

Project 1

Keshav Ramesh & Calista Harris

Load require packages

```
library(tidyverse)
```

Data Processing

Question 1: Reading in Data

```
edu01a <- read_csv("EDU01a.csv", show_col_types = FALSE) |>
  select(
    area_name = Area_name, #rename Area_name
    STCOU,
    ends_with("D") #select all columns ending in "D"
  )

#display the first 5 rows
edu01a |>
  slice(1:5)
```

```
# 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: Pivot Data

```
edu_long <- edu01a %>%
  pivot_longer(
    cols = ends_with("D"),
    names_to = "surveyID_full", #store original column names (ex. "EST1234D")
    values_to = "enrollment"
  )

#display the first 5 rows
head(edu_long, 5)
```

```
# A tibble: 5 x 4
  area_name      STCOU surveyID_full enrollment
  <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: Extracting the year

```
long_updated = edu_long |>
  mutate(
    #extract the 2-digit year from the 8th and 9th characters of surveyID_full
    surveyID_year = substr(surveyID_full, 8, 9),

    #convert the 2-digit year into numeric
    year = as.numeric(surveyID_year),
    # if 2 year digit is greater than 80 add 1900 + year, else 2000 + year
    year = ifelse(year >= 80, 1900 + year, 2000 + year),
    surveyID = substr(surveyID_full, 1, 7)
  ) |>
  select(-surveyID_year)

#display the first 5 rows
head(long_updated, 5)
```

```
# A tibble: 5 x 6
  area_name      STCOU surveyID_full enrollment   year surveyID
  <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: Identifying County Data

```
#identify county rows: ", XX" (where XX is a two-letter state abbreviation)
county_indices <- grep(pattern = ", \\w\\w", long_updated$area_name)

#create county tibble and assign custom classes
county_tibble <- long_updated[county_indices, ]
class(county_tibble) <- c("county", class(county_tibble))

#create non-county tibble and assign custom classes
state_tibble <- long_updated[-county_indices, ]
class(state_tibble) <- c("state", class(state_tibble))

#display the first 10 rows for both data sets
head(county_tibble, 10)
```

```
# A tibble: 10 x 6
  area_name      STCOU surveyID_full enrollment   year surveyID
  <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 surveyID_full enrollment   year surveyID
  <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: Add state Variable to the County Tibble

```
county_tibble <- county_tibble |>
  mutate(
    #use nchar to get the last 2 characters of area_name
    state = substr(area_name, nchar(area_name) - 1, nchar(area_name))
  )

#display the first 5 rows
county_tibble |>
  slice(1:5)
```

```
# A tibble: 5 x 7
  area_name      STCOU surveyID_full enrollment   year surveyID 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: Add division Variable to the Non-county Tibble

```
state_tibble <- state_tibble %>%
  mutate(
    state = substr(area_name, nchar(area_name) - 1, nchar(area_name)),
    division = case_when(
      state %in% c("CT", "ME", "MA", "NH", "RI", "VT") ~ "New England",
      state %in% c("NJ", "NY", "PA") ~ "Mid-Atlantic",

      state %in% c("IL", "IN", "MI", "OH", "WI") ~ "East North Central",
      state %in% c("IA", "KS", "MN", "MO", "NE",
                   "ND", "SD") ~ "West North Central",

      state %in% c("DE", "DC", "FL", "GA", "MD", "NC",
                   "SC", "VA", "WV") ~ "South Atlantic",
      state %in% c("AL", "KY", "MS", "TN") ~ "East South Central",
      state %in% c("AR", "LA", "OK", "TX") ~ "West South Central",

      state %in% c("AZ", "CO", "ID", "MT", "NV",
                   "NM", "UT", "WY") ~ "Mountain",
      state %in% c("AK", "CA", "HI", "OR", "WA") ~ "Pacific",

      TRUE ~ "ERROR" #return error for non-states like "UNITED STATES"
    )
  ) |>
  #remove the temporary intermediate column
  select(-state)

