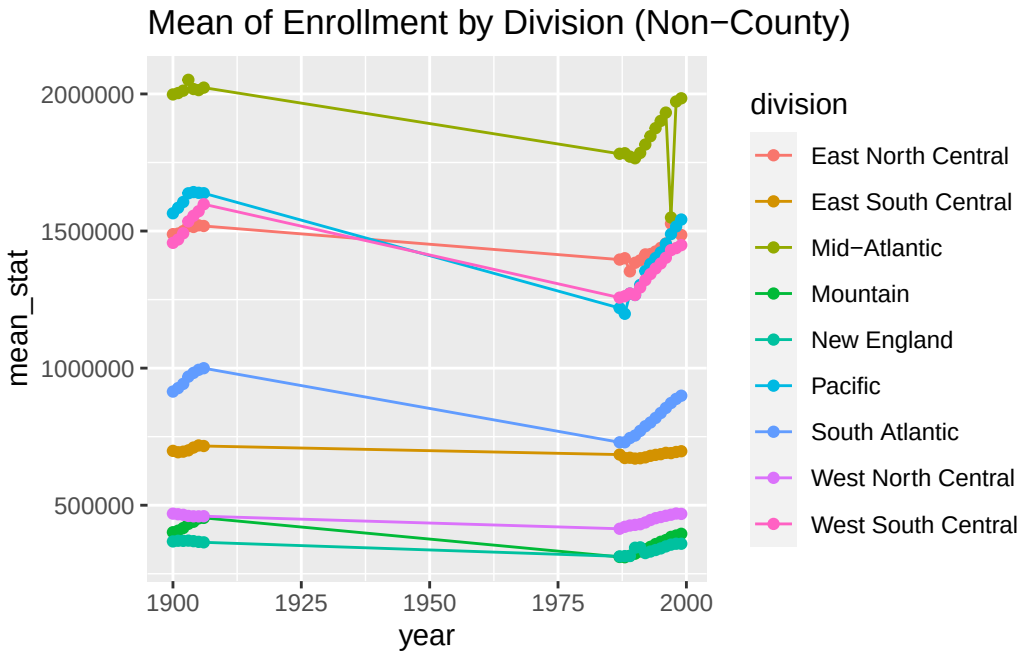
```
combine_function(census_a, census_b)[[2]]
```

```
# A tibble: 1,060 x 7
   area_name     STCOU code_year  total_enrolled  year new_measure division
   <chr>         <chr> <chr>               <dbl> <dbl> <chr>       <chr>
 1 UNITED STATES 00000 EDU010187D       40024299  1987 EDU0101     ERROR
```

```
 2 UNITED STATES 00000 EDU010188D       39967624  1988 EDU0101     ERROR
 3 UNITED STATES 00000 EDU010189D       40317775  1989 EDU0101     ERROR
 4 UNITED STATES 00000 EDU010190D       40737600  1990 EDU0101     ERROR
 5 UNITED STATES 00000 EDU010191D       41385442  1991 EDU0101     ERROR
 6 UNITED STATES 00000 EDU010192D       42088151  1992 EDU0101     ERROR
 7 UNITED STATES 00000 EDU010193D       42724710  1993 EDU0101     ERROR
 8 UNITED STATES 00000 EDU010194D       43369917  1994 EDU0101     ERROR
 9 UNITED STATES 00000 EDU010195D       43993459  1995 EDU0101     ERROR
10 UNITED STATES 00000 EDU010196D       44715737  1996 EDU0101     ERROR
# i 1,050 more rows
```

