HW 02 - What should I major in?

Moriah Ruggerio

The first step in the process of turning information into knowledge process is to summarize and describe the raw information - the data. In this assignment we explore data on college majors and earnings, specifically the data begin the FiveThirtyEight story "The Economic Guide To Picking A College Major".

These data originally come from the American Community Survey (ACS) 2010-2012 Public Use Microdata Series. While this is outside the scope of this assignment, if you are curious about how raw data from the ACS were cleaned and prepared, see the code FiveThirtyEight authors used.

We should also note that there are many considerations that go into picking a major. Earnings potential and employment prospects are two of them, and they are important, but they don't tell the whole story. Keep this in mind as you analyze the data.

Warm up

Before we introduce the data, let's warm up with some simple exercises.

- Update the YAML, changing the author name to your name, and **knit** the document.
- Commit your changes with a meaningful commit message.
- Push your changes to GitHub.
- Go to your repo on GitHub and confirm that your changes are visible in your Rmd **and** md files. If anything is missing, commit and push again.

Packages

We'll use the **tidyverse** package for much of the data wrangling and visualisation, the **scales** package for better formatting of labels on visualisations, and the data lives in the **fivethirtyeight** package. You will notice that when you go to knit that the packages have not been installed. You can install them manually or click yes when R asks you.

You can load them by running the following in your Console:

```
library(tidyverse)
library(scales)
library(fivethirtyeight)
```

Data

The data can be found in the **fivethirtyeight** package, and it's called `college_recent_grads`. Since the dataset is distributed with the package, we don't need to load it separately; it becomes available to us when we load the package. You can find out more about the dataset by inspecting its documentation, which you can access by running `?college_recent_grads` in the Console or using the Help menu in RStudio to search for `college_recent_grads`. You can also find this information here.

You can also take a quick peek at your data frame and view its dimensions with the `glimpse` function.

```
glimpse(college_recent_grads)
```

Figure 1: Photo by Marleena Garris on Unsplash

```
## Rows: 173
## Columns: 21
## $ rank                          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,~
## $ major_code                    <int> 2419, 2416, 2415, 2417, 2405, 2418, 6202, ~
## $ major                         <chr> "Petroleum Engineering", "Mining And Miner~
## $ major_category                <chr> "Engineering", "Engineering", "Engineering~
## $ total                         <int> 2339, 756, 856, 1258, 32260, 2573, 3777, 1~
## $ sample_size                   <int> 36, 7, 3, 16, 289, 17, 51, 10, 1029, 631, ~
## $ men                           <int> 2057, 679, 725, 1123, 21239, 2200, 2110, 8~
## $ women                         <int> 282, 77, 131, 135, 11021, 373, 1667, 960, ~
## $ sharewomen                    <dbl> 0.1205643, 0.1018519, 0.1530374, 0.1073132~
## $ employed                      <int> 1976, 640, 648, 758, 25694, 1857, 2912, 15~
## $ employed_fulltime             <int> 1849, 556, 558, 1069, 23170, 2038, 2924, 1~
## $ employed_parttime             <int> 270, 170, 133, 150, 5180, 264, 296, 553, 1~
## $ employed_fulltime_yearround <int> 1207, 388, 340, 692, 16697, 1449, 2482, 82~
## $ unemployed                    <int> 37, 85, 16, 40, 1672, 400, 308, 33, 4650, ~
## $ unemployment_rate             <dbl> 0.018380527, 0.117241379, 0.024096386, 0.0~
## $ p25th                         <dbl> 95000, 55000, 50000, 43000, 50000, 50000, ~
## $ median                        <dbl> 110000, 75000, 73000, 70000, 65000, 65000,~
## $ p75th                         <dbl> 125000, 90000, 105000, 80000, 75000, 10200~
## $ college_jobs                  <int> 1534, 350, 456, 529, 18314, 1142, 1768, 97~
## $ non_college_jobs              <int> 364, 257, 176, 102, 4440, 657, 314, 500, 1~
## $ low_wage_jobs                 <int> 193, 50, 0, 0, 972, 244, 259, 220, 3253, 3~
```

The `college_recent_grads` data frame is a trove of information. Let's think about some questions we might want to answer with these data:

- Which major has the lowest unemployment rate?
- Which major has the highest percentage of women?
- How do the distributions of median income compare across major categories?
- Do women tend to choose majors with lower or higher earnings?