#display the first 5 rows
state_tibble |>
  slice(1:5)
```

A tibble: 5 x 7

	area_name	STCOU	surveyID_full	enrollment	year	surveyID	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: Repeating Process with 2nd Component of Data Set

Create a Function for Steps 1 and 2

```
#read in the data set
edu01b <- read_csv("EDU01b.csv", show_col_types = FALSE)

select_pivot <- function(data, column = "enrollment") {
  data |>
    #step 1
    select(
      area_name = Area_name,
      STCOU,
      ends_with("D")
    ) |>
    #step 2
    pivot_longer(
      cols = ends_with("D"),
      names_to = "surveyID_full",
      values_to = column
    )
}
```

Create a Function for Taking Output of Step 2 and Step 3

```
extract_year_id <- function(data) {
  data |>
    mutate(
      surveyID_year = substr(surveyID_full, 8, 9),
      year = as.numeric(surveyID_year),
      year = ifelse(year >= 90, 1900 + year, 2000 + year),
      surveyID = substr(surveyID_full, 1, 7)
    ) |>
    select(-surveyID_year)
}
```

Create a Function for Step 5

```
#only to be used for the county tibble
extract_state <- function(county_tbl){
  county_tbl |>
    mutate(
      state = substr(area_name, nchar(area_name) - 1, nchar(area_name))
    )
}
```

Create a Function for Step 6

```
#only to be used for the non-county (state) tibble
assign_division <- function(state_tbl){
  state_tbl |>
    mutate(
      state = substr(area_name, nchar(area_name) - 1, nchar(area_name)),
      division = case_when(
        state %in% c("CT", "ME", "MA", "NH", "RI", "VT") ~ "New England",
        state %in% c("NJ", "NY", "PA") ~ "Mid-Atlantic",

        state %in% c("IL", "IN", "MI", "OH", "WI") ~ "East North Central",
        state %in% c("IA", "KS", "MN", "MO", "NE",
                     "ND", "SD") ~ "West North Central",

        state %in% c("DE", "DC", "FL", "GA", "MD", "NC",
                     "SC", "VA", "WV") ~ "South Atlantic",
        state %in% c("AL", "KY", "MS", "TN") ~ "East South Central",
        state %in% c("AR", "LA", "OK", "TX") ~ "West South Central",

        state %in% c("AZ", "CO", "ID", "MT", "NV",
                     "NM", "UT", "WY") ~ "Mountain",
        state %in% c("AK", "CA", "HI", "OR", "WA") ~ "Pacific",

        TRUE ~ "ERROR"
      )
    ) |>
    select(-state)
}
```

Create a Function Returning Two Final Tibbles

```
identify_locations <- function(data) {  
  #step 4  
  county_indices <- grep(pattern = ", \\w\\w", long_updated$area_name)  
  
  county_tibble <- long_updated[county_indices, ]  
  class(county_tibble) <- c("county", class(county_tibble))  
  
  state_tibble <- long_updated[-county_indices, ]  
  class(state_tibble) <- c("state", class(state_tibble))  
  
  #step 5 using the functions create  
  county_tibble <- extract_state(county_tibble)  
  state_tibble <- assign_division(state_tibble)  
  
  #return both tibbles as a list  
  return(list(county = county_tibble, state = state_tibble))  
}
```

Create All Into One Function Call - Wrapper Function

```
my_wrapper <- function(url, column = "enrollment") {  
  result <- read_csv(url, show_col_types = FALSE) |>  
    select_pivot() |>  
    extract_year_id() |>  
    identify_locations()  
  return(result)  
}
```

Call It and Combine Your Data

```
#call wrapper twice for the 2 data sets  
edu01a_parsed <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01a.csv")  
edu01b_parsed <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01b.csv")  
  
combine_data <- function(data1, data2){  
  combined_county <- bind_rows(data1$county, data2$county)
```



```

combined_state <- bind_rows(data1$state, data2$state)

