

# Project 1

Tamdan Le, Alise Miller

## Data Processing

### Question 1 Select columns

```
library(tidyverse)
```

Warning: package 'tidyverse' was built under R version 4.4.3

Warning: package 'ggplot2' was built under R version 4.4.3

Warning: package 'tibble' was built under R version 4.4.3

Warning: package 'tidyr' was built under R version 4.4.3

Warning: package 'readr' was built under R version 4.4.3

Warning: package 'purrr' was built under R version 4.4.3

Warning: package 'dplyr' was built under R version 4.4.3

Warning: package 'stringr' was built under R version 4.4.3

Warning: package 'forcats' was built under R version 4.4.3

Warning: package 'lubridate' was built under R version 4.4.3

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.4
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(readr)
```

```
EDU01A <-read_csv("data/EDU01a.csv")
```

```
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.
```

```
EDU01A |>
  select(Area_name, STCOU, ends_with("D")) |>
  rename(area_name = Area_name) |>
  head(EDU01A, 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>
```

## Question 2 Convert to long format

```
EDU01A_long<-  
  EDU01A |>  
  select(Area_name, STCOU, ends_with("D")) |>  
  rename(area_name = Area_name) |>  
  pivot_longer(cols= ends_with("D"),  
               names_to = "EDU_combined",  
               values_to = "enrollment_value"  
               )  
  head(EDU01A_long, n=5)
```

```
# A tibble: 5 x 4  
  area_name      STCOU EDU_combined enrollment_value  
  <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 Parsing

```
EDU01A_longer <-  
  EDU01A_long |>  
  mutate(two_digit_year= (substr(EDU_combined, start=8, stop = 9)) ,  
         year_dbl = as.double(two_digit_year),  
         year= if_else(year_dbl >25, 1900 + year_dbl, year_dbl <=25 + 2000),  
         survey_value = substr(EDU_combined, start=1, stop = 7)  
         )  
  
long_updated <- select(EDU01A_longer, area_name,STCOU, enrollment_value,year,  
                      survey_value)  
  
head(long_updated, n=5)
```

```
# A tibble: 5 x 5  
  area_name      STCOU enrollment_value year survey_value  
  <chr>         <chr>         <dbl> <dbl> <chr>
```

1	UNITED STATES	00000	40024299	1987	EDU0101
2	UNITED STATES	00000	39967624	1988	EDU0101
3	UNITED STATES	00000	40317775	1989	EDU0101
4	UNITED STATES	00000	40737600	1990	EDU0101
5	UNITED STATES	00000	41385442	1991	EDU0101

#### Question 4 Two Tibbles

```
County_indices <- grep(pattern = "[A-Z]{2}", long_updated$area_name)
noncounty_tibble <- long_updated[~County_indices, ]
county_tibble <- long_updated[County_indices, ]

class(county_tibble) <- c("county", class(county_tibble))
class(noncounty_tibble) <- c("state", class(noncounty_tibble))

head(county_tibble, n=10)
```

```
# A tibble: 10 x 5
  area_name STCOU enrollment_value year survey_value
  <chr>      <chr>          <dbl> <dbl> <chr>
1 Autauga, AL 01001      6829  1987 EDU0101
2 Autauga, AL 01001      6900  1988 EDU0101
3 Autauga, AL 01001      6920  1989 EDU0101
4 Autauga, AL 01001      6847  1990 EDU0101
5 Autauga, AL 01001      7008  1991 EDU0101
6 Autauga, AL 01001      7137  1992 EDU0101
7 Autauga, AL 01001      7152  1993 EDU0101
8 Autauga, AL 01001      7381  1994 EDU0101
9 Autauga, AL 01001      7568  1995 EDU0101
10 Autauga, AL 01001      7834  1996 EDU0101
```

```
head(noncounty_tibble, n=10)
```

```
# A tibble: 10 x 5
  area_name STCOU enrollment_value year survey_value
  <chr>      <chr>          <dbl> <dbl> <chr>
1 UNITED STATES 00000      40024299  1987 EDU0101
2 UNITED STATES 00000      39967624  1988 EDU0101
3 UNITED STATES 00000      40317775  1989 EDU0101
```