Use the plot function on the state data frame

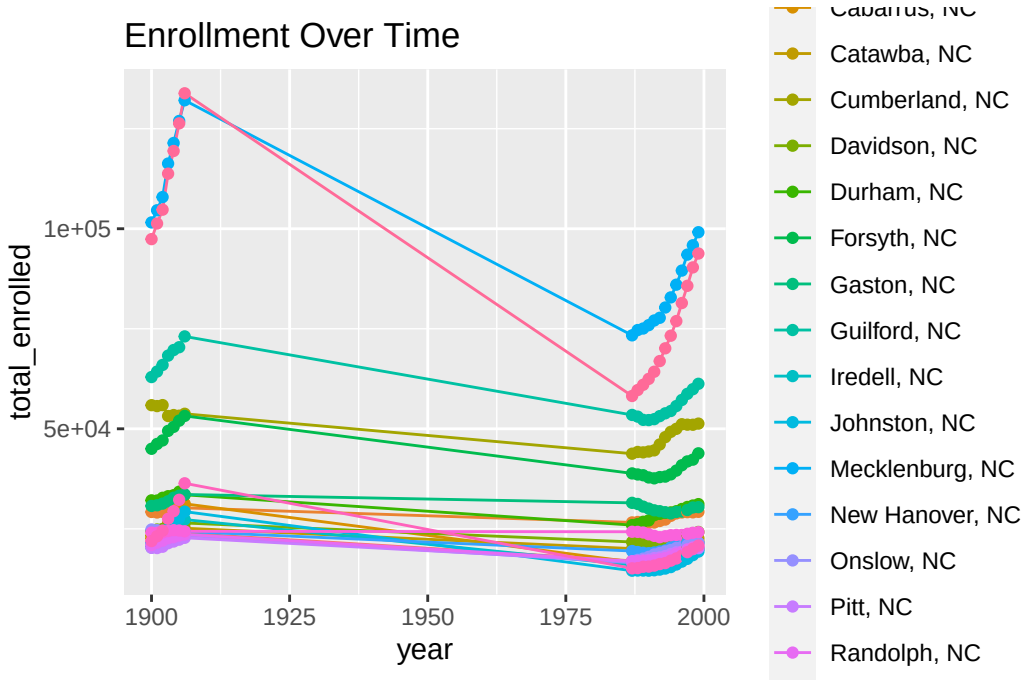
```
plot.state(combine_function(census_a, census_b)[[2]])
```



Use the plot function on the county data frame

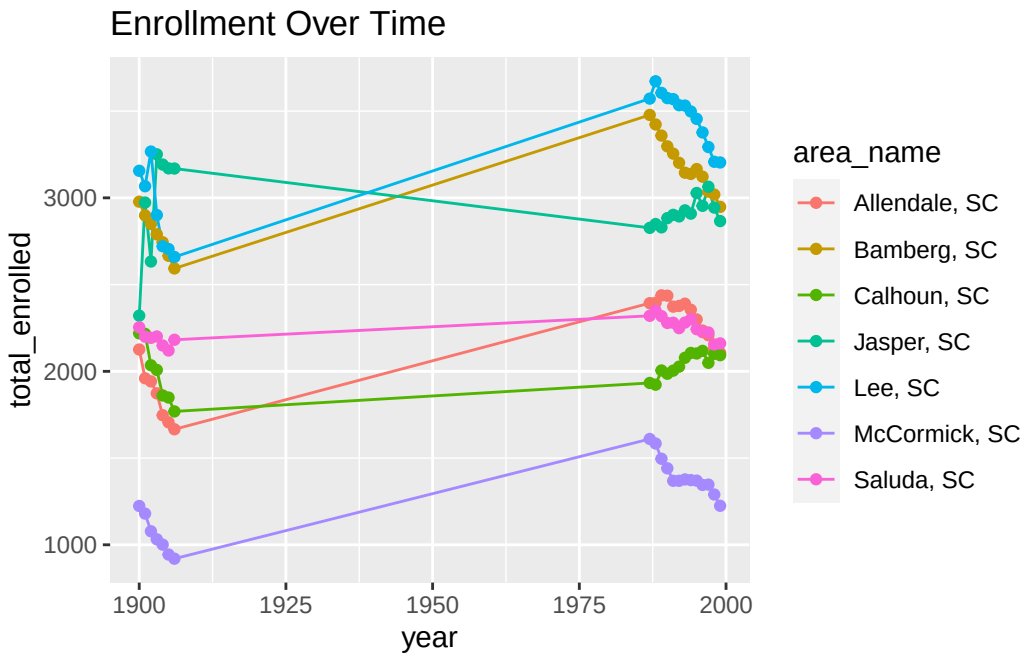
Once specifying the state to be "NC", the group being the top, the number looked at being 20

```
plot.county(combine_function(census_a, census_b)[[1]], num_top_bottom = 20)
```



