Week One, Class Two

ML

1/10/2022

Load Data and Packages

We'll continue using Table 2 from Opportunity Insights' Mobility Report Cards paper. Load the data as a data frame called `mobility` and load the `tidyverse` packages.

```
mobility <-  read.csv("https://opportunityinsights.org/wp-content/uploads/2018/04/mrc_table2.csv", stri

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Reviewing Mobility Tables

We ended class yesterday by making a function to create a mobility table for any college in our data frame. Let's rerun the function so we can use it again.

```
mobility_table <- function(x) {
  df <- mobility |>
    filter(name == x)

  pq1 <- c(df$kq1_cond_parq1,
           df$kq2_cond_parq1,
           df$kq3_cond_parq1,
           df$kq4_cond_parq1,
           df$kq5_cond_parq1)

  pq2 <- c(df$kq1_cond_parq2,
           df$kq2_cond_parq2,
           df$kq3_cond_parq2,
           df$kq4_cond_parq2,
```

```
          df$kq5_cond_parq2)

  pq3 <- c(df$kq1_cond_parq3,
           df$kq2_cond_parq3,
           df$kq3_cond_parq3,
           df$kq4_cond_parq3,
           df$kq5_cond_parq3)

  pq4 <- c(df$kq1_cond_parq4,
           df$kq2_cond_parq4,
           df$kq3_cond_parq4,
           df$kq4_cond_parq4,
           df$kq5_cond_parq4)

  pq5 <- c(df$kq1_cond_parq5,
           df$kq2_cond_parq5,
           df$kq3_cond_parq5,
           df$kq4_cond_parq5,
           df$kq5_cond_parq5)

  college_table <- rbind(pq1, pq2, pq3, pq4, pq5)

  colnames(college_table) <- c("kq1", "kq2", "kq3", "kq4", "kq5")

  round(college_table,3)
}
```

And use this function to get Middlebury's mobility table.

```
mobility_table("Middlebury College")
```

```
##       kq1   kq2   kq3   kq4   kq5
## pq1 0.098 0.103 0.077 0.175 0.546
## pq2 0.121 0.084 0.224 0.114 0.457
## pq3 0.094 0.150 0.081 0.208 0.468
## pq4 0.098 0.090 0.099 0.239 0.474
## pq5 0.075 0.100 0.097 0.150 0.578
```

Remember that the top right cell is what Chetty et al call the *success rate*.

What other colleges did you try? Any colleges from the Leonhardt article?

Adapting the Mobility Table Function

In the mobility table above, each row sums to 1. That helps us summarize the destination quintiles for students from each origin quintile. But that doesn't help us know what proportion of *all* students at a college are in each cell. We need to be able to compare these other proportions to know which colleges have the highest proportions of their students experiencing mobility.

How can we adapt the function we already created to write a new function (called `mobility_table_2`) that will give us a table showing the proportion of all students at a college in each pq-kq cell?

REPLACE THIS LINE WITH YOUR CODE CHUNK

```
mobility_table_2 <- function(x) {
  df <- mobility |>
    filter(name == x)

  pq1 <- c(df$kq1_cond_parq1,
           df$kq2_cond_parq1,
           df$kq3_cond_parq1,
           df$kq4_cond_parq1,
           df$kq5_cond_parq1) * df$par_q1 # par_q1 is the *access rate*!

  pq2 <- c(df$kq1_cond_parq2,
           df$kq2_cond_parq2,
           df$kq3_cond_parq2,
           df$kq4_cond_parq2,
           df$kq5_cond_parq2) * df$par_q2

  pq3 <- c(df$kq1_cond_parq3,
           df$kq2_cond_parq3,
           df$kq3_cond_parq3,
           df$kq4_cond_parq3,
           df$kq5_cond_parq3) * df$par_q3

  pq4 <- c(df$kq1_cond_parq4,
           df$kq2_cond_parq4,
           df$kq3_cond_parq4,
           df$kq4_cond_parq4,
           df$kq5_cond_parq4) * df$par_q4

  pq5 <- c(df$kq1_cond_parq5,
           df$kq2_cond_parq5,
           df$kq3_cond_parq5,
           df$kq4_cond_parq5,
           df$kq5_cond_parq5) * df$par_q5

  college_table <- rbind(pq1, pq2, pq3, pq4, pq5)

  colnames(college_table) <- c("kq1", "kq2", "kq3", "kq4", "kq5")

  round(college_table,3)
}
```

Run this same function the same way as the original function by adding a college's name.