In the next section we aim to answer these questions.

Exercises

Which major has the lowest unemployment rate?

In order to answer this question all we need to do is sort the data. We use the `arrange` function to do this, and sort it by the `unemployment_rate` variable. By default `arrange` sorts in ascending order, which is what we want here – we're interested in the major with the *lowest* unemployment rate.

```
college_recent_grads %>%
  arrange(unemployment_rate)
```

```
## # A tibble: 173 x 21
##     rank major_code major           major_category total sample_size   men women
##    <int>      <int> <chr>           <chr>          <int>       <int> <int> <int>
##  1    53       4005 Mathematics An~ Computers & M~   609           7   500   109
##  2    74       3801 Military Techn~ Industrial Ar~   124           4   124     0
##  3    84       3602 Botany          Biology & Lif~  1329           9   626   703
##  4   113       1106 Soil Science    Agriculture &~   685           4   476   209
##  5   121       2301 Educational Ad~ Education        804           5   280   524
##  6    15       2409 Engineering Me~ Engineering     4321          30  3526   795
##  7    20       3201 Court Reporting Law & Public ~  1148          14   877   271
##  8   120       2305 Mathematics Te~ Education      14237         123  3872 10365
```

```
##  9     1       2419 Petroleum Engi~ Engineering     2339          36  2057   282
## 10    65       1100 General Agricu~ Agriculture &~ 10399         158  6053  4346
## # i 163 more rows
## # i 13 more variables: sharewomen <dbl>, employed <int>,
## #   employed_fulltime <int>, employed_parttime <int>,
## #   employed_fulltime_yearround <int>, unemployed <int>,
## #   unemployment_rate <dbl>, p25th <dbl>, median <dbl>, p75th <dbl>,
## #   college_jobs <int>, non_college_jobs <int>, low_wage_jobs <int>
```

This gives us what we wanted, but not in an ideal form. First, the name of the major barely fits on the page. Second, some of the variables are not that useful (e.g. `major_code`, `major_category`) and some we might want front and center are not easily viewed (e.g. `unemployment_rate`).

We can use the `select` function to choose which variables to display, and in which order:

```
Note how easily we expanded our code with adding another step to our pipeline,
with the pipe operator: `%>%`.
```

```
college_recent_grads %>%
  arrange(unemployment_rate) %>%
  select(rank, major, unemployment_rate)
```

```
## # A tibble: 173 x 3
##     rank major                                      unemployment_rate
##    <int> <chr>                                                  <dbl>
##  1    53 Mathematics And Computer Science                      0
##  2    74 Military Technologies                                 0
##  3    84 Botany                                                0
##  4   113 Soil Science                                         0
##  5   121 Educational Administration And Supervision           0
##  6    15 Engineering Mechanics Physics And Science            0.00633
##  7    20 Court Reporting                                      0.0117
##  8   120 Mathematics Teacher Education                        0.0162
##  9     1 Petroleum Engineering                                0.0184
## 10    65 General Agriculture                                  0.0196
## # i 163 more rows
```

Ok, this is looking better, but do we really need to display all those decimal places in the unemployment variable? Not really!

We can use the `percent()` function to clean up the display a bit.

```
college_recent_grads %>%
  arrange(unemployment_rate) %>%
  select(rank, major, unemployment_rate) %>%
  mutate(unemployment_rate = percent(unemployment_rate))
```

```
## # A tibble: 173 x 3
##     rank major                                      unemployment_rate
##    <int> <chr>                                      <chr>
##  1    53 Mathematics And Computer Science           0.00000%
##  2    74 Military Technologies                      0.00000%
##  3    84 Botany                                     0.00000%
##  4   113 Soil Science                              0.00000%
##  5   121 Educational Administration And Supervision 0.00000%
##  6    15 Engineering Mechanics Physics And Science  0.63343%
##  7    20 Court Reporting                            1.16897%
##  8   120 Mathematics Teacher Education              1.62028%
```

```
##  9     1 Petroleum Engineering                      1.83805%
## 10    65 General Agriculture                        1.96425%
## # i 163 more rows
```

Mathmatics and Computer Science, Military Technologies, Botany, Soil Science, and Educational Administration and Supervision all have the lowest unemployment rates of 0.00%.