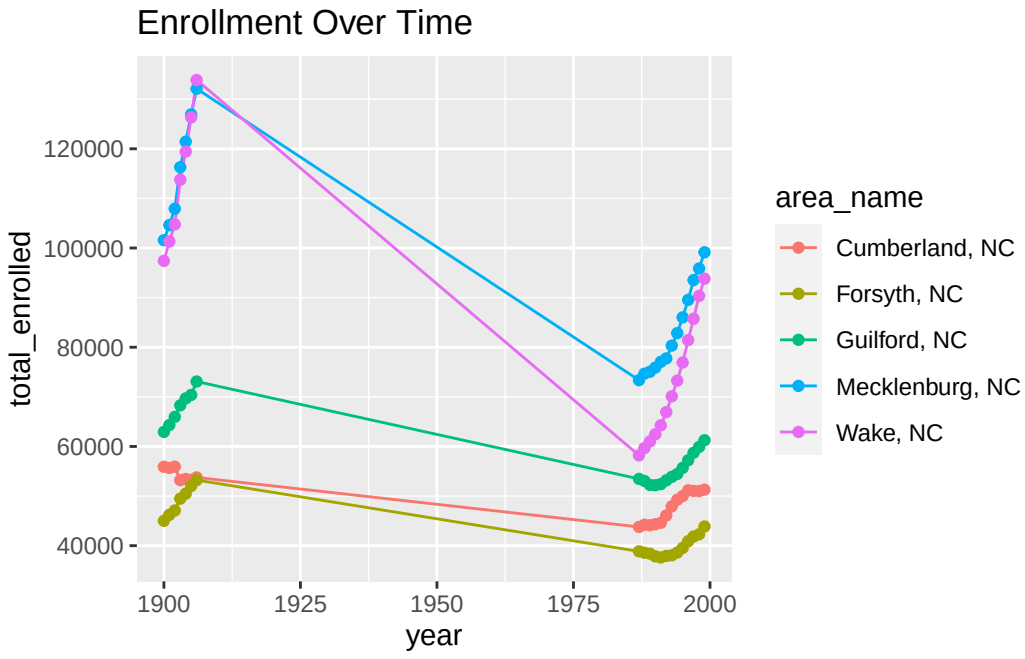
Once specifying the state to be "SC", the group being the bottom, the number looked at being 7

```
plot.county(combine_function(census_a, census_b)[[1]], num_top_bottom = 7,
            state_of_interest = "SC", top_bottom = "bottom")
```