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

#test and display using the combine function
combine_data(edu01a_parsed, edu01b_parsed)

```

\$county

A tibble: 62,900 x 7

	area_name	STCOU	surveyID_full	enrollment	year	surveyID	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	surveyID_full	enrollment	year	surveyID	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

Writing a Generic Function for Summarizing

Custom Plot for State Level Data

```
plot.state <- function(data1, column = "enrollment"){
  data1 |>
    filter(division != "ERROR") |> # remove any row with error
    group_by(division, year) |> # group data
    summarise(mean_value = mean(get(column), na.rm = T)) |> # summarize data to get means
  # plot data
  ggplot(aes(
    x = year,
    y = mean_value,
    color = division
  )) + # add aesthetic markers
  geom_line(linewidth = 1) + #connect dots with a line
  geom_point() +
  labs(
    title = paste0("Mean Enrollment by Division and Year"),
    x = "Year",
    y = paste0("Mean ", column),
    color = "Division"
  ) +
  guides(color = guide_legend(ncol = 2)) +
  theme_minimal() +
  theme(legend.position = "bottom")
}
```

Custom Plot for County Level Data

```
plot.county <- function(data,
                        column = "enrollment",
                        state = "NC", direction = "top",
                        n = 5){
  state_data = data |>
    filter({{state}} == state) ## filter data for specified state

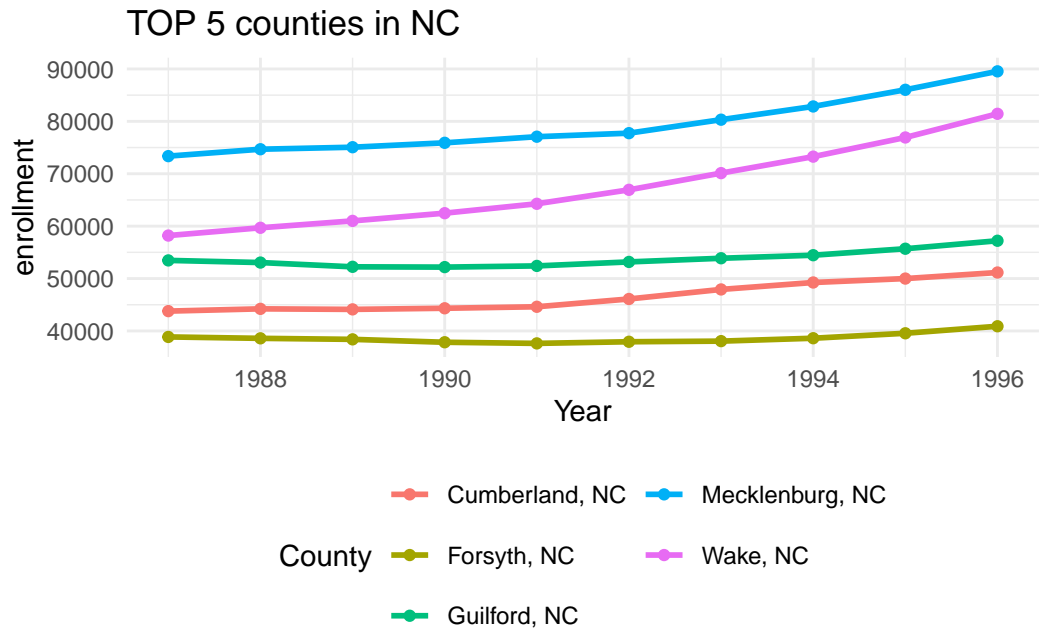
  data1 = state_data |> ## group by area name (county)
    group_by(area_name) |>
```