4	UNITED STATES	00000	40737600	1990	EDU0101
5	UNITED STATES	00000	41385442	1991	EDU0101
6	UNITED STATES	00000	42088151	1992	EDU0101
7	UNITED STATES	00000	42724710	1993	EDU0101
8	UNITED STATES	00000	43369917	1994	EDU0101
9	UNITED STATES	00000	43993459	1995	EDU0101
10	UNITED STATES	00000	44715737	1996	EDU0101

### Question 5 County level new variable

```
county_tibble |>
mutate(state = substr(area_name, nchar(area_name) - 1, nchar(area_name))
)
```

```
# A tibble: 31,450 x 6
  area_name STCOU enrollment_value year survey_value state
  <chr>      <chr>          <dbl> <dbl> <chr>      <chr>
1 Autauga, AL 01001          6829  1987 EDU0101    AL
2 Autauga, AL 01001          6900  1988 EDU0101    AL
3 Autauga, AL 01001          6920  1989 EDU0101    AL
4 Autauga, AL 01001          6847  1990 EDU0101    AL
5 Autauga, AL 01001          7008  1991 EDU0101    AL
6 Autauga, AL 01001          7137  1992 EDU0101    AL
7 Autauga, AL 01001          7152  1993 EDU0101    AL
8 Autauga, AL 01001          7381  1994 EDU0101    AL
9 Autauga, AL 01001          7568  1995 EDU0101    AL
10 Autauga, AL 01001          7834  1996 EDU0101    AL
# i 31,440 more rows
```

### Question 6 Non-county “division”

```
noncounty_tibble <- noncounty_tibble |>
mutate(
  state = sub(".*,\\s*", "", area_name),
division = case_when(
  state %in% c("CONNECTICUT", "MAINE", "MASSACHUSETTS", "NEW HAMPSHIRE",
    "RHODE ISLAND", "VERMONT")
    ~ "New England",
  state %in% c("NEW JERSEY", "NEW YORK", "PENNSYLVANIA")
```

```

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

```

## Function for Steps 1 and 2 - Reading in Data

```

readData <- function(filepath, columns= "!area_name & !STCOU") {
  data2 <- read.csv(filepath)
  filterdata <- select(data2, c(area_name = "Area_name", "STCOU"),
                        ends_with("D"))
  long_data <- pivot_longer(filterdata, cols = (!area_name & !STCOU),
                            names_to = "EDU_combined",
                            values_to = "enrollment_value")
}

```

## Function for Step 3 - Parsing

```

dataYear <- function(step2) {
  long_updated = mutate(step2, year_dbl= as.double(substr(EDU_combined,
                                                           start=8, stop = 9)),

```

```

year = if_else(year_dbl > 25, 1900 + year_dbl, 2000 + year_dbl),
survey_value = substr(EDU_combined, start=1, stop = 7)
)
long_updated <- subset(long_updated, select = -year_dbl)
}

```

### Function for Step 5 - County Level Tibble

```

state_function <- function(county_tibble){
  new_county_tibble <- mutate(county_tibble,
                              state = substr(area_name, nchar(area_name) - 1,
                                              nchar(area_name))
  )
  return(new_county_tibble)
}

```

### Function for Step 6 - Non-county Level tibble

```

division_function <- function(noncounty_tibble) {
  new_noncounty_tibble <- mutate(noncounty_tibble,
    state = sub(".*\\s*", "", area_name),
    division = case_when(state %in% c("CONNECTICUT", "MAINE", "MASSACHUSETTS",
      "NEW HAMPSHIRE", "RHODE ISLAND", "VERMONT")
      ~ "New England",
      state %in% c("NEW JERSEY", "NEW YORK", "PENNSYLVANIA")
      ~ "Mid-Atlantic",
      state %in% c("ILLINOIS", "INDIANA", "MICHIGAN", "OHIO",
        "WISCONSIN")
      ~ "East North Central",
      state %in% c("IOWA", "KANSAS", "MINNESOTA", "NEBRASKA",
        "NORTH DAKOTA", "SOUTH DAKOTA")
      ~ "West North Central",
      state %in% c("DELAWARE", "DISTRICT OF COLUMBIA",
        "District of Columbia", "FLORIDA",
        "GEORGIA", "MARYLAND", "NORTH CAROLINA",
        "SOUTH CAROLINA", "VIRGINIA",
        "WEST VIRGINIA") ~ "South Atlantic",
      state %in% c("ALABAMA", "KENTUCKY", "MISSISSIPPI",