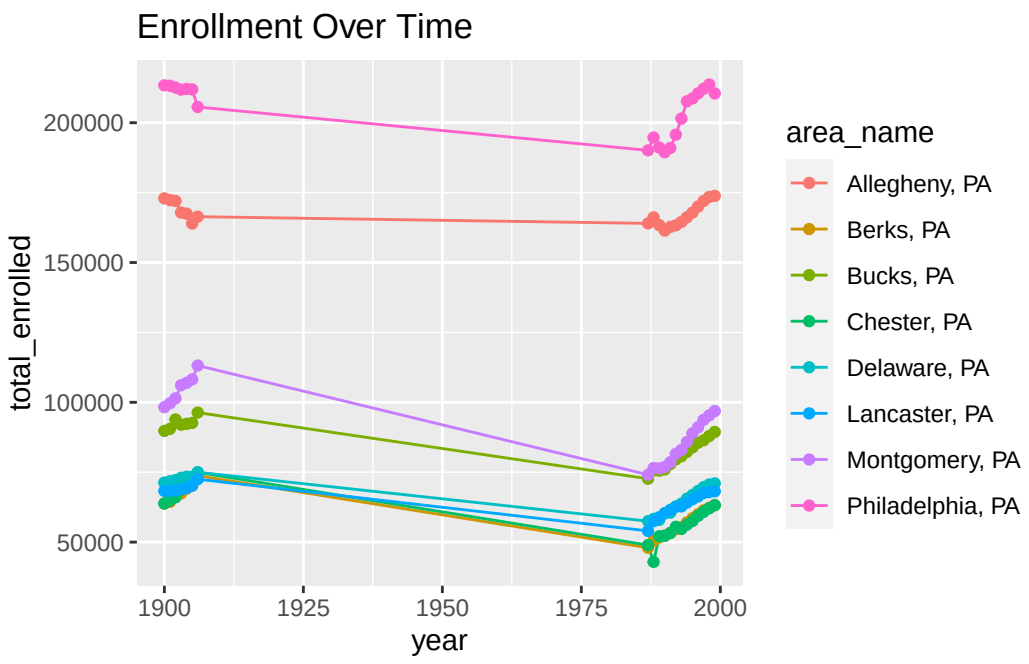
Once without specifying anything (defaults used)

```
plot.county(combine_function(census_a, census_b)[[1]])
```



Once specifying the state to be "PA", the group being the top, the number looked at being 8

```
plot.county(combine_function(census_a, census_b)[[1]], num_top_bottom = 8,
            state_of_interest = "PA")
```



Applying Features to New Data Sets

Run your data processing function on the four data sets at URLs given below:

```
PST_data_a <- my_wrapper("PST01a.csv")

PST_data_b <- my_wrapper("PST01b.csv")

PST_data_c <- my_wrapper("PST01c.csv")

PST_data_d <- my_wrapper("PST01d.csv")
```

Run your data combining function (probably three times) to put these into one object (with two data frames)

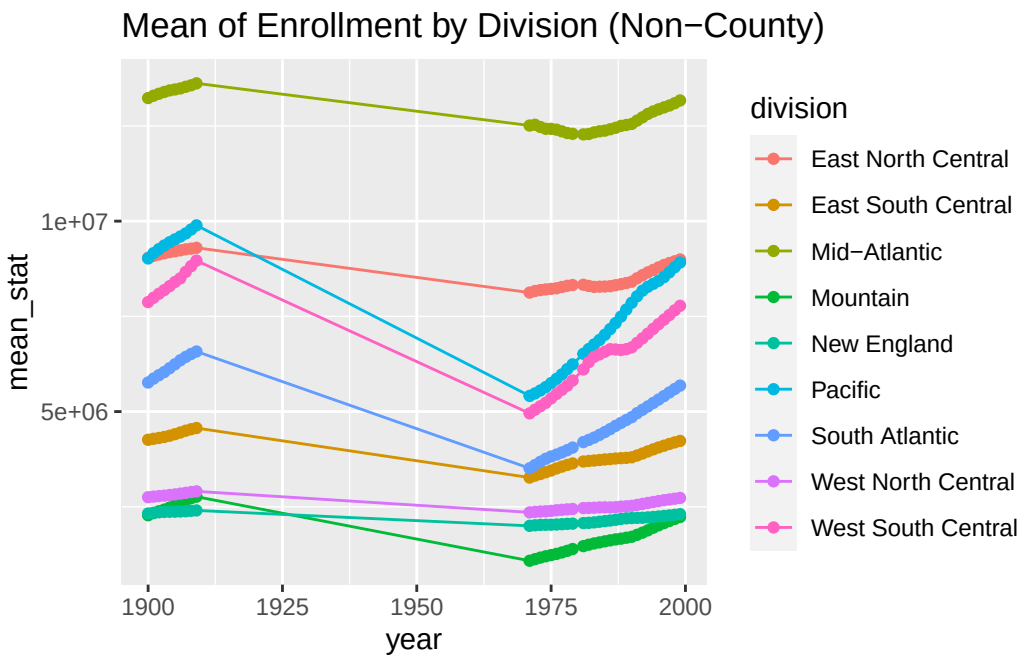
```
PST_data_ab <- combine_function(PST_data_a, PST_data_b)

PST_data_abc <- combine_function(PST_data_ab, PST_data_c)

PST_data_abcd <- combine_function(PST_data_abc, PST_data_d)
```