```

    summarise(mean_value = mean(get(column), na.rm = T))# calculate column mean
# sort counties based on direction
if (direction == "top") {
  data1 <- data1 |>
    arrange(desc(mean_value))
} else {
  data1 <- data1 |>
    arrange(mean_value)
}
# get names of top/bottom N countries
areas <- data1 %>%
  slice_head(n = n) %>%
  pull(area_name)
# Plot data
data_plotted = state_data |>
  filter(area_name %in% areas) |>
  ggplot(aes(
    x = year,
    y = get(column),
    color = area_name
  )) + # add aesthetic arguments
  geom_line(linewidth = 1) +
  geom_point() +
  labs(
    title = paste(toupper(direction), n, "counties in", state),
    x = "Year",
    y = column,
    color = "County"
  ) +
  guides(color = guide_legend(ncol = 2)) +
  theme_minimal() +
  theme(legend.position = "bottom")
return(data_plotted)
}

plot(edu01a_parsed$county)

```



Put it Together

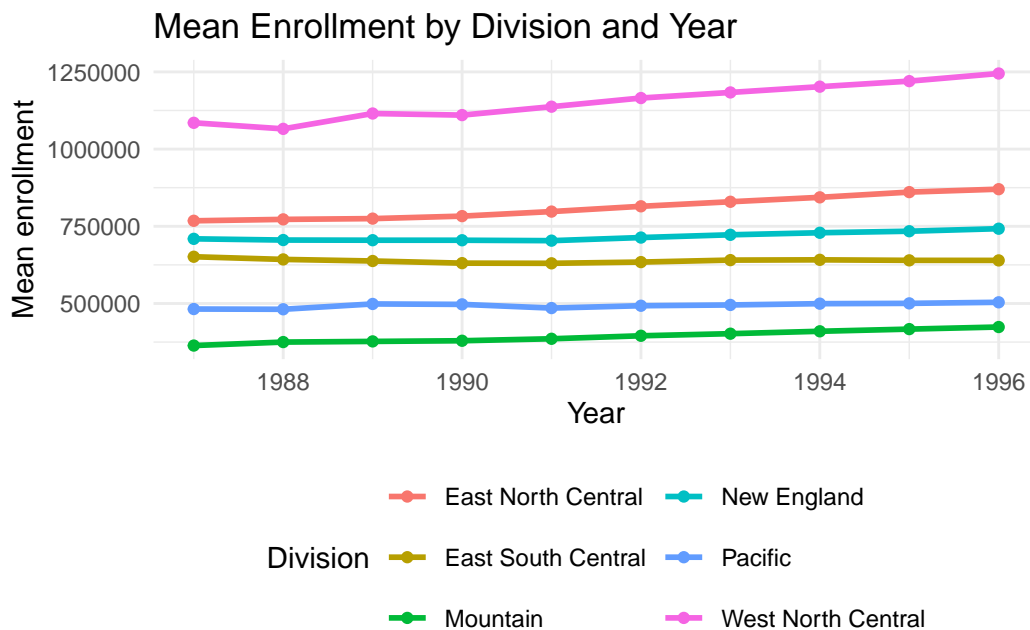
The 2 EDU01 Data Sets

```
#run wrapper function twice for edu files
edu01a <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01a.csv")
edu01b <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01b.csv")

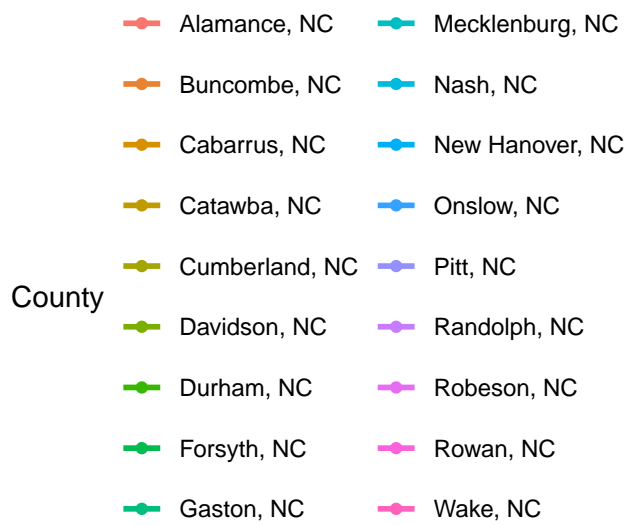
#combine the two data sets
edu_combined <- combine_data(edu01a, edu01b)