Which major has the highest percentage of women?

To answer such a question we need to arrange the data in descending order. For example, if earlier we were interested in the major with the highest unemployment rate, we would use the following:

```
The `desc` function specifies that we want `unemployment_rate` in descending order.
```

```
college_recent_grads %>%
  arrange(desc(unemployment_rate)) %>%
  select(rank, major, unemployment_rate)
```

```
## # A tibble: 173 x 3
##     rank major                                        unemployment_rate
##    <int> <chr>                                                   <dbl>
##  1     6 Nuclear Engineering                                     0.177
##  2    90 Public Administration                                   0.159
##  3    85 Computer Networking And Telecommunications              0.152
##  4   171 Clinical Psychology                                     0.149
##  5    30 Public Policy                                          0.128
##  6   106 Communication Technologies                             0.120
##  7     2 Mining And Mineral Engineering                         0.117
##  8    54 Computer Programming And Data Processing               0.114
##  9    80 Geography                                              0.113
## 10    59 Architecture                                           0.113
## # i 163 more rows
```

1. Using what you've learned so far, arrange the data in descending order with respect to proportion of women in a major, and display only the major, the total number of people with major, and proportion of women. Show only the top 3 majors by adding `top_n(3)` at the end of the pipeline.

```
college_recent_grads %>%
  arrange(desc(sharewomen)) %>%         # arrange from most to least for proportion of women (sharewome
  select(major, total, sharewomen) %>%
  mutate(sharewomen = percent(sharewomen)) %>%
  top_n(3)
```

```
## Selecting by sharewomen
```

```
## # A tibble: 3 x 3
##   major                                       total sharewomen
##   <chr>                                       <int> <chr>
## 1 Early Childhood Education                   37589 96.8954%
## 2 Communication Disorders Sciences And Services 38279 96.7998%
## 3 Medical Assisting Services                  11123 92.7807%
```

Early Childhood Education has the greatest proportion of women (96.90%), followed closely by Communication Disorders Sciences and Services (96.80%) and Medical Assisting Services (92.78%).

How do the distributions of median income compare across major categories?

```
A percentile is a measure used in statistics indicating the value below which a given percentage of obs
```

There are three types of incomes reported in this data frame: `p25th`, `median`, and `p75th`. These correspond to the 25th, 50th, and 75th percentiles of the income distribution of sampled individuals for a given major.

2. Why do we often choose the median, rather than the mean, to describe the typical income of a group of people?

The median is often used to describe the typical income of a group of people because it is not heavily affected by outliers (extreme scores) unlike the mean. If 1 or 2 people made significantly more than the rest of the group, the mean could be much higher than what the majority of the people in the group made. The median, however, would still be representative of the majority.

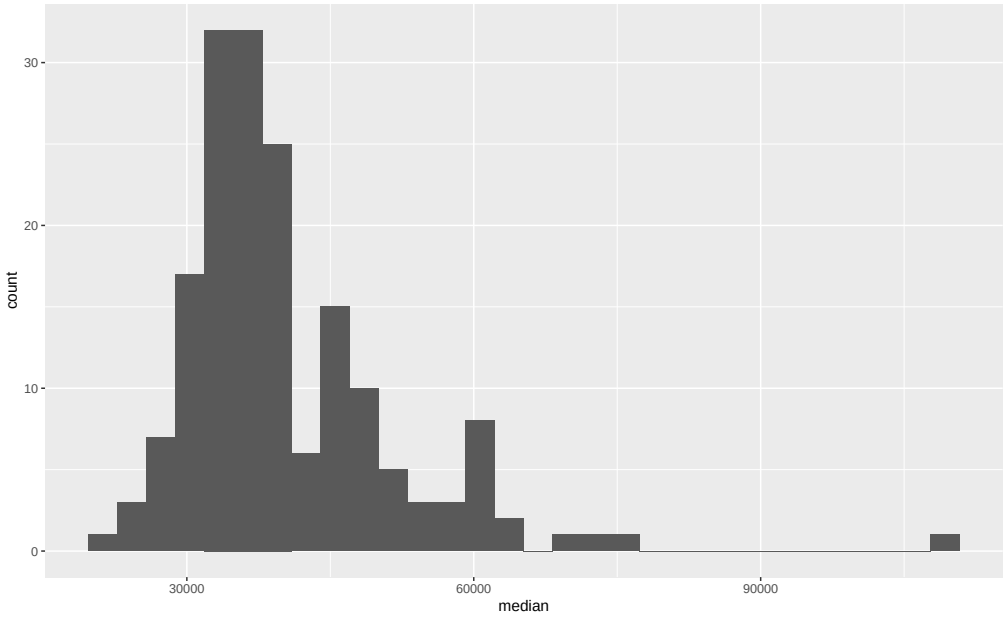
The question we want to answer "How do the distributions of median income compare across major categories?". We need to do a few things to answer this question: First, we need to group the data by `major_category`. Then, we need a way to summarize the distributions of median income within these groups. This decision will depend on the shapes of these distributions. So first, we need to visualize the data.

