Problem Set #5

Nicole Wood (in group NJ with Jade Levandofsky and Jessika Viveros)

October 28, 2023

Overview

Data

Colleges and universities purchase data on prospective students from vendors like the College Board in order to identify and recruit students to their institution. In this problem set, we will be working with the student list data that University of Illinois-Chicago purchased from College Board. Specifically, we will use the list from one specific order, where UI-Chicago filtered for all prospects who identified as American Indian or Alaska Native and scored within a specified test score/GPA range. Here is the order summary file containing the detailed search criteria.

To this student list data, we have also merged in Census data on zip-code characteristics and NCES data on high school characteristics for each prospect. Thus, some variables in the data are prospect-level variables, while others are measured at the zip-code level or school level. These include characteristics for the zip code the prospect lives in and characteristics for the high school which the prospect attends – those variables do not vary across prospects within the same zip-code or school.

Task

In this task, we are analyzing the characteristics of prospective students who identified as American Indian or Alaska Native when they took the SAT test. We analyze the ethnicity categories and race categories these students selected, where these students live, and their intended major. With respect to course learning goals, these analyses will help you practice processing across observations. From a substantive perspective, quantitative analyses seldom focus on students who identify as American Indian or Alaska Native, so the UI-Chicago student list purchase offers an opportunity to learn a little more about these students.

A note on terms for race and ethnicity categories: This problem set uses categories adopted by the U.S. Census. For example, the problem set uses "American Indian or Alaska Native" rather than the terms "Native American" or "Indigenous" and use the term "Hispanic" rather than "Latinx."

Part I: Loading library and data

1. Load the `tidyverse` library.

```
library(tidyverse)
#> -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
#> v dplyr     1.1.3     v readr     2.1.4
#> v forcats   1.0.0     v stringr   1.5.0
#> v ggplot2   3.4.4     v tibble    3.2.1
```

```
#> v lubridate 1.9.3     v tidyr     1.3.0
#> v purrr     1.0.2
#> -- 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 error
```

2. Use `load()` and `url()` to load the `list_native_df` dataframe from `https://github.com/anyone-can-cook/rclass1/`

```
load(url("https://github.com/anyone-can-cook/rclass1/raw/master/data/prospect_list/list_native_df.RData
```