#plot for state data frame
plot(edu_combined$state)
```

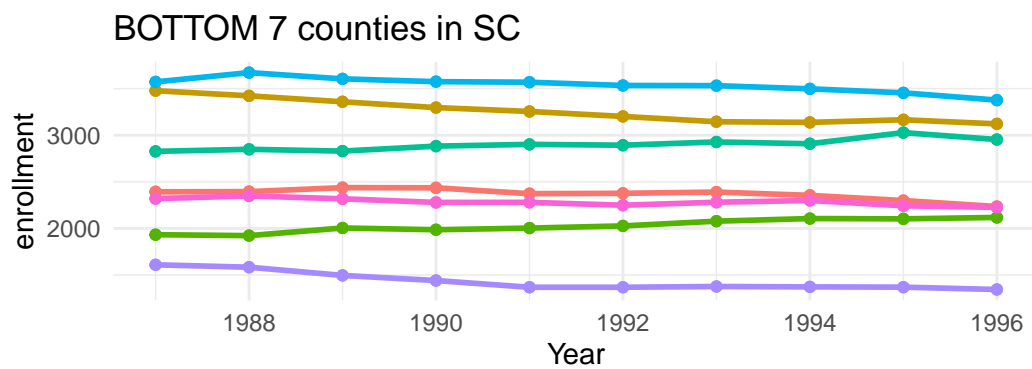
`summarise()` has grouped output by 'division'. You can override using the
`.groups` argument.



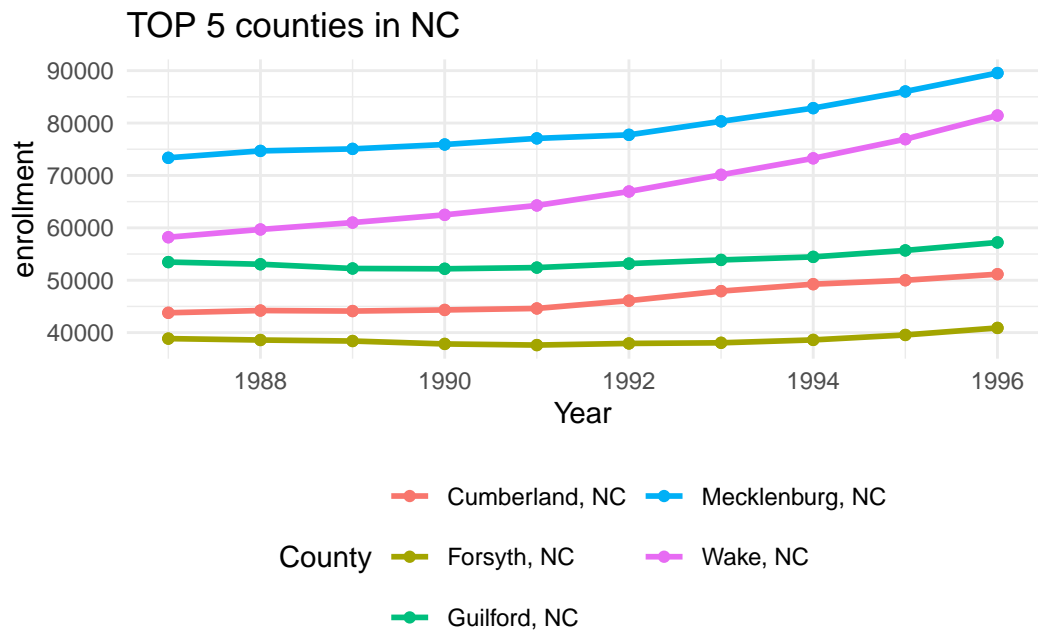
```
#plot for county data frame with the 4 different calls
plot(edu_combined$county,
     state = "NC", direction = "top", n = 20)
```



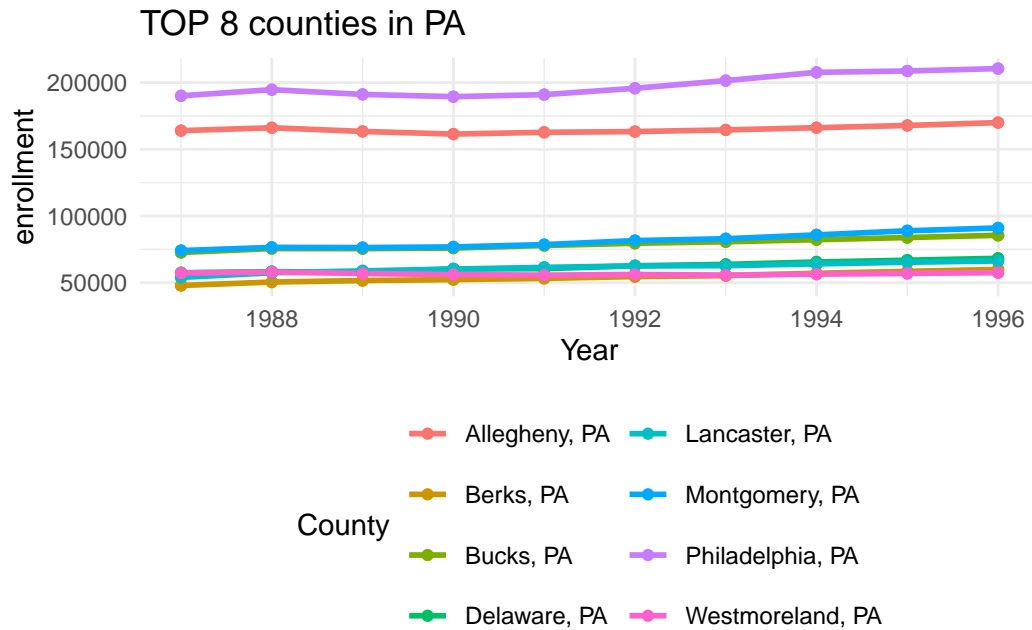
```
plot(edu_combined$county,
     state = "SC", direction = "bottom", n = 7)
```