```
mobility_table_2("Middlebury College")
```

```
##       kq1   kq2   kq3   kq4   kq5
## pq1 0.002 0.002 0.002 0.004 0.013
## pq2 0.005 0.003 0.009 0.005 0.018
## pq3 0.007 0.012 0.006 0.017 0.037
## pq4 0.010 0.009 0.010 0.024 0.047
## pq5 0.056 0.076 0.074 0.114 0.438
```

The top right cell of this table is what Chetty et al call the *mobility rate*.

Comparing Mobility Rates

What colleges have the highest mobility rates? There is already a variable in the data frame measuring the mobility rate: `mr_kq5_pq1`. The easiest thing to do at an early stage of analysis is to open the spreadsheet view of the data frame and sort by the values of that variable.

```
#View(mobility)
```

Tomorrow we'll see the `select()` function which helps us choose to keep certain columns. Here's a preview:

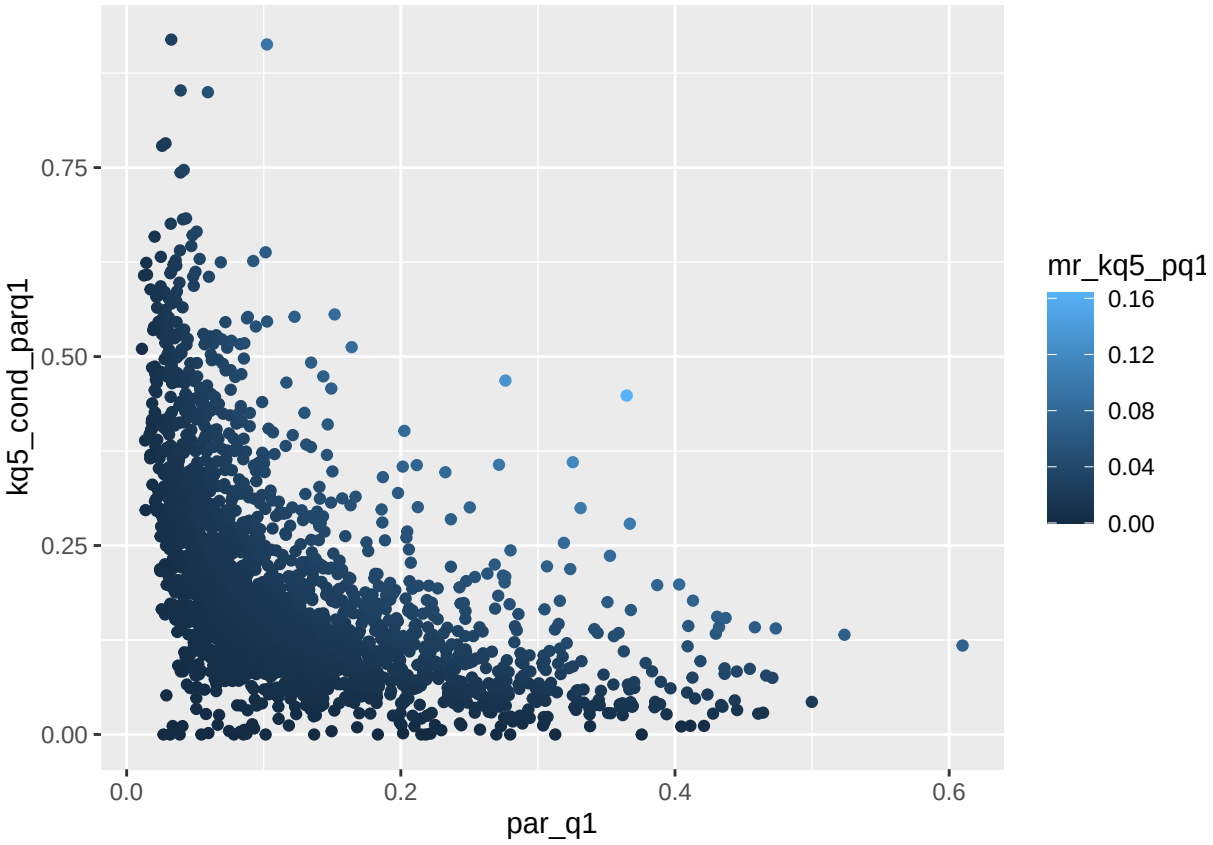
```
high_mobility_rate <- select(mobility, # the full data frame
                             name, tier_name, mr_kq5_pq1) # the variables to keep