```

```

        "TENNESSEE") ~ "East South Central",
state %in% c("ARKANSAS", "LOUISIANA", "OKLAHOMA",
            "TEXAS") ~ "West South Central",
state %in% c("ARIZONA", "COLORADO", "IDAHO", "MONTANA",
            "NEVADA", "NEW MEXICO", "UTAH", "WYOMING")
~ "Mountain",
state %in% c("ALASKA", "CALIFORNIA", "HAWAII", "OREGON",
            "WASHINGTON") ~ "Pacific",
TRUE ~ "ERROR" )
)
return(new_noncounty_tibble)
}

```

## Function for Step 4 - Create Two Tibbles

```

create_datasets <- function(long_data) {
  County_indices <- grep(pattern = "[A-Z]{2}", long_updated$area_name)
  noncounty_tibble <- long_updated[-County_indices, ]
  county_tibble <- long_updated[County_indices, ]
  class(county_tibble) <- c("county", class(county_tibble))
  class(noncounty_tibble) <- c("state", class(noncounty_tibble))
  final_county_tibble <- state_function(county_tibble)
  final_noncounty_tibble <- division_function(noncounty_tibble)
  return(list(final_county_tibble, final_noncounty_tibble))
}

```

## Wrapper Function

```

my_wrapper <- function(url, value = "Enrollment Value"){
  result <- read_csv(url) |>
  readData(value = value) |>
  dataYear() |>
  create_datasets()
  return(result)
}

```



## Call It and Combine Your Data

```
Data_1A <-my_wrapper("data/EDU01a.csv")
Data_1B <-my_wrapper("data/EDU01b.csv")

combine <-function (input1,input2) {
  all_county <-bind_rows(input1[[1]], input2[[1]])
  all_noncounty<-bind_rows(input1[[2]], input2[[2]])
  return(list(all_county, all_noncounty))
}

combined_data <-combine (Data_1A,Data_1B)
```

## Writing a Generic Function for Summarizing

### Plot Function for State

```
plot.state <- function(df, var_name = "enrollment_value") {
  ggplot(df[[2]] |>
    filter(division != "ERROR") |>
    group_by(division, year) |>
    mutate(mean = mean(get(var_name))),
    aes(x = year, y = mean, color = division)) +
  geom_line(aes(color = division)) +
  labs(x = "Year", y = "Mean of Enrollment Values",
    title = "State plot by Division") +
  scale_color_discrete(name = "Division")
}
```

### Plot Function for County

```
plot.county <- function(df, state_name = "NC", var_name = "enrollment_value",
  sortby = "top", sortvalue = 5) {
  newdf <- df[[1]] |>
    filter(state == state_name) |>
    group_by(area_name) |>
    mutate(mean = mean(get(var_name)))
  sortdf <- if (sortby == "top") {
```

```

    head(arrange(newdf, desc(mean)), n = sortvalue)
  } else if (sortby == "bottom") {
    head(arrange(newdf, mean), n = sortvalue)
  }
  ggplot(sortdf,
    aes(x = year, y = get(var_name), color = area_name)
  ) +
    geom_line(aes(color = area_name)) +
    labs(x = "Year", y = var_name, title = "County plot") +
    scale_color_discrete(name = "Area Name")
}

```

## Put It Together

### Data Processing Function

```

EDU01AWrapped<-my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01a.csv")
EDU01BWrapped<-my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/EDU01b.csv")