3. Let's investigate the `list_native_df` dataframe. First, use `head()` and `glimpse()` to preview the data.

```
head(list_native_df)
#> # A tibble: 6 x 71
#>   univ_id ord_num univ_state univ_zip stu_state stu_city        stu_zip_code
#>   <chr>   <chr>   <hvn_lbll> <chr>    <chr>     <chr>           <chr>
#> 1 145600  487927  IL         60607    GA        Marietta        30062
#> 2 145600  487927  IL         60607    MD        Silver Spring   20904
#> 3 145600  487927  IL         60607    FL        Miramar         33029
#> 4 145600  487927  IL         60607    MD        Silver Spring   20904
#> 5 145600  487927  IL         60607    TX        College Station 77845
#> 6 145600  487927  IL         60607    TX        Houston         77079
#> # i 64 more variables: stu_geomarket <chr>, stu_country <chr>, stu_in_us <dbl>,
#> #   stu_hs_code <chr>, stu_county_code <chr>, stu_gender <chr>,
#> #   stu_cuban <chr>, stu_mexican <chr>, stu_puerto_rican <chr>,
#> #   stu_other_hispanic <chr>, stu_non_hispanic <chr>,
#> #   stu_ethnicity_no_response <chr>, stu_american_indian <chr>,
#> #   stu_asian <chr>, stu_black <chr>, stu_native_hawaiian <chr>,
#> #   stu_white <chr>, stu_race_no_response <chr>, stu_major_1 <chr>, ...
glimpse(list_native_df)
#> Rows: 14,681
#> Columns: 71
#> $ univ_id                  <chr> "145600", "145600", "145600", "145600", "1~
#> $ ord_num                  <chr> "487927", "487927", "487927", "487927", "4~
#> $ univ_state               <hvn_lbll> IL, IL, IL, IL, IL, IL, IL, IL, IL, I~
#> $ univ_zip                 <chr> "60607", "60607", "60607", "60607", "60607~
#> $ stu_state                <chr> "GA", "MD", "FL", "MD", "TX", "TX", "NH", ~
#> $ stu_city                 <chr> "Marietta", "Silver Spring", "Miramar", "S~
#> $ stu_zip_code             <chr> "30062", "20904", "33029", "20904", "77845~
#> $ stu_geomarket            <chr> "GA01", "MD02", "FL05", "MD02", "TX12", "T~
#> $ stu_country              <chr> "united states", "united states", "united ~
#> $ stu_in_us                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
#> $ stu_hs_code              <chr> "111986", "210959", "101807", "210959", "4~
#> $ stu_county_code          <chr> "13067", "24031", "12011", "24031", "48041~
#> $ stu_gender               <chr> "M", "F", "M", "F", "F", "M", "F", "M", "M~
#> $ stu_cuban                <chr> NA, NA, "Y", NA, NA, NA, NA, NA, NA, NA, N~
#> $ stu_mexican              <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
#> $ stu_puerto_rican         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
#> $ stu_other_hispanic       <chr> "Y", "Y", "Y", "Y", NA, NA, NA, NA, NA, "Y~
#> $ stu_non_hispanic         <chr> NA, NA, NA, NA, "Y", NA, "Y", "Y", "Y", NA~
```

```
#> $ stu_ethnicity_no_response   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
#> $ stu_american_indian         <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y~
#> $ stu_asian                   <chr> NA, NA, NA, NA, NA, "Y", NA, NA, NA, NA, "~
#> $ stu_black                   <chr> NA, NA, "Y", NA, NA, NA, NA, NA, NA, NA, N~
#> $ stu_native_hawaiian         <chr> NA, NA, NA, NA, NA, "Y", NA, NA, NA, NA, N~
#> $ stu_white                   <chr> NA, NA, NA, NA, NA, "Y", "Y", NA, "Y", "Y"~
#> $ stu_race_no_response        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
#> $ stu_major_1                 <chr> "26.0202", "51.10", "998", "51", "13.1311"~
#> $ stu_major_1_group           <chr> "26", "51", "998", "51", "13", "14", "998"~
#> $ stu_major_1_text            <chr> "Biochemistry", "Clinical/medical laborato~
#> $ stu_major_1_group_text      <chr> "Biological Science", "Health Professions"~
#> $ na_zip_acs                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ zip_cbsa                    <chr> "12060", "47900", "33100", "47900", "17780~
#> $ zip_cbsatitle               <chr> "Atlanta-Sandy Springs-Roswell, GA", "Wash~
#> $ zip_csacode                 <chr> "122", "548", "370", "548", NA, "288", "14~
#> $ zip_csatitle                <chr> "Atlanta--Athens-Clarke County--Sandy Spri~
#> $ zip_median_household_income <dbl> 102269, 85376, 113082, 85376, 82929, 91375~
#> $ zip_pop_total               <dbl> 65801, 55275, 48161, 55275, 66649, 34122, ~
#> $ zip_pop_white               <dbl> 45546, 11275, 14770, 11275, 48224, 19629, ~
#> $ zip_pop_black               <dbl> 6869, 25738, 6728, 25738, 3337, 4262, 0, 2~
#> $ zip_pop_amerindian          <dbl> 215, 70, 90, 70, 255, 56, 0, 152, 0, 14, 6~
#> $ zip_pop_asian               <dbl> 6071, 8130, 2746, 8130, 5078, 2720, 35, 43~
#> $ zip_pop_nativehawaii        <dbl> 23, 23, 30, 23, 29, 0, 0, 12, 0, 1, 8, 4, ~
#> $ zip_pop_otherrace           <dbl> 538, 121, 275, 121, 26, 98, 5, 0, 0, 15, 5~
#> $ zip_pop_tworaces            <dbl> 2373, 1749, 1361, 1749, 1029, 855, 142, 17~
#> $ zip_pop_hispanic            <dbl> 4166, 8169, 22161, 8169, 8671, 6502, 71, 4~
#> $ na_hs                       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ hs_private                  <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ hs_cbsa                     <chr> "12060", "47900", "33100", "47900", "17780~
#> $ hs_cbsatitle                <chr> "Atlanta-Sandy Springs-Roswell, GA", "Wash~
#> $ hs_csacode                  <chr> "122", "548", "370", "548", NA, "288", "14~
#> $ hs_csatitle                 <chr> "Atlanta--Athens-Clarke County--Sandy Spri~
#> $ hs_name                     <chr> "THE WALKER SCHOOL", "James Hubert Blake H~
#> $ hs_ncessch                  <chr> "00297383", "240048001044", "120018004052"~
#> $ hs_state_code               <chr> "GA", "MD", "FL", "MD", "TX", "TX", "NH", ~
#> $ hs_zip_code                 <chr> "30062", "20905", "33027", "20905", "77840~
#> $ hs_total_students           <int> 821, 1624, 2477, 1624, 1996, 2148, 790, 15~
#> $ hs_total_amerindian         <int> 1, 2, 5, 2, 5, 5, 2, 1, 1, 4, 1, 1, 2, 9, ~
#> $ hs_total_asian              <int> 84, 151, 213, 151, 157, 180, 6, 8, 48, 137~
#> $ hs_total_black              <int> 51, 668, 973, 668, 191, 209, 11, 8, 164, 1~
#> $ hs_total_hispanic           <int> 46, 424, 999, 424, 380, 587, 10, 1425, 92,~
#> $ hs_total_nativehawaii       <int> 0, 0, 0, 0, 2, 3, 1, 0, 2, 0, 29, 1, 0, 1,~
#> $ hs_total_tworaces           <int> 56, 84, 58, 84, 50, 51, 17, 6, 18, 72, 42,~
#> $ hs_total_unknown            <int> NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
#> $ hs_total_white              <int> 583, 295, 229, 295, 1211, 1113, 743, 101, ~
#> $ hs_pct_amerindian           <dbl> 0.12180268, 0.12315271, 0.20185709, 0.1231~
#> $ hs_pct_asian                <dbl> 10.2314251, 9.2980296, 8.5991118, 9.298029~
#> $ hs_pct_black                <dbl> 6.2119367, 41.1330049, 39.2813888, 41.1330~
#> $ hs_pct_hispanic             <dbl> 5.6029233, 26.1083744, 40.3310456, 26.1083~
#> $ hs_pct_nativehawaii         <dbl> 0.00000000, 0.00000000, 0.00000000, 0.0000~
#> $ hs_pct_tworaces             <dbl> 6.8209501, 5.1724138, 2.3415422, 5.1724138~
#> $ hs_pct_unknown              <dbl> NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
#> $ hs_pct_white                <dbl> 71.0109622, 18.1650246, 9.2450545, 18.1650~
```

4. For each of the following **ethnicity** variables, use the `count()` function to count its unique values:

    - stu_cuban
    - stu_mexican
    - stu_puerto_rican
    - stu_other_hispanic
    - stu_non_hispanic
    - stu_ethnicity_no_response

```
list_native_df %>% count(stu_cuban)
#> # A tibble: 2 x 2
#>   stu_cuban     n
#>   <chr>     <int>
#> 1 Y           182
#> 2 <NA>      14499
list_native_df %>% count(stu_mexican)
#> # A tibble: 2 x 2
#>   stu_mexican     n
#>   <chr>       <int>
#> 1 Y            4160
#> 2 <NA>        10521
list_native_df %>% count(stu_puerto_rican)
#> # A tibble: 2 x 2
#>   stu_puerto_rican     n
#>   <chr>            <int>
#> 1 Y                  593
#> 2 <NA>             14088
list_native_df %>% count(stu_other_hispanic)
#> # A tibble: 2 x 2
#>   stu_other_hispanic     n
#>   <chr>              <int>
#> 1 Y                   2912
#> 2 <NA>               11769
list_native_df %>% count(stu_non_hispanic)
#> # A tibble: 2 x 2
#>   stu_non_hispanic     n
#>   <chr>            <int>
#> 1 Y                 7248
#> 2 <NA>              7433
list_native_df %>% count(stu_ethnicity_no_response)
#> # A tibble: 2 x 2
#>   stu_ethnicity_no_response     n
#>   <chr>                     <int>
#> 1 Y                            76
#> 2 <NA>                     14605
```