Use the plot function on the state data frame

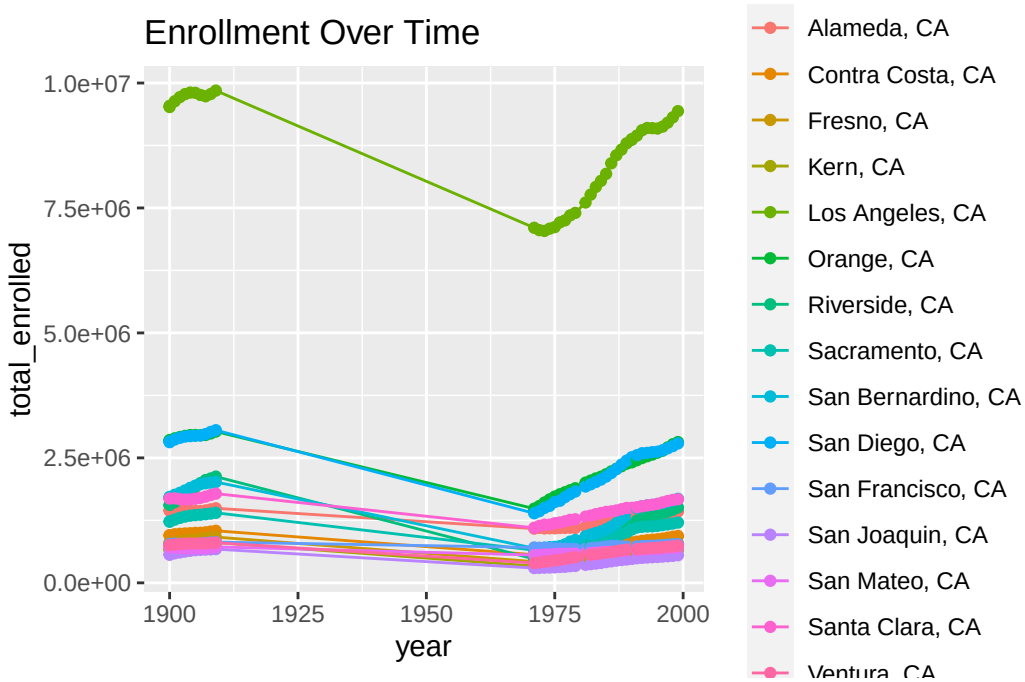
```
plot.state(PST_data_abcd[[2]])
```



Use the plot function on the county data frame

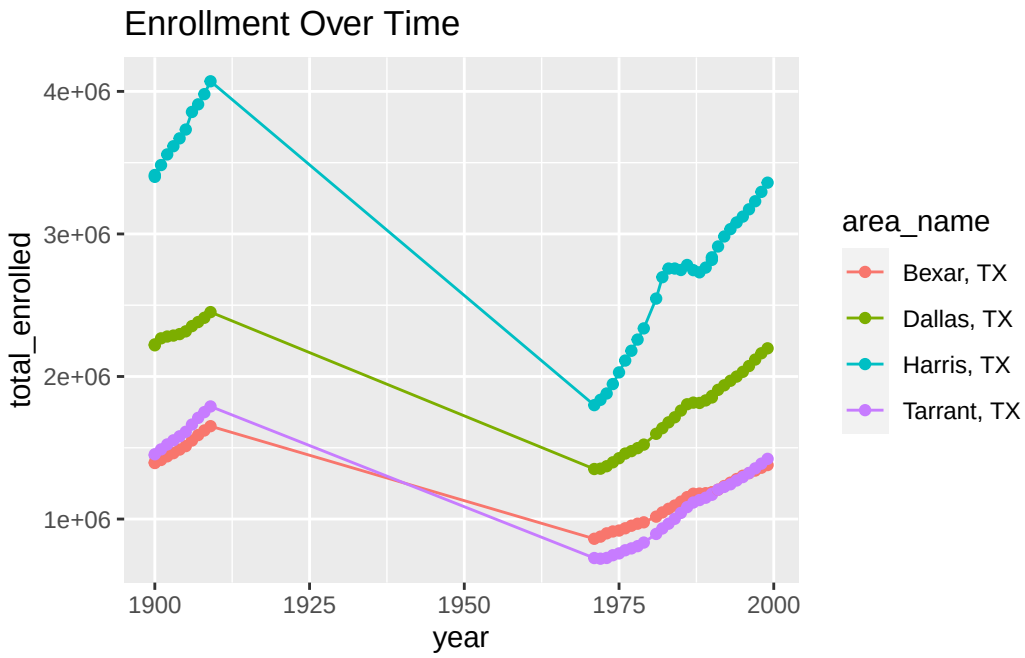
Once specifying the state to be "CA", the group being the top, the number looked at being 15

```
plot.county(PST_data_abcd[[1]], state_of_interest = "CA", num_top_bottom = 15)
```



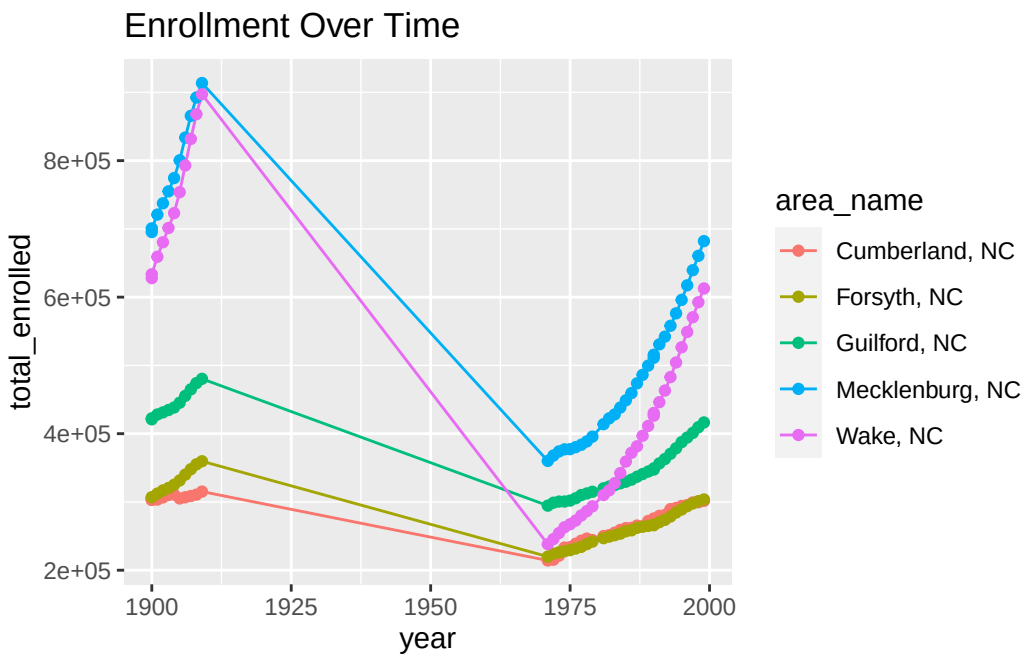
Once specifying the state to be "TX", the group being the top, the number looked at being 4

```
plot.county(PST_data_abcd[[1]], state_of_interest = "TX", num_top_bottom = 4)
```



Once without specifying anything (defaults used)

```
plot.county(PST_data_abcd[[1]])
```



Once specifying the state to be "NY", the group being the top, the number looked at being 10

```
plot.county(PST_data_abcd[[1]], state_of_interest = "NY", num_top_bottom = 10)
```