```

### Combine Function

```

combined_enrolled <- combine (EDU01AWrapped,EDU01BWrapped)

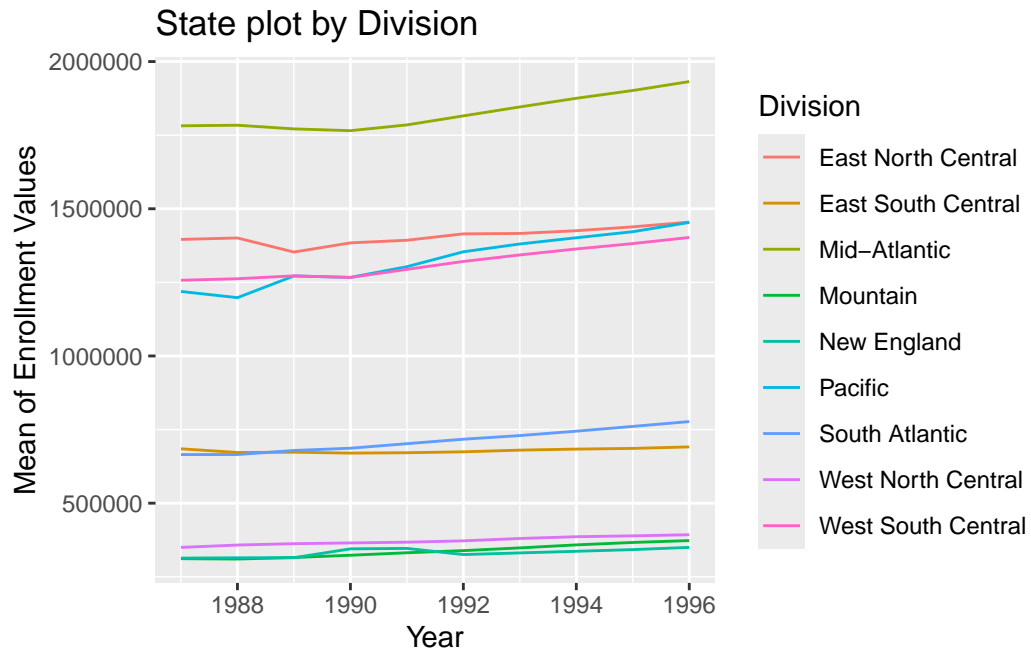
```