5. For each of the following **race** variables, use the `count()` function to count its unique values:

    - stu_american_indian
    - stu_asian
    - stu_black
    - stu_native_hawaiian
    - stu_white
    - stu_race_no_response

```
list_native_df %>% count(stu_american_indian)
#> # A tibble: 2 x 2
#>   stu_american_indian     n
#>   <chr>               <int>
#> 1 Y                   14572
#> 2 <NA>                  109
list_native_df %>% count(stu_asian)
#> # A tibble: 2 x 2
#>   stu_asian     n
#>   <chr>     <int>
#> 1 Y          1120
#> 2 <NA>      13561
list_native_df %>% count(stu_black)
#> # A tibble: 2 x 2
#>   stu_black     n
#>   <chr>     <int>
#> 1 Y          2202
#> 2 <NA>      12479
list_native_df %>% count(stu_native_hawaiian)
#> # A tibble: 2 x 2
#>   stu_native_hawaiian     n
#>   <chr>               <int>
#> 1 Y                     414
#> 2 <NA>                14267
list_native_df %>% count(stu_white)
#> # A tibble: 2 x 2
#>   stu_white     n
#>   <chr>     <int>
#> 1 Y          7970
#> 2 <NA>       6711
list_native_df %>% count(stu_race_no_response)
#> # A tibble: 2 x 2
#>   stu_race_no_response     n
#>   <chr>                <int>
#> 1 Y                       23
#> 2 <NA>                 14658
```

Part II: Recreating College Board's aggregate race/ethnicity variable

In the questionnaire that students fill out during the College Board exams, they are allowed to select multiple ethnicity and race categories that they identify as. For example, a student who checks the box for "Cuban" could also check the box for "Non-hispanic." Similarly, a student who checks the box for "American Indian or Alaska Native" could also check the box for "Black." Here are more details on how College Board defines their race and ethnicity data.

These College Board variables are based off of the U.S. Census variables, as defined here. The specific Census variables we use in our dataset can be found here.

College Board also reports the student's aggregate race/ethnicity per U.S. Department of Education reporting guidelines, as defined here (see last page). This derived category allocates each student into 1 category. Below, we will recreate this College Board variable (`race_cb`).

To do that, we will first create 0/1 indicators for each disaggregated race and ethnicity variables. For example, we will create the 0/1 indicator variable `stu_hispanic_01`, whose value will be `1` if the student identifies as hispanic and `0` otherwise. Then, these 0/1 indicators, along with a couple other variables we create, will be used as input to recreate the `race_cb` variable.

Run the following code to create the new race/ethnicity categories. All code is provided for you, all you need to do is run the code chunk below. Make sure to remove the `eval = F` from the code chunk when you are ready to run this part.

```
list_native_df <- list_native_df %>% mutate(
  # create 0/1 variable for identifies as hispanic
  stu_hispanic_01 = case_when(
    (stu_cuban == 'Y' | stu_mexican == 'Y' | stu_puerto_rican == 'Y' | stu_other_hispanic == 'Y') ~ 1,
    (stu_non_hispanic == 'Y' & is.na(stu_cuban) & is.na(stu_mexican) & is.na(stu_puerto_rican) & is.na(s
  ),
  # create 0/1 variables for each ethnicity group
  stu_cuban_01 = case_when(stu_cuban == 'Y' ~ 1,is.na(stu_cuban) & is.na(stu_ethnicity_no_response) ~ 0)
  stu_mexican_01 = case_when(stu_mexican == 'Y' ~ 1,is.na(stu_mexican) & is.na(stu_ethnicity_no_response
  stu_puerto_rican_01 = case_when(stu_puerto_rican == 'Y' ~ 1,is.na(stu_puerto_rican) & is.na(stu_ethni
  stu_other_hispanic_01 = case_when(stu_other_hispanic == 'Y' ~ 1,is.na(stu_other_hispanic) & is.na(stu_
  # create 0/1 variables for each race group
  stu_american_indian_01 = case_when(stu_american_indian == 'Y' ~ 1,is.na(stu_american_indian) & is.na(s
  stu_asian_01 = case_when(stu_asian == 'Y' ~ 1,is.na(stu_asian) & is.na(stu_race_no_response) ~ 0),
  stu_black_01 = case_when(stu_black == 'Y' ~ 1,is.na(stu_black) & is.na(stu_race_no_response) ~ 0),
  stu_native_hawaiian_01 = case_when(stu_native_hawaiian == 'Y' ~ 1,is.na(stu_native_hawaiian) & is.na(s
  stu_white_01 = case_when(stu_white == 'Y' ~ 1,is.na(stu_white) & is.na(stu_race_no_response) ~ 0),
  # create count of number of race groups
  race_ct = rowSums(dplyr::across(c(stu_american_indian_01,stu_asian_01,stu_black_01,stu_native_hawaiian
  # create 0/1 measure of multi-race
  multi_race_01 = if_else(race_ct >=2,1,0, missing = NULL),
  # create college board categorical ethnicity race variable
  race_cb = case_when(
    # 0    No Response
    (is.na(stu_hispanic_01)==1 | (stu_hispanic_01==0 & stu_race_no_response=='Y')) ~ 'no_response',
    # 1    American Indian/Alaska Native
    (stu_american_indian_01==1 & multi_race_01 == 0 & stu_hispanic_01 == 0) ~ 'ai_an',
    # 2    Asian
    (stu_asian_01==1 & multi_race_01 == 0 & stu_hispanic_01 == 0) ~ 'asian',
    # 3    Black/African American
    (stu_black_01==1 & multi_race_01 == 0 & stu_hispanic_01 == 0) ~ 'black',
    # 4    Hispanic/Latino
    (stu_hispanic_01==1) ~ 'hispanic',
    # 8    Native Hawaiian or Other Pacific Islander
    (stu_native_hawaiian_01==1 & multi_race_01 == 0 & stu_hispanic_01 == 0) ~ 'nh_pi',
    # 9    White
    (stu_white_01==1 & multi_race_01 == 0 & stu_hispanic_01 == 0) ~ 'white',
    # 12   Two Or More Races, Non-Hispanic
    (multi_race_01 == 1 & stu_hispanic_01 == 0) ~ 'multi_race'
  )
) %>%
  # drop input ethnicity/race vars
  select(-stu_cuban,-stu_mexican,-stu_puerto_rican,-stu_other_hispanic,-stu_non_hispanic,-stu_american_i
```