```
plot(edu_combined$county) # uses defaults: NC, top, 5
```



```
plot(edu_combined$county,  
     state = "PA", direction = "top", n = 8)
```



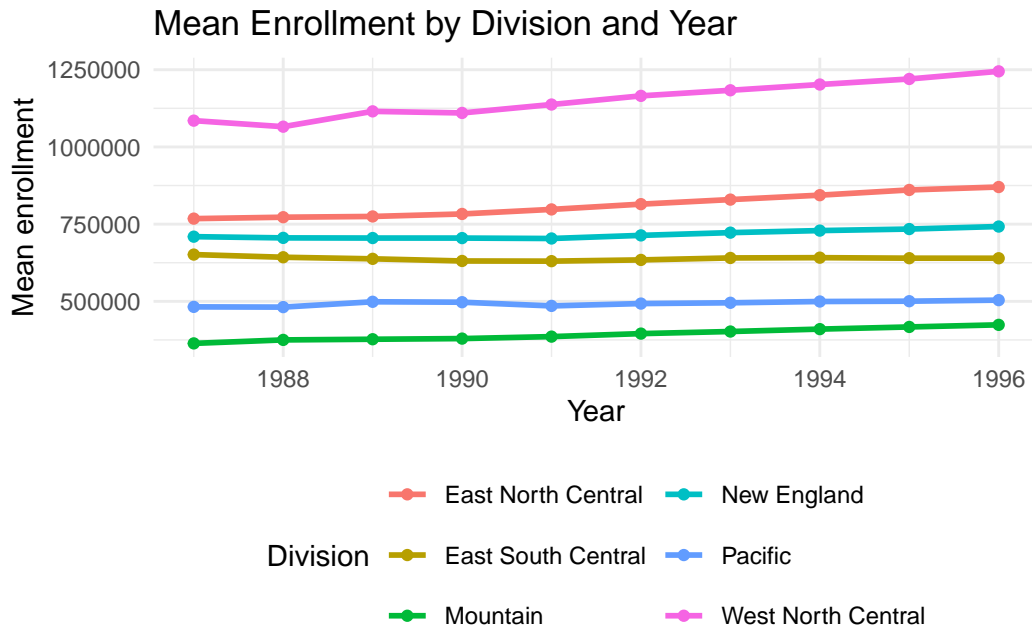
The 5 PST01 Data Sets