### Plot Function on State Data Frame

```

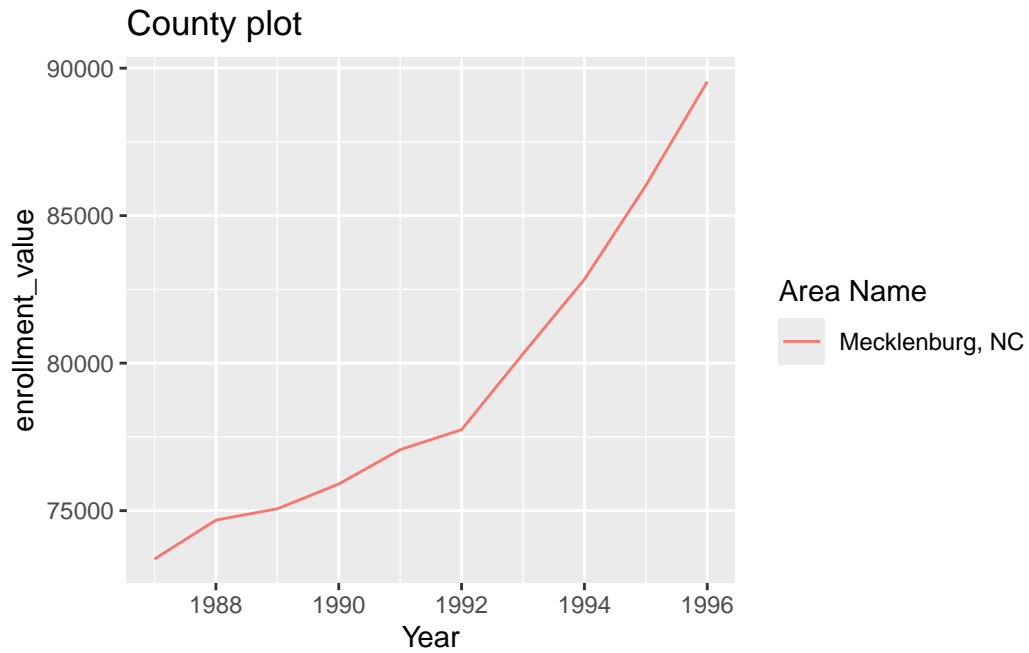
plot.state(combined_enrolled)

```

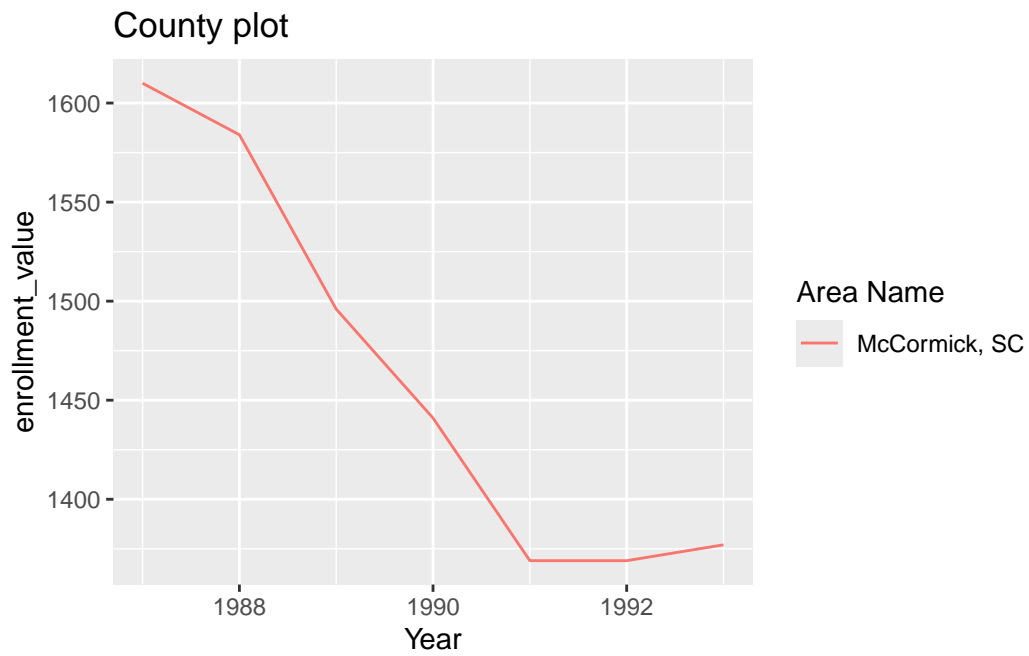


### Plot Function on County Data Frame

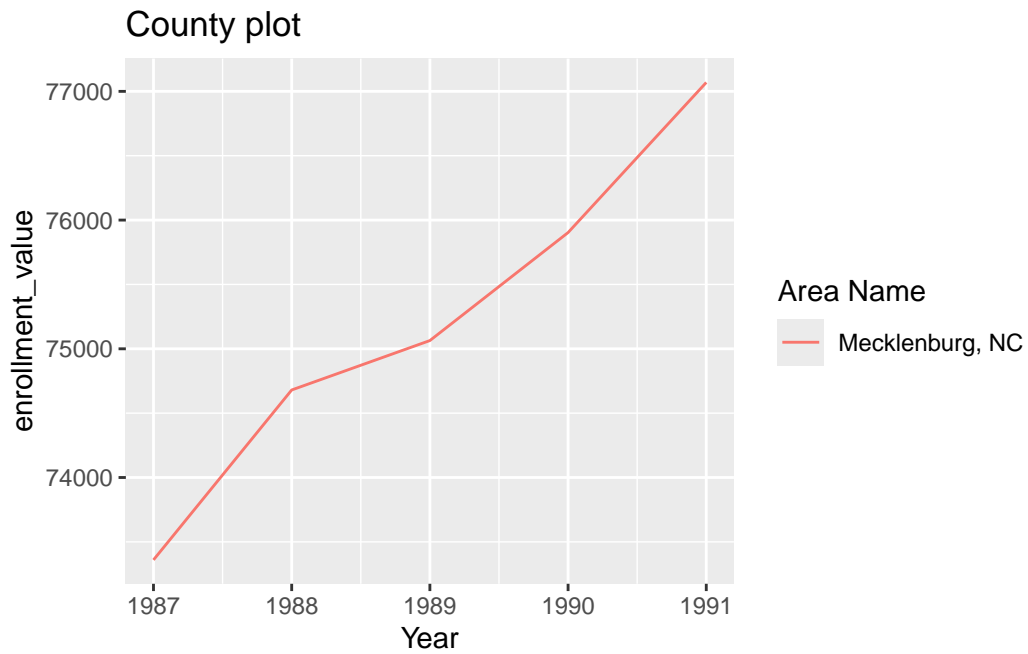
```
plot.county(combined_enrolled, sortvalue = 20)
```