#View(high_mobility_rate)
```

How can we use some of the geom types we saw this morning to visualize the relationship across the variables we have been using? Try a scatterplot of access rates (`par_q1`) by success rates (`kq5_cond_parq1`) with points shaded by their mobility rates (`mr_kq5_pq1`).

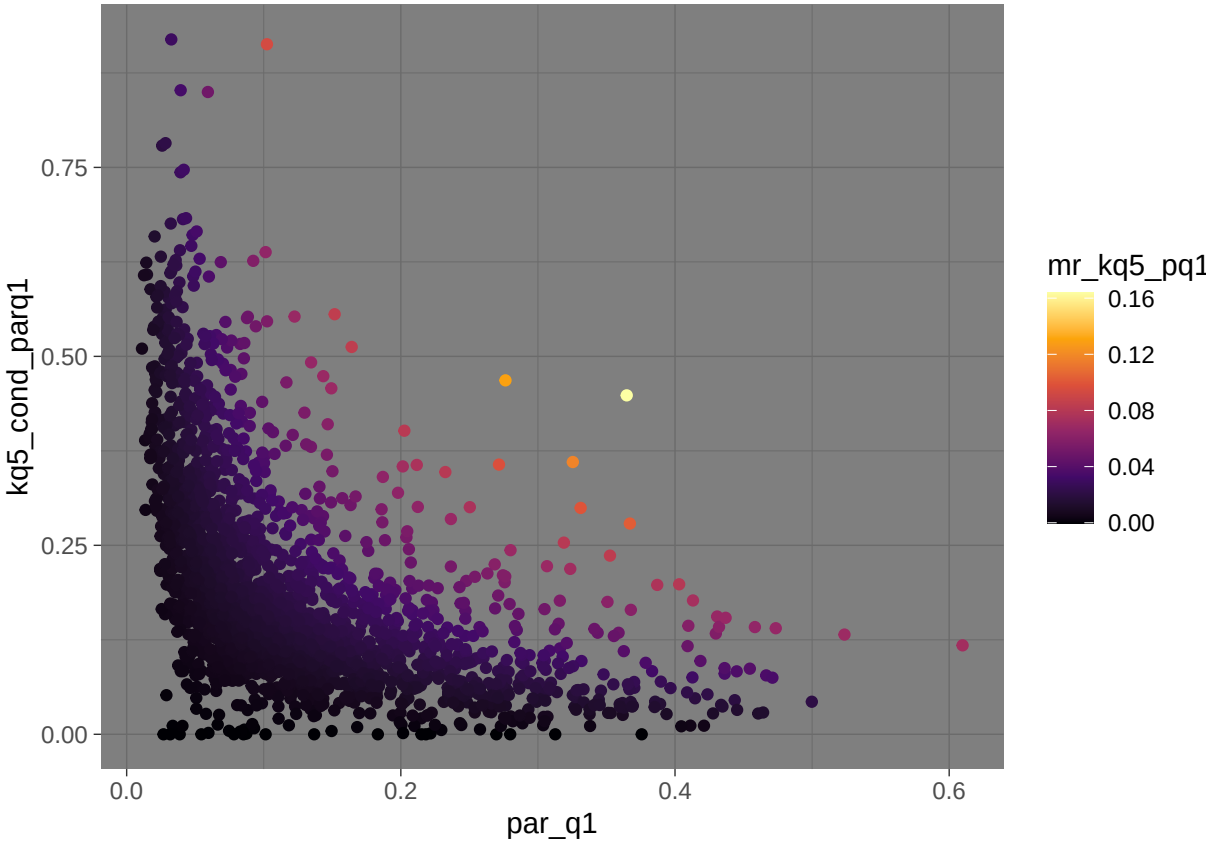
REPLACE THIS LINE WITH YOUR CODE CHUNK

```
ggplot(mobility, aes(x = par_q1, y = kq5_cond_parq1, color = mr_kq5_pq1)) +
  geom_point()
```



How could we improve this visualization? Let's try changing the scale of the colors and changing the background.