```
#run wrapper function for each data set
pst01a <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01a.csv")
pst01b <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01b.csv")
pst01c <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01c.csv")
pst01d <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01d.csv")

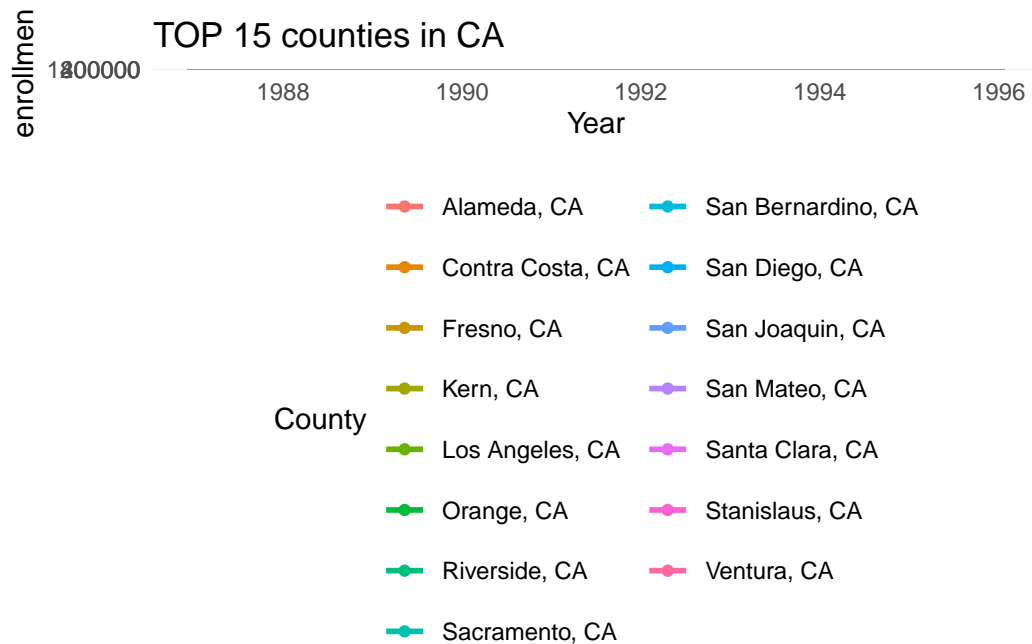
#combine the 4 data sets
#first, combine a and b
pst_combined_ab <- combine_data(pst01a, pst01b)
#second, combine c and d
pst_combined_cd <- combine_data(pst01c, pst01d)
#finally, combine ab with cd
pst_combined <- combine_data(pst_combined_ab, pst_combined_cd)