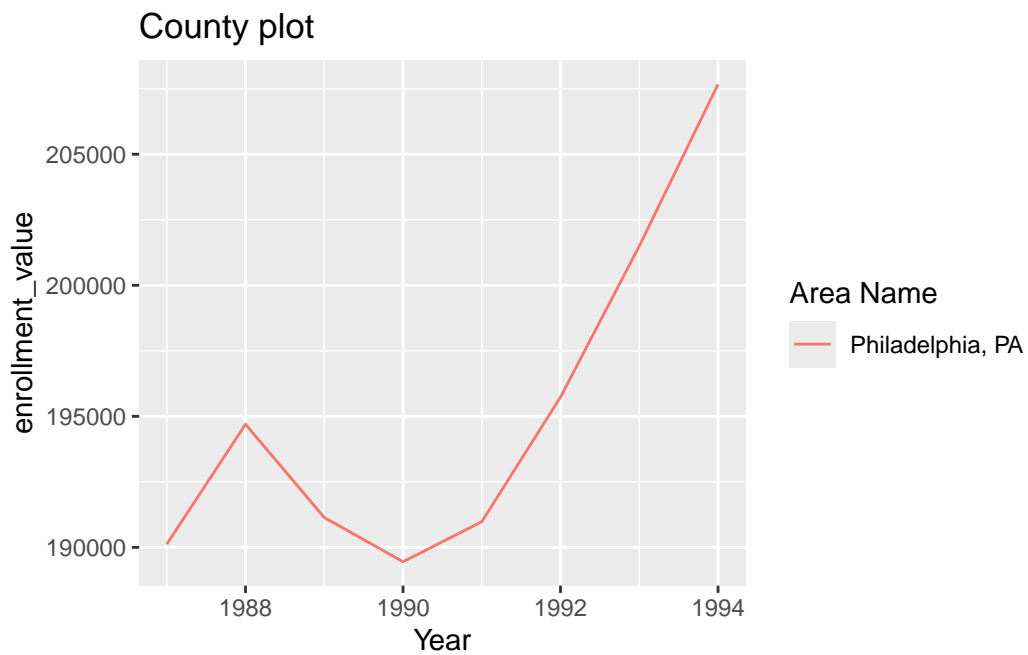
```
plot.county(combined_enrolled, state_name = "SC", var_name = "enrollment_value",  
            sortby = "bottom", sortvalue = 7 )
```



```
plot.county(combined_enrolled)
```



```
plot.county(combined_enrolled, state_name = "PA", sortby = "top", sortvalue = 8)
```



## Applying Functions to Other Data Sets

### Data Processing Function

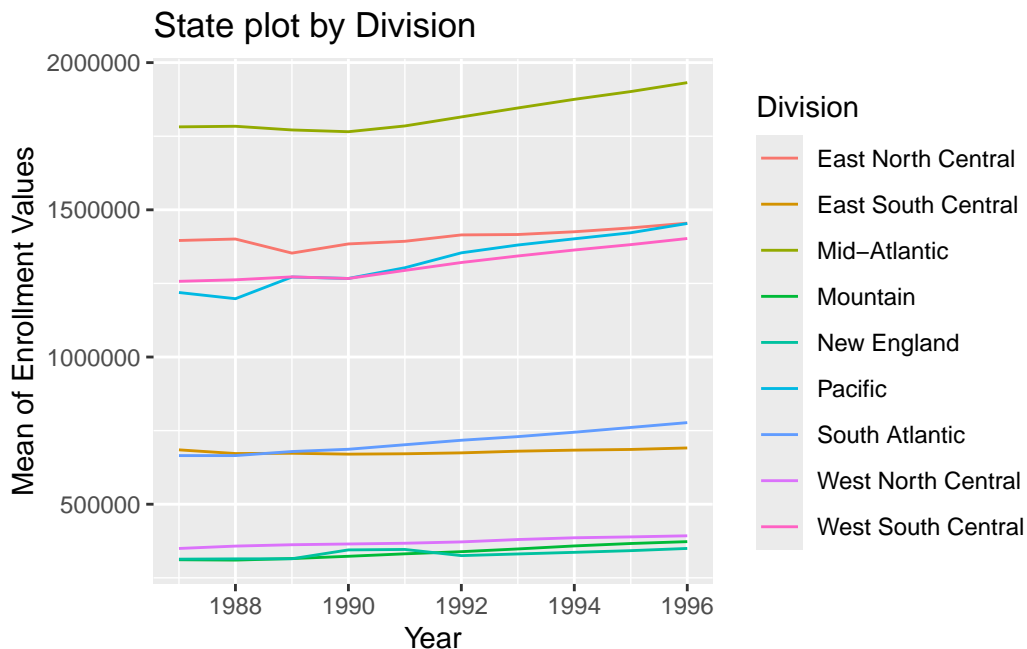
```
PST01a <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01a.csv")
PST01b <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01b.csv")
PST01c <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01c.csv")
PST01d <- my_wrapper("https://www4.stat.ncsu.edu/~online/datasets/PST01d.csv")
```