1. After adding the new variables, let's investigate the `list_native_df` dataframe again. Use `head()` and `glimpse()` to preview the data.

```
head(list_native_df)
#> # A tibble: 6 x 72
#>   univ_id ord_num univ_state univ_zip stu_state stu_city        stu_zip_code
#>   <chr>   <chr>   <hvn_lbll> <chr>    <chr>     <chr>           <chr>
#> 1 145600  487927  IL         60607    GA        Marietta        30062
#> 2 145600  487927  IL         60607    MD        Silver Spring   20904
#> 3 145600  487927  IL         60607    FL        Miramar         33029
#> 4 145600  487927  IL         60607    MD        Silver Spring   20904
#> 5 145600  487927  IL         60607    TX        College Station 77845
#> 6 145600  487927  IL         60607    TX        Houston         77079
#> # i 65 more variables: stu_geomarket <chr>, stu_country <chr>, stu_in_us <dbl>,
#> #   stu_hs_code <chr>, stu_county_code <chr>, stu_gender <chr>,
#> #   stu_major_1 <chr>, stu_major_1_group <chr>, stu_major_1_text <chr>,
#> #   stu_major_1_group_text <chr>, na_zip_acs <dbl>, zip_cbsa <chr>,
#> #   zip_cbsatitle <chr>, zip_csacode <chr>, zip_csatitle <chr>,
#> #   zip_median_household_income <dbl>, zip_pop_total <dbl>,
#> #   zip_pop_white <dbl>, zip_pop_black <dbl>, zip_pop_amerindian <dbl>, ...
glimpse(list_native_df)
#> Rows: 14,681
#> Columns: 72
#> $ univ_id                    <chr> "145600", "145600", "145600", "145600", "1~
#> $ ord_num                    <chr> "487927", "487927", "487927", "487927", "4~
#> $ univ_state                 <hvn_lbll> IL, IL, IL, IL, IL, IL, IL, IL, IL, I~
#> $ univ_zip                   <chr> "60607", "60607", "60607", "60607", "60607~
#> $ stu_state                  <chr> "GA", "MD", "FL", "MD", "TX", "TX", "NH", ~
#> $ stu_city                   <chr> "Marietta", "Silver Spring", "Miramar", "S~
#> $ stu_zip_code               <chr> "30062", "20904", "33029", "20904", "77845~
#> $ stu_geomarket              <chr> "GA01", "MD02", "FL05", "MD02", "TX12", "T~
#> $ stu_country                <chr> "united states", "united states", "united ~
#> $ stu_in_us                  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
#> $ stu_hs_code                <chr> "111986", "210959", "101807", "210959", "4~
#> $ stu_county_code            <chr> "13067", "24031", "12011", "24031", "48041~
#> $ stu_gender                 <chr> "M", "F", "M", "F", "F", "M", "F", "M", "M~
#> $ stu_major_1                <chr> "26.0202", "51.10", "998", "51", "13.1311"~
#> $ stu_major_1_group          <chr> "26", "51", "998", "51", "13", "14", "998"~
#> $ stu_major_1_text           <chr> "Biochemistry", "Clinical/medical laborato~
#> $ stu_major_1_group_text     <chr> "Biological Science", "Health Professions"~
#> $ na_zip_acs                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ zip_cbsa                   <chr> "12060", "47900", "33100", "47900", "17780~
#> $ zip_cbsatitle              <chr> "Atlanta-Sandy Springs-Roswell, GA", "Wash~
#> $ zip_csacode                <chr> "122", "548", "370", "548", NA, "288", "14~
#> $ zip_csatitle               <chr> "Atlanta--Athens-Clarke County--Sandy Spri~
#> $ zip_median_household_income <dbl> 102269, 85376, 113082, 85376, 82929, 91375~
#> $ zip_pop_total              <dbl> 65801, 55275, 48161, 55275, 66649, 34122, ~
#> $ zip_pop_white              <dbl> 45546, 11275, 14770, 11275, 48224, 19629, ~
#> $ zip_pop_black              <dbl> 6869, 25738, 6728, 25738, 3337, 4262, 0, 2~
#> $ zip_pop_amerindian         <dbl> 215, 70, 90, 70, 255, 56, 0, 152, 0, 14, 6~
#> $ zip_pop_asian              <dbl> 6071, 8130, 2746, 8130, 5078, 2720, 35, 43~
#> $ zip_pop_nativehawaii       <dbl> 23, 23, 30, 23, 29, 0, 0, 12, 0, 1, 8, 4, ~
#> $ zip_pop_otherrace          <dbl> 538, 121, 275, 121, 26, 98, 5, 0, 0, 15, 5~
#> $ zip_pop_tworaces           <dbl> 2373, 1749, 1361, 1749, 1029, 855, 142, 17~
#> $ zip_pop_hispanic           <dbl> 4166, 8169, 22161, 8169, 8671, 6502, 71, 4~
#> $ na_hs                      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