We use the `ggplot()` function to do this. The first argument is the data frame, and the next argument gives the mapping of the variables of the data to the `aes`thetic elements of the plot.

Let's start simple and take a look at the distribution of all median incomes, without considering the major categories.

```
ggplot(data = college_recent_grads, mapping = aes(x = median)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Along with the plot, we get a message:

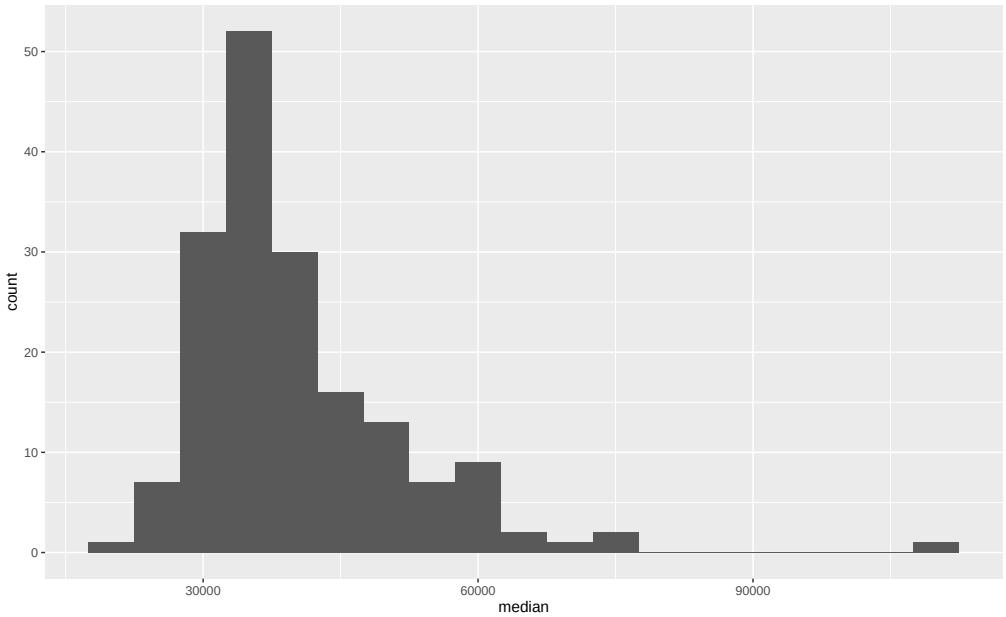
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

This is telling us that we might want to reconsider the binwidth we chose for our histogram – or more accurately, the binwidth we didn't specify. It's good practice to always think in the context of the data and

try out a few binwidths before settling on a binwidth. You might ask yourself: "What would be a meaningful difference in median incomes?" $1 is obviously too little, $10000 might be too high.

3. Try binwidths of $1000 and $5000 and choose one. Explain your reasoning for your choice. Note that the binwidth is an argument for the `geom_histogram` function. So to specify a binwidth of $1000, you would use `geom_histogram(binwidth = 1000)`.