```
ggplot(mobility, aes(x = par_q1, y = kq5_cond_parq1, color = mr_kq5_pq1)) +
  geom_point() +
  scale_color_viridis_c(option = "inferno") + # a scale built in to ggplot
 theme_dark() # helpful for the light viridis colors to stand out +
```

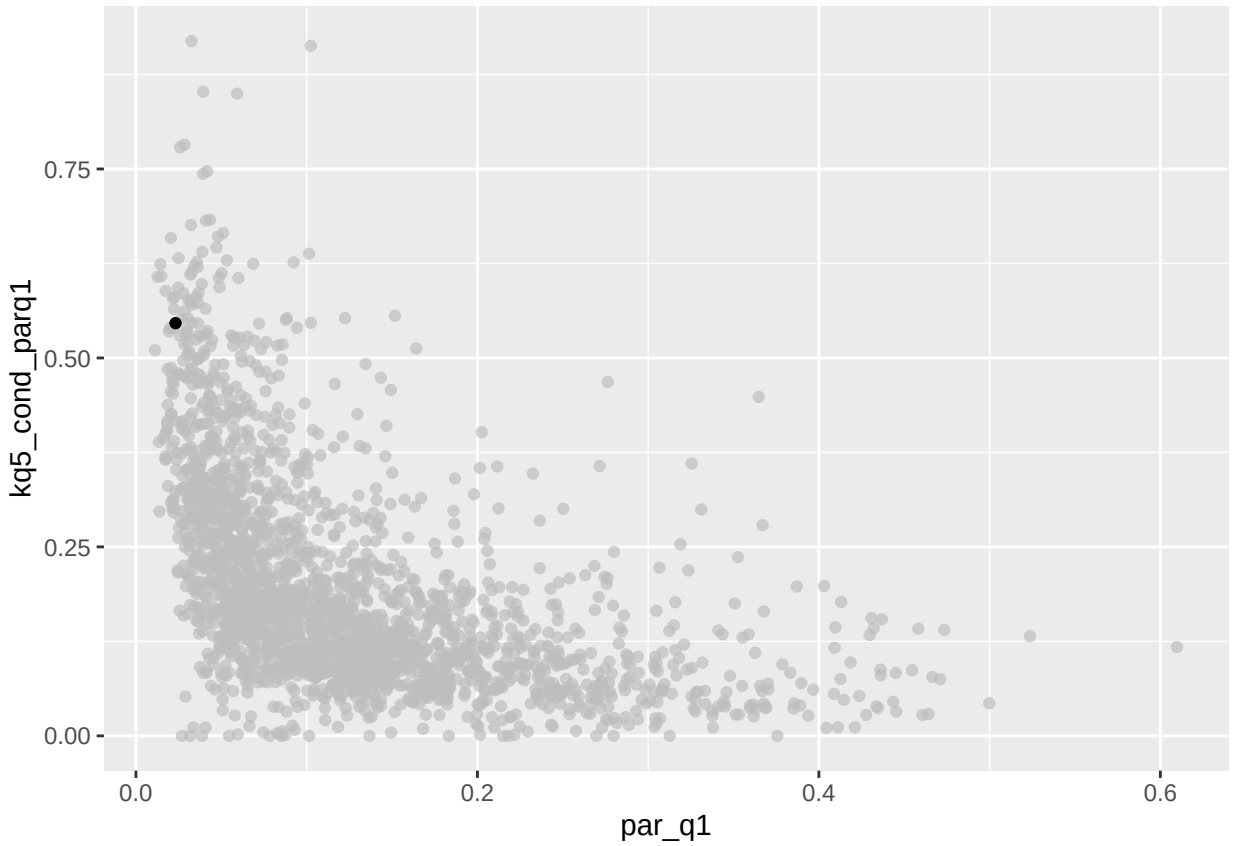


We might also want to highlight specific points based on values of a variable. I like the `gghighlight` package for this. Install and load the package.