```
#> $ hs_private                  <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ hs_cbsa                     <chr> "12060", "47900", "33100", "47900", "17780~
#> $ hs_cbsatitle                <chr> "Atlanta-Sandy Springs-Roswell, GA", "Wash~
#> $ hs_csacode                  <chr> "122", "548", "370", "548", NA, "288", "14~
#> $ hs_csatitle                 <chr> "Atlanta--Athens-Clarke County--Sandy Spri~
#> $ hs_name                     <chr> "THE WALKER SCHOOL", "James Hubert Blake H~
#> $ hs_ncessch                  <chr> "00297383", "240048001044", "120018004052"~
#> $ hs_state_code               <chr> "GA", "MD", "FL", "MD", "TX", "TX", "NH", ~
#> $ hs_zip_code                 <chr> "30062", "20905", "33027", "20905", "77840~
#> $ hs_total_students           <int> 821, 1624, 2477, 1624, 1996, 2148, 790, 15~
#> $ hs_total_amerindian         <int> 1, 2, 5, 2, 5, 5, 2, 1, 1, 4, 1, 1, 2, 9, ~
#> $ hs_total_asian              <int> 84, 151, 213, 151, 157, 180, 6, 8, 48, 137~
#> $ hs_total_black              <int> 51, 668, 973, 668, 191, 209, 11, 8, 164, 1~
#> $ hs_total_hispanic           <int> 46, 424, 999, 424, 380, 587, 10, 1425, 92,~
#> $ hs_total_nativehawaii       <int> 0, 0, 0, 0, 2, 3, 1, 0, 2, 0, 29, 1, 0, 1,~
#> $ hs_total_tworaces           <int> 56, 84, 58, 84, 50, 51, 17, 6, 18, 72, 42,~
#> $ hs_total_unknown            <int> NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
#> $ hs_total_white              <int> 583, 295, 229, 295, 1211, 1113, 743, 101, ~
#> $ hs_pct_amerindian           <dbl> 0.12180268, 0.12315271, 0.20185709, 0.1231~
#> $ hs_pct_asian                <dbl> 10.2314251, 9.2980296, 8.5991118, 9.298029~
#> $ hs_pct_black                <dbl> 6.2119367, 41.1330049, 39.2813888, 41.1330~
#> $ hs_pct_hispanic             <dbl> 5.6029233, 26.1083744, 40.3310456, 26.1083~
#> $ hs_pct_nativehawaii         <dbl> 0.00000000, 0.00000000, 0.00000000, 0.0000~
#> $ hs_pct_tworaces             <dbl> 6.8209501, 5.1724138, 2.3415422, 5.1724138~
#> $ hs_pct_unknown              <dbl> NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
#> $ hs_pct_white                <dbl> 71.0109622, 18.1650246, 9.2450545, 18.1650~
#> $ stu_hispanic_01             <dbl> 1, 1, 1, 1, 0, NA, 0, 0, 0, 1, 1, 0, 0, 0,~
#> $ stu_cuban_01                <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ stu_mexican_01              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ stu_puerto_rican_01         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ stu_other_hispanic_01       <dbl> 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, ~
#> $ stu_american_indian_01      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
#> $ stu_asian_01                <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, ~
#> $ stu_black_01                <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, ~
#> $ stu_native_hawaiian_01      <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ stu_white_01                <dbl> 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, ~
#> $ race_ct                     <dbl> 1, 1, 2, 1, 1, 4, 2, 1, 2, 2, 3, 4, 2, 2, ~
#> $ multi_race_01               <dbl> 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, ~
#> $ race_cb                     <chr> "hispanic", "hispanic", "hispanic", "hispa~
```

2. Now, let's take a look at the derived aggregate race/ethnicity variable `race_cb` we created. Create a new object `race_cb_freq` that stores the count for each race/ethnicity category as follows:

   - Use `count()` to get the count for each `race_cb` category
   - Use `arrange()` to sort by the count in descending order

```
race_cb_freq <- list_native_df %>% count(race_cb) %>% arrange(desc(n))
head(race_cb_freq)
#> # A tibble: 6 x 2
#>   race_cb         n
#>   <chr>       <int>
#> 1 hispanic     6987
#> 2 multi_race   5642
```

```
#> 3 ai_an        1529
#> 4 no_response   483
#> 5 white          33
#> 6 asian           3
```

3. Investigate the `race_cb_freq` object you created in the previous question by using the `typeof()` and `str()` functions. Run your code in the code chunk below and answer the following questions:

   - What type of object is this, and how many elements does it have?
     - **ANSWER**: Race_cb_freq is a list. There are 2 elements: race_cb and n.
   - Is this object a dataframe? If so, how many observations does it have, and what are the names of the variables?
     - **ANSWER**: Race_cb_freq is a dataframe, as it is a list in which each element is named. It has 9 observations; the variables are named race_cb and n.

```
typeof(race_cb_freq)
#> [1] "list"
str(race_cb_freq)
#> tibble [9 x 2] (S3: tbl_df/tbl/data.frame)
#>  $ race_cb: chr [1:9] "hispanic" "multi_race" "ai_an" "no_response" ...
#>  $ n      : int [1:9] 6987 5642 1529 483 33 3 2 1 1
```

4. Now, using `race_cb_freq`, add a column for the percentage of students in each `race_cb` category. Use `mutate()` to create a new variable that is the percent of students in each category. (Hint: Calculate the percent by dividing the count by the sum of all counts, then multiplying by 100)

```
race_cb_freq <- mutate(race_cb_freq, percent_stud = (n/sum(n)) * 100)
head(race_cb_freq)
#> # A tibble: 6 x 3
#>   race_cb         n percent_stud
#>   <chr>       <int>        <dbl>
#> 1 hispanic     6987      47.6
#> 2 multi_race   5642      38.4
#> 3 ai_an        1529      10.4
#> 4 no_response   483       3.29
#> 5 white          33       0.225
#> 6 asian           3       0.0204
```

Part III: Summarizing across rows

1. Now, let's investigate the 0/1 indicator variables we created earlier for each race/ethnicity variable. First, we'll take a look at `stu_hispanic_01`. Use `summarise()` to create the following variables (Hint: Refer to the lecture to figure out which helper functions to use):

   - The total number of students
   - The total number of students where `stu_hispanic_01` is missing
   - The percentage of students who identify as hispanic

```
list_native_df %>% summarise(total_students = n(),
                             total_missing_hisp_stu = sum(is.na(stu_hispanic_01)), perc_hisp_stu = mean
#> # A tibble: 1 x 3
#>   total_students total_missing_hisp_stu perc_hisp_stu
#>            <int>                  <int>         <dbl>
#> 1          14681                    482          49.2
```

2. Next, use `summarise()` to calculate the percentage of students who identify as each of the following category of race and ethnicity, and assign the result to an object named `race_ethnicity_pct`:

   - `stu_cuban_01`
   - `stu_mexican_01`
   - `stu_puerto_rican_01`
   - `stu_other_hispanic_01`
   - `stu_hispanic_01`
   - `stu_american_indian_01`
   - `stu_black_01`
   - `stu_native_hawaiian_01`
   - `stu_white_01`

   How do these percentages differ from the aggregated `race_cb` variable in which each student can only be in one group?

   - **ANSWER**: It is difficult to compare the percentages between the race_ethnicity_pct and race_cb dataframes, as they are measuring different variables. However, the race_ethnicity_pct allows for a more nuanced understanding of the students under study. The primary difference in the dataframes in the more detailed categories for students with South and Central American heritage and the ability to assign a student to more than one category. In the race_cb dataframe, we learn that 38% of students identify as "multi_race," and in the race_ethnicity_pct dataframe, we can see evidence of what that "multi_race" composition might be. For example, in race_cb, only 0.22 % of students were classified as white. In race_ethnicity_pct, 54.37% of students were classified as white, meaning that a large number of students likely have some combination of Native American/Native Alaskan and European heritage.