```
ggplot(data = college_recent_grads, mapping = aes(x = median)) +
  geom_histogram(binwidth = 5000)
```



I chose a binwidth of $5000 because $1000 broke up the data too many times. This made it appear to be heavily multimodal, while a binwidth of $5000 showed the overall trends. Also people tend to like "satisfying" numbers more (i.e. 10s and 5s) so it is likely to have more values on these numbers. Breaking it up by by $1000 causes these numbers unrepresentatively stand out.

We can also calculate summary statistics for this distribution using the `summarise` function:

```
college_recent_grads %>%
  summarise(min = min(median), max = max(median),
            mean = mean(median), med = median(median),
            sd = sd(median),
            q1 = quantile(median, probs = 0.25),
            q3 = quantile(median, probs = 0.75))
```

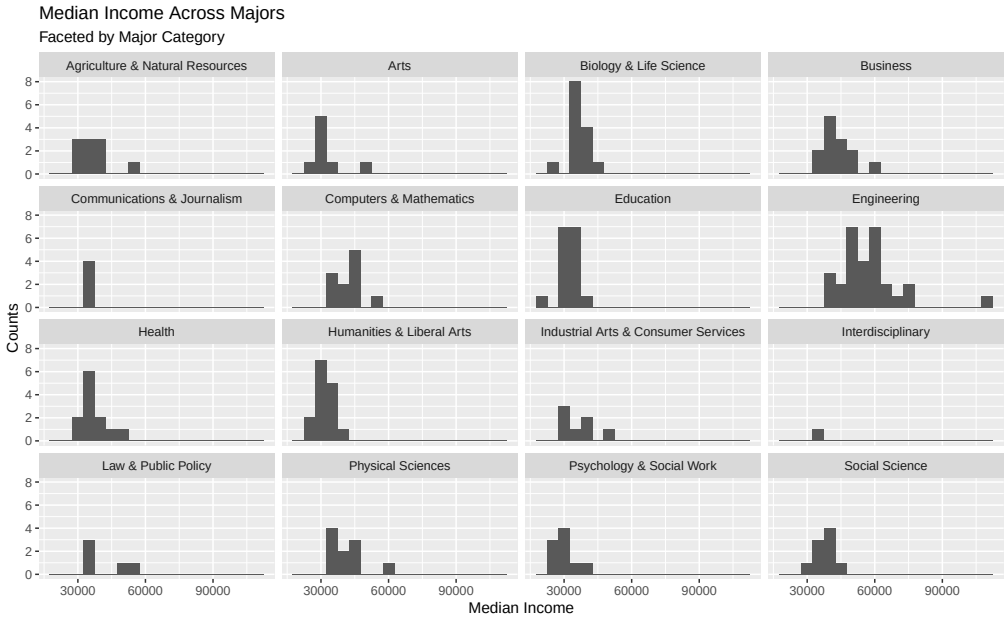
```
## # A tibble: 1 x 7
##     min    max   mean   med     sd    q1    q3
##   <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 22000 110000 40151. 36000 11470. 33000 45000
```

4. Based on the shape of the histogram you created in the previous exercise, determine which of these summary statistics is useful for describing the distribution. Write up your description (remember shape, center, spread, any unusual observations) and include the summary statistic output as well.

Based on the shape of the histogram, the median and interquartile range (q1 and q3) are most helpful in describing the distribution. The distribution is skewed to the high end and centered around the median ($36000). Most of the data is spread from $33000 to $45000 (q1 and q3). There is one extreme value at around $110,000.

5. Plot the distribution of `median` income using a histogram, faceted by `major_category`. Use the `binwidth` you chose in the earlier exercise.

```
ggplot(data = college_recent_grads, mapping = aes(x = median)) +
  geom_histogram(binwidth = 5000) +
  facet_wrap(~major_category) +
  labs(title = "Median Income Across Majors", subtitle = "Faceted by Major Category", x = "Median Income
```



Now that we've seen the shapes of the distributions of median incomes for each major category, we should have a better idea for which summary statistic to use to quantify the typical median income.

6. Which major category has the highest typical (you'll need to decide what this means) median income? Use the partial code below, filling it in with the appropriate statistic and function. Also note that we are looking for the highest statistic, so make sure to arrange in the correct direction.