#plot for state data frame
plot(pst_combined$state, column = "enrollment")
```

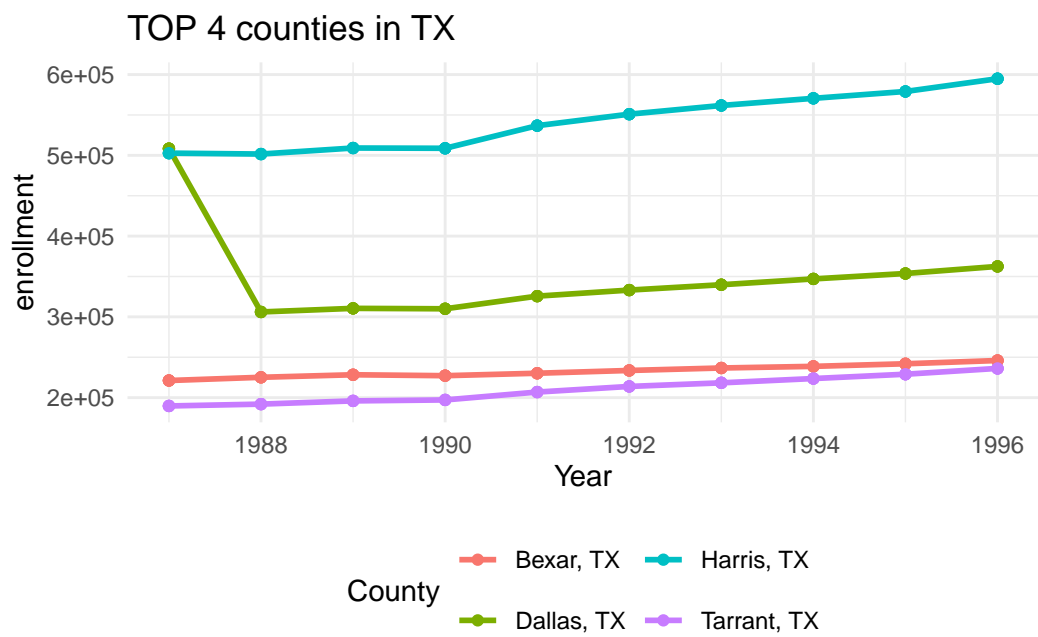
`summarise()` has grouped output by 'division'. You can override using the
 ` .groups ` argument.



```
#plot for county data frame with the 4 different calls
plot(pst_combined$county,
     state = "CA", direction = "top", n = 15)
```

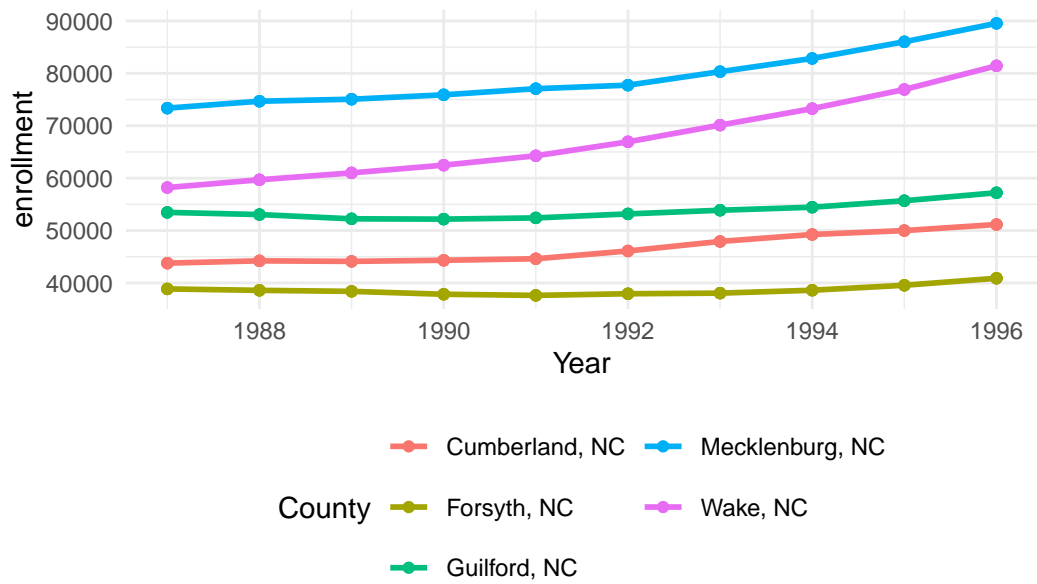


```
plot(pst_combined$county,  
     state = "TX", direction = "top", n = 4)
```



```
plot(pst_combined$county)
```

TOP 5 counties in NC



```
plot(pst_combined$county,
     state = "NY", direction = "top", n = 10)
```

TOP 10 counties in NY