```
race_ethnicity_pct <- list_native_df %>% summarize(perc_cuban_stu = mean(stu_cuban_01*100, na.rm = TRUE)
   perc_mexican_stu = mean(stu_mexican_01*100, na.rm = TRUE),
   perc_puerto_rican_stu = mean(stu_puerto_rican_01*100, na.rm = TRUE),
   perc_other_hispanic_stu = mean(stu_other_hispanic_01*100, na.rm = TRUE),
   perc_hispanic_stu = mean(stu_hispanic_01*100, na.rm = TRUE),
   perc_american_indian_stu = mean(stu_american_indian_01*100, na.rm = TRUE),
   perc_black_stu = mean(stu_black_01*100, na.rm = TRUE),
   perc_native_hawaiian_stu = mean(stu_native_hawaiian_01*100, na.rm = TRUE),
   perc_white_stu = mean(stu_white_01*100, na.rm = TRUE))

print(race_ethnicity_pct)
#> # A tibble: 1 x 9
#>   perc_cuban_stu perc_mexican_stu perc_puerto_rican_stu perc_other_hispanic_stu
#>            <dbl>            <dbl>                 <dbl>                   <dbl>
#> 1           1.25             28.5                  4.06                    19.9
#> # i 5 more variables: perc_hispanic_stu <dbl>, perc_american_indian_stu <dbl>,
#> #   perc_black_stu <dbl>, perc_native_hawaiian_stu <dbl>, perc_white_stu <dbl>
```

3. Investigate the `race_ethnicity_pct` object you created in the previous question by using the `typeof()` and `str()` functions. Run your code in the code chunk below and answer the following questions:

- What type of object is this, and how many elements does it have?
  - **ANSWER**: The object type of race_ethnicity_pct is list, and it has 9 elements.
- Is this object a dataframe? If so, how many observations does it have, and what are the names of the variables?
  - **ANSWER**: Race_ethnicity_pct is a dataframe, as it is a list in which each element is named. It has 1 observation. The variable names are perc_cuban_stu, perc_mexican_stu, perc_puerto_rican_stu, perc_other_hispanic_stu, perc_hispanic_stu, perc_american_indian_stu, per_black_stu, per_native_hawaiian_stu, and perc_white_stu.

```
typeof(race_ethnicity_pct)
#> [1] "list"
str(race_ethnicity_pct)
#> tibble [1 x 9] (S3: tbl_df/tbl/data.frame)
#>  $ perc_cuban_stu          : num 1.25
#>  $ perc_mexican_stu        : num 28.5
#>  $ perc_puerto_rican_stu   : num 4.06
#>  $ perc_other_hispanic_stu : num 19.9
#>  $ perc_hispanic_stu       : num 49.2
#>  $ perc_american_indian_stu: num 99.4
#>  $ perc_black_stu          : num 15
#>  $ perc_native_hawaiian_stu: num 2.82
#>  $ perc_white_stu          : num 54.4
```

Part IV: Grouping and summarizing

1. Now, we will use `group_by()` in conjunction with `summarise()` to calculate summary results for each group. First, group by core-based statistical area (`zip_cbsatitle`) and calculate the following statistics for each CBSA:

   - The total number of students
   - The percentage of students who identify as each of the following race/ethnicity category:
     - `stu_cuban_01`
     - `stu_mexican_01`
     - `stu_puerto_rican_01`
     - `stu_other_hispanic_01`
     - `stu_hispanic_01`
     - `stu_american_indian_01`
     - `stu_black_01`
     - `stu_native_hawaiian_01`
     - `stu_white_01`

   Lastly, sort by the number of students per CBSA in descending order, and answer the following question. Note that a core-based statistical area by definition only includes urban areas. Observations where `zip_cbsatitle` is `NA` indicates that the student does not live in a CBSA (i.e., rural location).

   - In one or two sentences, what is something you find interesting about these results?
     - **ANSWER**: On the whole, the results tend to match larger trends of the US population. For example, larger percentages of Hispanic students are found in Texas and Southern California, while larger percentages of Black students are found in New York, Washington DC, and Detroit. While it's interesting that the largest percentage of Puerto Rican students live in New York, as opposed to the southwestern US, that is also in keeping with general population distributions, as Puerto Ricans represent nearly 10% of the population of New York City.

```r
CBSA_desc <- list_native_df %>%
  group_by(zip_cbsatitle) %>%
  summarise(total_stu = n(),
   perc_cuban = mean(stu_cuban_01*100, na.rm = TRUE),
   perc_mexican = mean(stu_mexican_01*100, na.rm = TRUE),
   perc_puerto_rican = mean(stu_puerto_rican_01*100, na.rm = TRUE),
   perc_other_hispanic = mean(stu_other_hispanic_01*100, na.rm = TRUE),
   perc_hispanic = mean(stu_hispanic_01*100, na.rm = TRUE),
   perc_american_indian = mean(stu_american_indian_01*100, na.rm = TRUE),
   perc_black = mean(stu_black_01*100, na.rm = TRUE),
   perc_native_hawaiian = mean(stu_native_hawaiian_01*100, na.rm = TRUE),
   perc_white = mean(stu_white_01*100, na.rm = TRUE)) %>%
  arrange(desc(total_stu))