### Combine Function

```
Combined_PST01ab<-combine(PST01a,PST01b)
Combined_PST01cd <-combine(PST01c, PST01d)
Combined_PST01all <- combine(Combined_PST01ab, Combined_PST01cd )
```

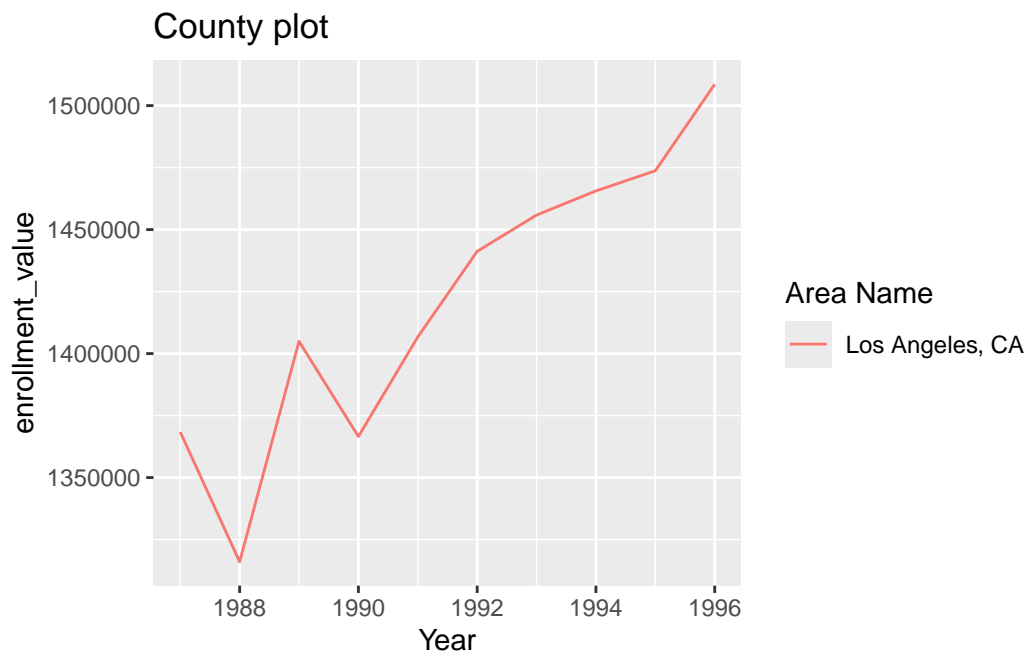
### Plot Function on State Data Frame

```
plot.state(Combined_PST01all)
```