```
#install.packages("gghighlight")
library(gghighlight)
```

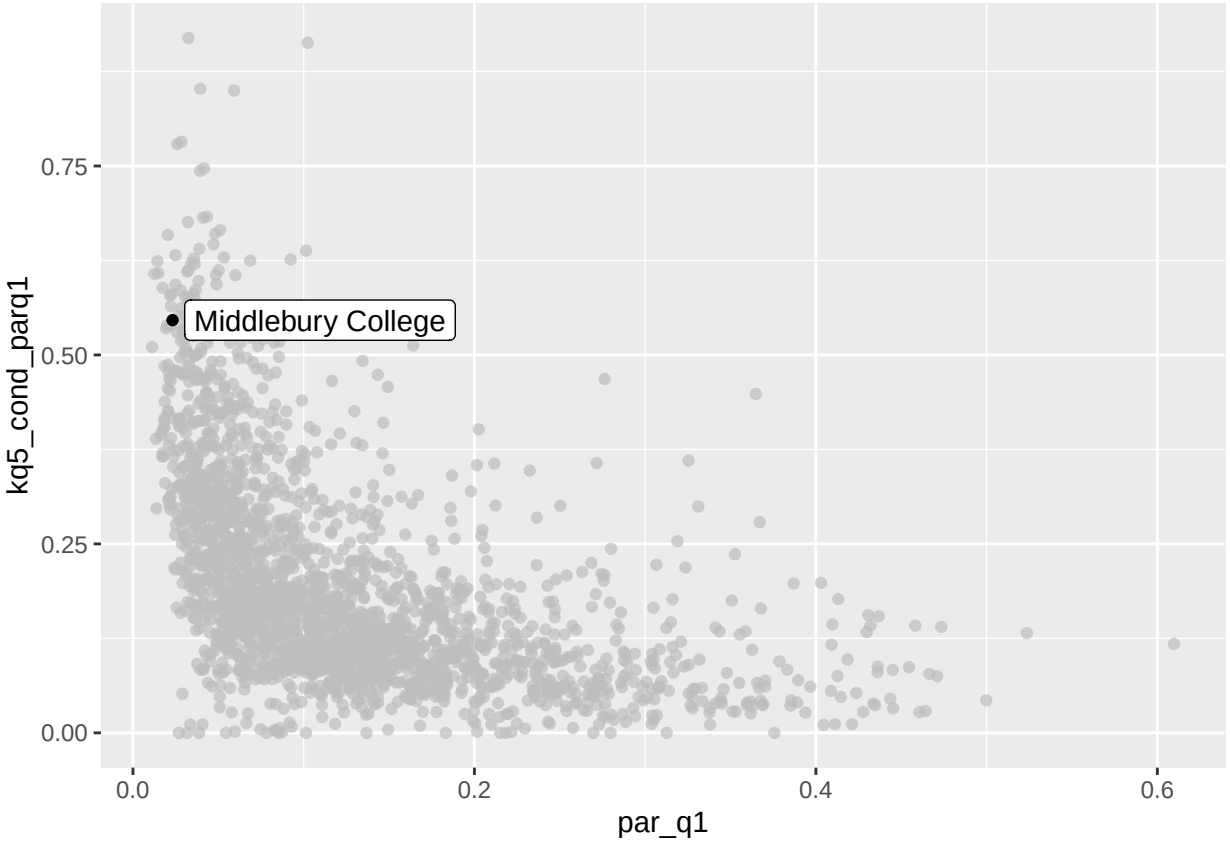
The `gghighlight()` function works like the `filter()` function. Simply assert the conditions for the highlighted layer.

```
ggplot(mobility, aes(x = par_q1, y = kq5_cond_parq1)) + geom_point() +
  gghighlight(name == "Middlebury College")
```



Add the `label_key =` option for a label based on the value of a variable:

```
ggplot(mobility, aes(x = par_q1, y = kq5_cond_parq1)) + geom_point() +
  gghighlight(name == "Middlebury College", label_key = name)
```



What other points might be interesting to highlight?

"America's Great Working Class Colleges"

What do we need to replicate this table from Leonhardt's column:

https://raw.githubusercontent.com/mjclawrence/soci1230_w22/main/notes_slides/1230_figures/mrc_upward_mobility_top10.png)

Pay attention to the notes at the top and the bottom. What variables do we need? What functions do they make you think of?

We'll put our table in a new data frame called `top10mobility` based on the existing `mobility` data frame. Start by using `filter()` to choose the rows we want to keep.

```
top10mobility <- filter(mobility, count>500 & par_q1>.1)
```

Then there are a few new things to learn to finish the table.

```
top10mobility |> # this "pipe" allows us to chain lots of functions together
  mutate(kq345_cond_parq1 = kq3_cond_parq1 +
           kq4_cond_parq1 + kq5_cond_parq1) |>  # to create a new variable
  arrange(desc(kq345_cond_parq1)) |> # to sort by highest value to lowest value of a variable
  select(name, kq345_cond_parq1) |> # to keep specific columns
  top_n(10, kq345_cond_parq1) # to keep certain number of observations
```

```
##                                          name kq345_cond_parq1
```

```
## 1               New Jersey Institute Of Technology       0.8508686
## 2                                  Pace University       0.8227316
## 3          California State University, Bakersfield       0.8221039
## 4                  University Of California, Irvine       0.8087090
## 5  California State Polytechnic University, Pomona       0.8078726
## 6                    Xavier University Of Louisiana       0.7949287
## 7       State University Of New York At Stony Brook       0.7884463
## 8                           San Jose State University       0.7881660
## 9                    CUNY Bernard M. Baruch College       0.7862932
## 10          California State University, Long Beach       0.7809506
```