

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)
  ) %>%
  mutate(
    #convert the 2-digit year into a 4-digit year (assuming all are 1900s)
    year = as.numeric(paste0("19", surveyID_year))
  ) %>%
  mutate(
    #extract the survey ID (first 7 characters of surveyID_full)
    surveyID = substr(surveyID_full, 1, 7)
  ) %>%
  #remove the temporary intermediate column
  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
    )
}

#test the function to see if it works correctly
edu_long <- select_pivot(edu01b)

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

```
# A tibble: 5 x 4
  area_name      STCOU surveyID_full enrollment
  <chr>          <chr> <chr>          <dbl>
1 UNITED STATES 00000 EDU010197D      44534459
2 UNITED STATES 00000 EDU010198D      46245814
3 UNITED STATES 00000 EDU010199D      46368903
4 UNITED STATES 00000 EDU010200D      46818690
5 UNITED STATES 00000 EDU010201D      47127066
```

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(paste0("19", surveyID_year)),  
      surveyID = substr(surveyID_full, 1, 7)  
    ) |>  
    select(-surveyID_year)  
}  
  
#test the function to see if it works correctly  
long_updated <- extract_year_id(edu_long)  
  
#display the first 5 rows  
long_updated |>  
  slice(1:5)
```

```
# A tibble: 5 x 6  
  area_name      STCOU surveyID_full enrollment   year surveyID  
  <chr>         <chr> <chr>          <dbl> <dbl> <chr>  
1 UNITED STATES 00000 EDU010197D      44534459 1997 EDU0101  
2 UNITED STATES 00000 EDU010198D      46245814 1998 EDU0101  
3 UNITED STATES 00000 EDU010199D      46368903 1999 EDU0101  
4 UNITED STATES 00000 EDU010200D      46818690 1900 EDU0102  
5 UNITED STATES 00000 EDU010201D      47127066 1901 EDU0102
```

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))  
    )  
}  
  
#test the function to see if it works correctly
```

```

#first repeat steps 4 to get county_tibble
county_indices <- grep(pattern = ", \\w\\w", long_updated$area_name)
county_tibble <- long_updated[county_indices, ]
class(county_tibble) <- c("county", class(county_tibble))

#use the function
county_tibble <- extract_state(county_tibble)

#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 EDU010197D      8099  1997 EDU0101  AL
2 Autauga, AL 01001 EDU010198D      8211  1998 EDU0101  AL
3 Autauga, AL 01001 EDU010199D      8489  1999 EDU0101  AL
4 Autauga, AL 01001 EDU010200D      8912  1900 EDU0102  AL
5 Autauga, AL 01001 EDU010201D      8626  1901 EDU0102  AL

```

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)
}

#test the function to see if it works correctly

#first repeat steps 4 to get state_tibble
county_indices <- grep(pattern = "\b\\w\\b", long_updated$area_name)
state_tibble <- long_updated[-county_indices, ]
class(state_tibble) <- c("state", class(state_tibble))

#use the function
state_tibble <- assign_division(state_tibble)

#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 EDU010197D      44534459 1997 EDU0101  ERROR
2 UNITED STATES 00000 EDU010198D      46245814 1998 EDU0101  ERROR
3 UNITED STATES 00000 EDU010199D      46368903 1999 EDU0101  ERROR
4 UNITED STATES 00000 EDU010200D      46818690 1900 EDU0102  ERROR
5 UNITED STATES 00000 EDU010201D      47127066 1901 EDU0102  ERROR

```

Create a Function Returning Two Final Tibbles

```

identify_locations <- function(data) {
  #step 4
  county_indices <- grep(pattern = "\b\\w\\b", 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))
}

#test the function to see if it works correctly

#applied the functions created above first
read_edu <- select_pivot(edu01b)
long_updated <- extract_year_id(read_edu)
edu01b_split <- identify_locations(long_updated)

#display the individual outputs
edu01b_split$county |>
  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 EDU010197D      8099  1997 EDU0101 AL
2 Autauga, AL 01001 EDU010198D      8211  1998 EDU0101 AL
3 Autauga, AL 01001 EDU010199D      8489  1999 EDU0101 AL
4 Autauga, AL 01001 EDU010200D      8912  1900 EDU0102 AL
5 Autauga, AL 01001 EDU010201D      8626  1901 EDU0102 AL

```

```

edu01b_split$state |>
  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 EDU010197D      44534459  1997 EDU0101 ERROR

```

2	UNITED STATES	00000	EDU010198D	46245814	1998	EDU0101	ERROR
3	UNITED STATES	00000	EDU010199D	46368903	1999	EDU0101	ERROR
4	UNITED STATES	00000	EDU010200D	46818690	1900	EDU0102	ERROR
5	UNITED STATES	00000	EDU010201D	47127066	1901	EDU0102	ERROR