```
college_recent_grads %>%
  group_by(major_category) %>%
  summarise(median_income = median(median)) %>%    #using median to determine highest "typical" income
  arrange(desc(median_income))     #shows results from greatest to least
```

```
## # A tibble: 16 x 2
##    major_category                    median_income
##    <chr>                                    <dbl>
##  1 Engineering                              57000
##  2 Computers & Mathematics                  45000
##  3 Business                                 40000
##  4 Physical Sciences                        39500
##  5 Social Science                           38000
##  6 Biology & Life Science                   36300
##  7 Law & Public Policy                      36000
##  8 Agriculture & Natural Resources          35000
##  9 Communications & Journalism              35000
## 10 Health                                   35000
## 11 Industrial Arts & Consumer Services      35000
## 12 Interdisciplinary                        35000
## 13 Education                                32750
```

```
## 14 Humanities & Liberal Arts                    32000
## 15 Arts                                         30750
## 16 Psychology & Social Work                     30000
```

By median (to account for the skewed distributions), Engineering has the highest typical income (median income = $5700).

7. Which major category is the least popular in this sample? To answer this question we use a new function called `count`, which first groups the data and then counts the number of observations in each category (see below). Add to the pipeline appropriately to arrange the results so that the major with the lowest observations is on top.

```
college_recent_grads %>%
  count(major_category) %>%
  arrange(n)                    # shows results from least to greatest
```

```
## # A tibble: 16 x 2
##    major_category                          n
##    <chr>                               <int>
##  1 Interdisciplinary                       1
##  2 Communications & Journalism             4
##  3 Law & Public Policy                     5
##  4 Industrial Arts & Consumer Services     7
##  5 Arts                                    8
##  6 Psychology & Social Work                9
##  7 Social Science                          9
##  8 Agriculture & Natural Resources        10
##  9 Physical Sciences                      10
## 10 Computers & Mathematics                11
## 11 Health                                 12
## 12 Business                               13
## 13 Biology & Life Science                 14
## 14 Humanities & Liberal Arts              15
## 15 Education                              16
## 16 Engineering                            29
```

The least popular major category in this sample is Interdisciplinary (n = 1).

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

All STEM fields aren't the same

One of the sections of the FiveThirtyEight story is "All STEM fields aren't the same". Let's see if this is true.

First, let's create a new vector called `stem_categories` that lists the major categories that are considered STEM fields.

```
stem_categories <- c("Biology & Life Science",
                     "Computers & Mathematics",
                     "Engineering",
                     "Physical Sciences")
```

Then, we can use this to create a new variable in our data frame indicating whether a major is STEM or not.

```
college_recent_grads <- college_recent_grads %>%
  mutate(major_type = ifelse(major_category %in% stem_categories, "stem", "not stem"))
```

Let's unpack this: with `mutate` we create a new variable called `major_type`, which is defined as `"stem"` if the `major_category` is in the vector called `stem_categories` we created earlier, and as `"not stem"` otherwise.

`%in%` is a **logical operator**. Other logical operators that are commonly used are

Operator	Operation
`x < y`	less than
`x > y`	greater than
`x <= y`	less than or equal to
`x >= y`	greater than or equal to
`x != y`	not equal to
`x == y`	equal to
`x %in% y`	contains
x | y	or
`x & y`	and
`!x`	not

We can use the logical operators to also `filter` our data for STEM majors whose median earnings is less than median for all majors' median earnings, which we found to be $36,000 earlier.

```
college_recent_grads %>%
  filter(
    major_type == "stem",
    median < 36000
  )
```

```
## # A tibble: 10 x 22
##     rank major_code major        major_category  total sample_size    men  women
##    <int>      <int> <chr>        <chr>           <int>       <int>  <int>  <int>
##  1    93       1301 Environment~ Biology & Lif~  25965         225  10787  15178
##  2    98       5098 Multi-Disci~ Physical Scie~  62052         427  27015  35037
##  3   102       3608 Physiology   Biology & Lif~  22060          99   8422  13638
##  4   106       2001 Communicati~ Computers & M~  18035         208  11431   6604
##  5   109       3611 Neuroscience Biology & Lif~  13663          53   4944   8719
##  6   111       5002 Atmospheric~ Physical Scie~   4043          32   2744   1299
##  7   123       3699 Miscellaneo~ Biology & Lif~  10706          63   4747   5959
##  8   124       3600 Biology      Biology & Lif~ 280709        1370 111762 168947
##  9   133       3604 Ecology      Biology & Lif~   9154          86   3878   5276
## 10   169       3609 Zoology      Biology & Lif~   8409          47   3050   5359
## # i 14 more variables: sharewomen <dbl>, employed <int>,
## #   employed_fulltime <int>, employed_parttime <int>,
## #   employed_fulltime_yearround <int>, unemployed <int>,
## #   unemployment_rate <dbl>, p25th <dbl>, median <dbl>, p75th <dbl>,
## #   college_jobs <int>, non_college_jobs <int>, low_wage_jobs <int>,
## #   major_type <chr>
```

8. Which STEM majors have median salaries equal to or less than the median for all majors' median earnings? Your output should only show the major name and median, 25th percentile, and 75th percentile earning for that major as and should be sorted such that the major with the highest median earning is on top.