head(CBSA_desc, n = 10)
#> # A tibble: 10 x 11
#>    zip_cbsatitle             total_stu perc_cuban perc_mexican perc_puerto_rican
#>    <chr>                         <int>      <dbl>        <dbl>             <dbl>
#>  1 Houston-The Woodlands-Su~       947      1.06          45.2              1.16
#>  2 New York-Newark-Jersey C~       936      1.30          13.8             11.6
#>  3 Chicago-Naperville-Elgin~       894      0.561         57.4              5.94
#>  4 Los Angeles-Long Beach-A~       855      0.589         67.7              0.824
#>  5 Dallas-Fort Worth-Arling~       770      0.781         41.5              1.43
#>  6 <NA>                            561      0.182         12.2              1.64
#>  7 Riverside-San Bernardino~       406      0.494         58.0              0.988
#>  8 Washington-Arlington-Ale~       341      1.18          11.8              5.29
#>  9 San Antonio-New Braunfel~       333      0.601         47.4              3.30
#> 10 Detroit-Warren-Dearborn,~       291      1.03          13.7              2.75
#> # i 6 more variables: perc_other_hispanic <dbl>, perc_hispanic <dbl>,
#> #   perc_american_indian <dbl>, perc_black <dbl>, perc_native_hawaiian <dbl>,
#> #   perc_white <dbl>
```

2. Next, we will look at the students' zip-code level median household income (`zip_median_household_income`) by state. Group by state (`stu_state`) and calculate the following statistics for each state:

   - The total number of students
   - The total number of students where `zip_median_household_income` is missing
   - The average median household income of students
   - The maximum median household income of students
   - The minimum median household income of students

   Lastly, sort by the number of students per state in descending order.

```r
stu_state_desc <- list_native_df %>%
  group_by(stu_state) %>%
  summarise(total_stu = n(),
            total_missing_stu = sum(is.na(zip_median_household_income)),
            ave_median_hh_inc = mean(zip_median_household_income),
            max_median_hh_inc = max(zip_median_household_income),
            min_median_hh_inc = min(zip_median_household_income)) %>%
  arrange(desc(total_stu))

head(stu_state_desc)
```

```
#> # A tibble: 6 x 6
#>   stu_state total_stu total_missing_stu ave_median_hh_inc max_median_hh_inc
#>   <chr>         <int>             <int>              <dbl>              <dbl>
#> 1 TX             2866                27                 NA                 NA
#> 2 CA             2541                 9                 NA                 NA
#> 3 IL             1127                 0             71387.             196964
#> 4 FL              928                 6                 NA                 NA
#> 5 NY              841                 1                 NA                 NA
#> 6 MI              774                 4                 NA                 NA
#> # i 1 more variable: min_median_hh_inc <dbl>
```

3. In the next few questions, we'll take a look at the students' intended major choice. First, group by major choice (`stu_major_1_group_text`) and summarize the number of students per major. Sort by the number of students in descending order and assign the result to an object named `major_group_freq`.

```
major_group_freq <- list_native_df %>%
  group_by(stu_major_1_group_text) %>%
  summarise(student_major = n()) %>%
  arrange(desc(student_major))

head(major_group_freq)
#> # A tibble: 6 x 2
#>   stu_major_1_group_text student_major
#>   <chr>                         <int>
#> 1 Health Professions             2223
#> 2 Engineering                    1823
#> 3 Biological Science             1537
#> 4 Business/Mgmt                  1427
#> 5 Visual/perform Art              972
#> 6 Undecided                       843
```

4. Using `major_group_freq`, add a column for the percentage of students in each `stu_major_1_group_text` category.

```
major_group_freq <- major_group_freq %>%
  mutate(percent_stud = student_major/sum(student_major)*100)

head(major_group_freq)
#> # A tibble: 6 x 3
#>   stu_major_1_group_text student_major percent_stud
#>   <chr>                         <int>        <dbl>
#> 1 Health Professions             2223        15.1
#> 2 Engineering                    1823        12.4
#> 3 Biological Science             1537        10.5
#> 4 Business/Mgmt                  1427         9.72
#> 5 Visual/perform Art              972         6.62
#> 6 Undecided                       843         5.74
```

5. Now, create the same table as the previous question that shows the count and percentage of students for each major choice, but instead of using `group_by()` and `summarise()`, use `count()` to get the

counts from the original `list_native_df` dataframe. Make sure to sort by descending student count. (Hint: Use a similar approach you used to create the frequency count of `race_cb` in Part II)

```
major_group_freq_count <- list_native_df %>%
  count(stu_major_1_group_text, name = "student_major") %>%
  mutate(percent_stud = (student_major/sum(student_major)*100)) %>%
  arrange(desc(student_major))
head(major_group_freq_count)
#> # A tibble: 6 x 3
#>   stu_major_1_group_text student_major percent_stud
#>   <chr>                         <int>        <dbl>
#> 1 Health Professions             2223        15.1
#> 2 Engineering                    1823        12.4
#> 3 Biological Science             1537        10.5
#> 4 Business/Mgmt                  1427         9.72
#> 5 Visual/perform Art              972         6.62
#> 6 Undecided                       843         5.74
```

6. We can also group by multiple variables. In this question, group by both state (`stu_state`) and the student's intended major (`stu_major_1_group_text`), then summarize the number of students per state and major. Sort by state, then the number of students in descending order. Assign the result to an object named `major_by_state_freq`.