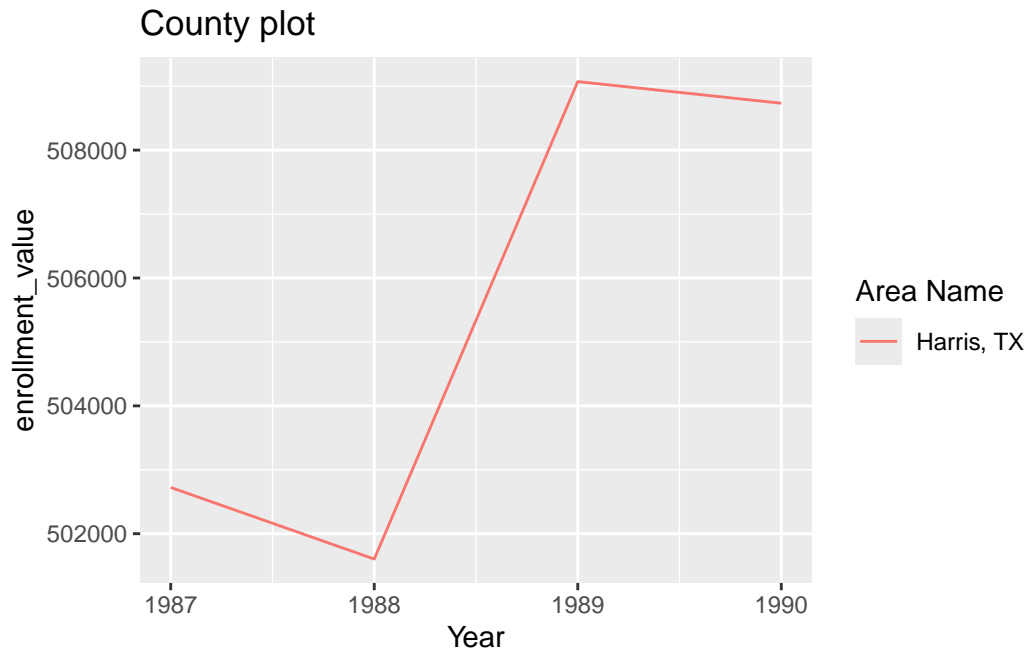


## Plot Function on County Data Frame

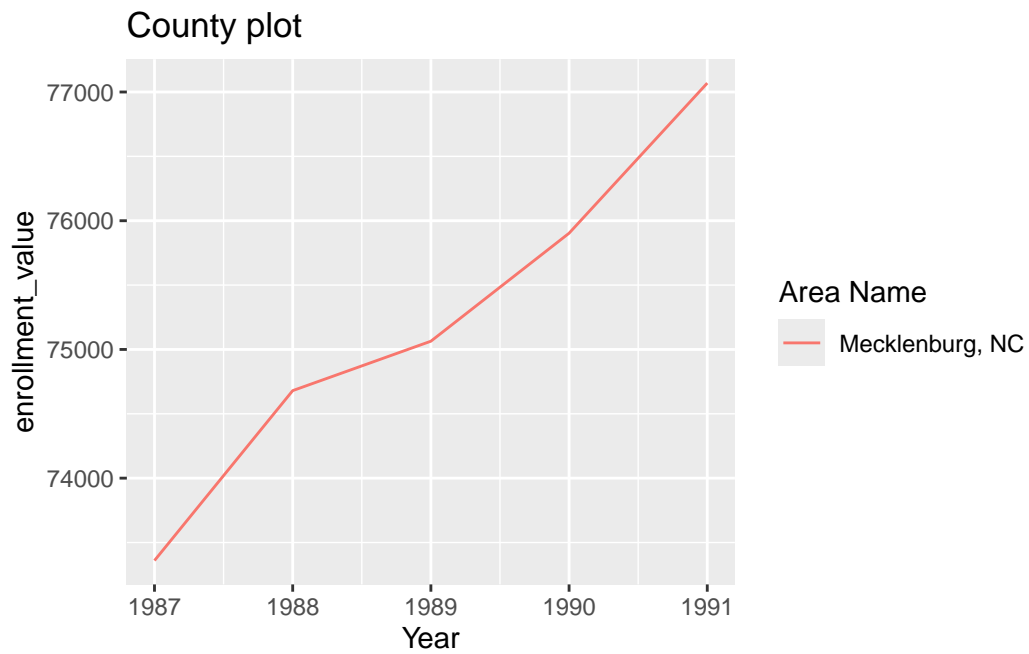
```
plot.county(Combined_PST01all, state_name = "CA", sortby = "top", sortvalue = 15)
```



```
plot.county(Combined_PST01all, state_name = "TX", sortby = "top", sortvalue = 4)
```



```
plot.county(Combined_PST01all)
```





```
plot.county(Combined_PST01all, state_name = "NY", sortby = "top", sortvalue = 10)
```