```
college_recent_grads %>%
  filter(
    major_type == "stem",
    median < 36000) %>%
  select(major, median, p25th, p75th) %>%
  arrange(desc(median))
```

```
## # A tibble: 10 x 4
##    major                                median p25th p75th
##    <chr>                                 <dbl> <dbl> <dbl>
##  1 Environmental Science                 35600 25000 40200
##  2 Multi-Disciplinary Or General Science 35000 24000 50000
##  3 Physiology                            35000 20000 50000
##  4 Communication Technologies            35000 25000 45000
##  5 Neuroscience                          35000 30000 44000
##  6 Atmospheric Sciences And Meteorology  35000 28000 50000
##  7 Miscellaneous Biology                 33500 23000 48000
##  8 Biology                               33400 24000 45000
##  9 Ecology                               33000 23000 42000
## 10 Zoology                               26000 20000 39000
```

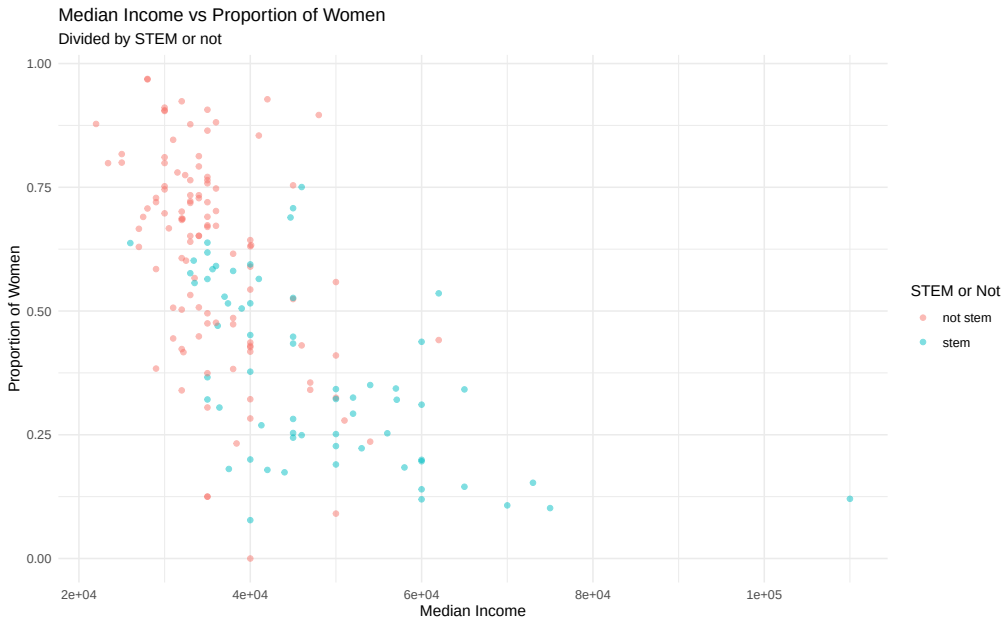
Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

What types of majors do women tend to major in?

9. Create a scatterplot of median income vs. proportion of women in that major, coloured by whether the major is in a STEM field or not. Describe the association between these three variables.

```
college_recent_grads %>%
  ggplot(mapping = aes(x = median, y = sharewomen, color = major_type)) +        #creates scatterplot w
  geom_point(alpha = 0.5) +
  labs(title = "Median Income vs Proportion of Women", subtitle = "Divided by STEM or not", x = "Median
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Non-STEM majors typically have more women than men involved. These fields typically make less than STEM majors. While some STEM majors have a proportion of women over 0.50, most STEM fields have more men. STEM majors also appear to have a higher median income than non-STEM. The highest median income STEM majors also primarily male dominated. Except for 2 points, all majors with median incomes over $50,000 have a higher proportion of men than women in them.