```
major_by_state_freq <- list_native_df %>%
  group_by(stu_state, stu_major_1_group_text) %>%
  summarize(n_obs = n()) %>%
  arrange(stu_state, desc(n_obs))
#> `summarise()` has grouped output by 'stu_state'. You can override using the
#> `.groups` argument.

head(major_by_state_freq)
#> # A tibble: 6 x 3
#> # Groups:   stu_state [1]
#>   stu_state stu_major_1_group_text n_obs
#>   <chr>     <chr>                  <int>
#> 1 AK        Health Professions        13
#> 2 AK        Engineering               11
#> 3 AK        Biological Science         8
#> 4 AK        Visual/perform Art         7
#> 5 AK        Psychology                 5
#> 6 AK        Computer/Info Sys          4
glimpse(major_by_state_freq)
#> Rows: 1,006
#> Columns: 3
#> Groups: stu_state [53]
#> $ stu_state              <chr> "AK", "AK", "AK", "AK", "AK", "AK", "AK", "AK",~
#> $ stu_major_1_group_text <chr> "Health Professions", "Engineering", "Biologica~
#> $ n_obs                  <int> 13, 11, 8, 7, 5, 4, 4, 4, 3, 2, 2, 2, 2, 2, 1, ~
```

7. Looking at the `major_by_state_freq` dataframe from the previous question, answer the following questions:

   - How many observations are there?

  - **ANSWER**:There are 1,006 observations (rows).
- What does each observation represent?
  - **ANSWER**:Each observation represents a group of students sharing the same major in the same state.
- If we were to group/summarize by state, how many observations would the resulting object have? You will do this in the next question.
  - **ANSWER**: Grouping/summarizing by state returns 53 observations: a grouped observation for each state, Guam, Puerto Rico, and NA values.

8. Finally, we will look at the top 3 intended major choices by students from each state. Using `major_by_state_freq`, group by state and create the following variables:

   - The top choice major by students per state
   - The second choice major by students per state
     - Hint: Use `nth()` to get the nth value per group
   - The third choice major by students per state

```
#dataframe already arranged by n_obs in question 4.7.

major_by_state_freq_t3 <- major_by_state_freq %>%
  group_by(stu_state) %>%
  summarise(top_major = first(stu_major_1_group_text),
            second_major = nth(stu_major_1_group_text, 2),
            third_major = nth(stu_major_1_group_text, 3))
head(major_by_state_freq_t3)
#> # A tibble: 6 x 4
#>   stu_state top_major          second_major       third_major
#>   <chr>     <chr>              <chr>              <chr>
#> 1 AK        Health Professions Engineering        Biological Science
#> 2 AL        Engineering        Health Professions Physical Sciences
#> 3 AR        Engineering        Biological Science Math/Statistics
#> 4 AZ        Engineering        Health Professions Biological Science
#> 5 CA        Health Professions Engineering        Biological Science
#> 6 CO        Engineering        Not Provided       Health Professions
glimpse(major_by_state_freq_t3)
#> Rows: 53
#> Columns: 4
#> $ stu_state    <chr> "AK", "AL", "AR", "AZ", "CA", "CO", "CT", "DC", "DE", "FL~
#> $ top_major    <chr> "Health Professions", "Engineering", "Engineering", "Engi~
#> $ second_major <chr> "Engineering", "Health Professions", "Biological Science"~
#> $ third_major  <chr> "Biological Science", "Physical Sciences", "Math/Statisti~
str(major_by_state_freq_t3)
#> tibble [53 x 4] (S3: tbl_df/tbl/data.frame)
#>  $ stu_state   : chr [1:53] "AK" "AL" "AR" "AZ" ...
#>   ..- attr(*, "label")= chr "state of prospect"
#>  $ top_major   : chr [1:53] "Health Professions" "Engineering" "Engineering" "Engineering" ...
#>  $ second_major: chr [1:53] "Engineering" "Health Professions" "Biological Science" "Health Professi
#>  $ third_major : chr [1:53] "Biological Science" "Physical Sciences" "Math/Statistics" "Biological S
unique(major_by_state_freq_t3$stu_state)
#>  [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DC" "DE" "FL" "GA" "GU" "HI" "IA" "ID"
#> [16] "IL" "IN" "KS" "KY" "LA" "MA" "MD" "ME" "MI" "MN" "MO" "MS" "MT" "NC" "NE"
#> [31] "NH" "NJ" "NM" "NV" "NY" "OH" "OK" "OR" "PA" "PR" "RI" "SC" "SD" "TN" "TX"
#> [46] "UT" "VA" "VI" "VT" "WA" "WI" "WV" NA
```

Part V: Bonus (up to 10% extra credit)

1. Perform an analysis of your choosing. Feel free to be creative!

```
gender_per_major <- list_native_df %>%
  group_by(stu_gender, stu_major_1_group_text) %>%
  summarise(n_obs = n()) %>%
  arrange(stu_gender, desc(n_obs))
#> `summarise()` has grouped output by 'stu_gender'. You can override using the
#> `.groups` argument.

head(gender_per_major)
#> # A tibble: 6 x 3
#> # Groups:   stu_gender [1]
#>   stu_gender stu_major_1_group_text n_obs
#>   <chr>      <chr>                  <int>
#> 1 F          Health Professions      1648
#> 2 F          Biological Science       990
#> 3 F          Visual/perform Art       575
#> 4 F          Business/Mgmt            539
#> 5 F          Psychology               479
#> 6 F          Engineering              403

gender_per_major_t3 <- gender_per_major %>%
  group_by(stu_gender) %>%
  summarise(top_major = first(stu_major_1_group_text),
            second_major = nth(stu_major_1_group_text, 2),
            third_major = nth(stu_major_1_group_text, 3))

head(gender_per_major_t3)
#> # A tibble: 2 x 4
#>   stu_gender top_major          second_major       third_major
#>   <chr>      <chr>              <chr>              <chr>
#> 1 F          Health Professions Biological Science Visual/perform Art
#> 2 M          Engineering        Business/Mgmt      Health Professions
```

Create a GitHub issue

- Go to the class repository and create a new issue.
- Refer to rclass1 student issues readme for instructions on how to post questions or reflections.
- You are also required to respond to at least one issue posted by another student.
- Paste the url to your issue here: https://github.com/anyone-can-cook/rclass1_student_issues_f23/issues/574
- Paste the url to the issue you responded to here: https://github.com/anyone-can-cook/rclass1_student_issues_f23/issues/572

Knit to pdf and submit problem set

Knit to pdf by clicking the "Knit" button near the top of your RStudio window (icon with blue yarn ball) or drop down and select "Knit to PDF"

- Go to the class website and under the "Readings & Assignments" » "Week 5" tab, click on the "Problem set 5 submission link"
- Submit both .Rmd and pdf files
- Use this naming convention "lastname_firstname_ps#" for your .Rmd and pdf files (e.g. jaquette_ozan_ps5.Rmd & jaquette_ozan_ps5.pdf)