Further exploration

10. Ask a question of interest to you, and answer it using summary statistic(s) and/or visualization(s).

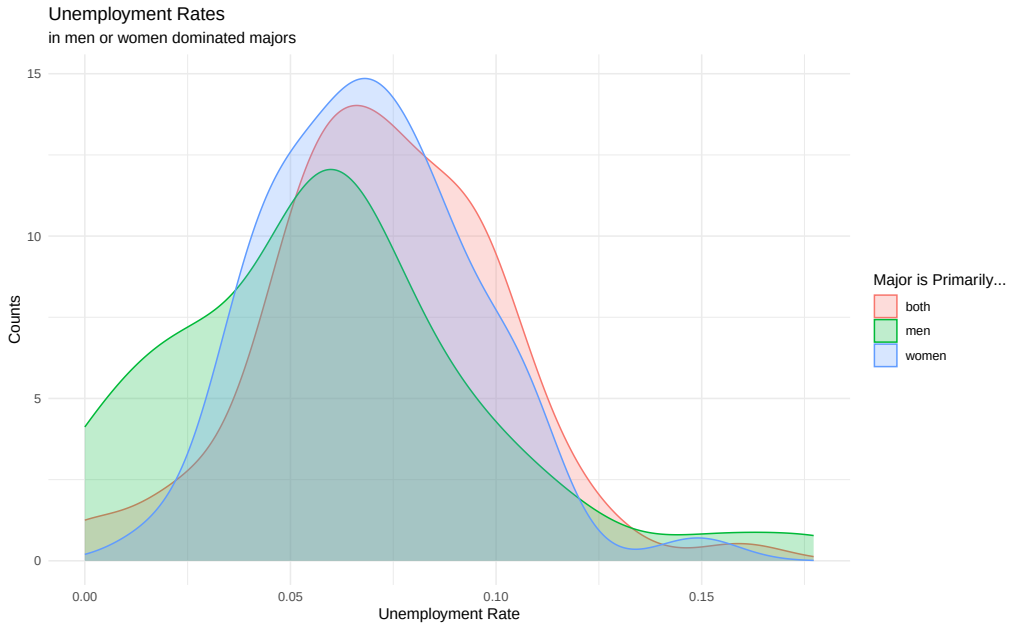
Do primarily women dominated majors (proportion of women >= 0.66) have a higher or lower unemployment rate than male dominated majors (proportion of women <= 0.33) and those that are split (0.33 < proportion of women > 0.66)?

```
college_recent_grads <- college_recent_grads %>%
  mutate(dominated_by = if_else(sharewomen >= 0.66, "women",          # first if then statement for wome
                        if_else(sharewomen <= 0.33, "men", "both")))     # second if then statement (lo
  #creates a new variable for whether the major is primarily men, women, or both

college_recent_grads %>%
  filter(dominated_by == "women" | dominated_by == "men" | dominated_by == "both") %>%
  group_by(dominated_by) %>%
  summarise(med = median(unemployment_rate),,
            q1 = quantile(unemployment_rate, probs = 0.25),
            q3 = quantile(unemployment_rate, probs = 0.75)) %>%
  arrange(med)
```

```
## # A tibble: 3 x 4
##   dominated_by    med     q1     q3
##   <chr>         <dbl>  <dbl>  <dbl>
## 1 men          0.0582 0.0289 0.0747
## 2 women        0.0687 0.0509 0.0850
## 3 both         0.0721 0.0553 0.0925
```

```r
college_recent_grads %>%
  filter(dominated_by == "women" | dominated_by == "men" | dominated_by == "both") %>%
  ggplot(mapping = aes(x = unemployment_rate, color = dominated_by, fill = dominated_by)) +         #cre
  geom_density(alpha = 0.25) +
  labs(title = "Unemployment Rates", subtitle = "in men or women dominated majors", x = "Unemployment Ra
  theme_minimal()
```



The median unemployment rate is lowest in majors that are dominated by men (0.058), followed by women (0.069), and highest when it is split (0.072). Looking at the chart, majors that are dominated by women and both are are fairly uniformly distributed with a couple of extremely high values. However, while the centers of the distribution are fairly close together, there are substantially more low values in the men dominated majors. This shows up in the IQRs (Men: q1 = 0.29, q2 = 0.75; Women: q1 = 0.051, q2 = 0.085; Both: q1 = 0.055, q2 = 0.092).

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards and review the md document on GitHub to make sure you're happy with the final state of your work.*