exercise-Instructor

July 1, 2021

1 Data Science Exercise

1.1 WICS - SYP 21

Notebook adapted from UGBA 88 course materials

1.2 Economic Mobility at Universities

In 2017, a team of researchers used anonymized data from the federal government to publish statistics for each college in the U.S. on the distribution of students' earnings in their thirties and their parents' incomes. We showed that students from low-income families have excellent long-term outcomes after attending selective schools, but that there are very few low-income students at these schools.

This work was highlighted in several news sites:

- NYTimes including interactive visualizations
- Vox

As many of you may be looking at colleges to attend in the near future. We can see how Data Science can help answer questions.

1.2.1 Goals

In this exercise, we will analyze the data looking at College Mobility. We will focus on public universities and community colleges in Michigan. An important justification for public spending on higher education is that colleges and universities may be seen as the 'engines of social mobility'.

We will do three things. First, we will investigate how access, success, and upward mobility rates vary across institutions. Second, we will explore how access has changed over time, as Michigan's spending on public higher education has declined or stagnated. Third, we will write a function that generates a Report Card for a provided institution.

The exercises are intended to illustrate how descriptive statistics alone can provide valuable insights and motivate new questions.

1.2.2 Table of Contents

Dependencies:

[1]:
```
from datascience import *
import numpy as np

#These lines set up graphing capabilities.
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

1.3 1. Comparing Outcomes Mobility Across Institutions

The first dataset we'll use has one row of data for each college and university in the US.

(Though we discuss the columns we'll use in this lab, look here for more documentation on the remaining contents of these data.)

First, let's load the data and the specific columns we'll use in this lab.

[4]:
```
mobility = Table.read_table("data/mrc_q1.csv")

print("Table Dimensions:", mobility.num_columns , "X" , mobility.num_rows)
mobility.show(5)
```

```
Table Dimensions: 10 X 2202
```

```
<IPython.core.display.HTML object>
```

In this exercise, we will focus on Michigan public institutions. Let's filter the data to reflect this.

[6]:
```
mi_pub_mobility = mobility.where('type', are.equal_to('Public')).where('state',␣
↪are.equal_to('MI'))

print("Table Dimensions:", mi_pub_mobility.num_columns , "X" , mi_pub_mobility.
↪num_rows)
mi_pub_mobility.show(20)
```

```
Table Dimensions: 10 X 40
```

```
<IPython.core.display.HTML object>
```

*Note: See how we can "chained" **.where** statements?*

This is because calling .where() on a table object, returns another table object, so you can use as many .where() statements as you like that each filter out rows of the table.

We are left with a total of 40 institutions.

1.4 Exploring the Data

We will first describe the distributions of *access, success rates, and mobility rates* across institutions. We use the same definitions of these terms used in the paper and described in lecture:

- **access:** the percentage of students enrolled that are 'low income'–those whose parents' income is in the bottom quintile (bottom 20%) of the parental income distribution. Note: values range from 0 to 100.
- **success:** the percentage of low income students with post-graduation incomes in the top quintile (top 20%) of the student income distribution, measured at age 32-34.
- **mobility:** the percentage of students enrolled that are both 'low income' and later have earnings in the top quintile (top 20%) of the student income distribution.

Recall that `mobility` = `access` × `success`. Hence, institutions with high mobility will tend to have more low income students and high 'success' rates with those students.

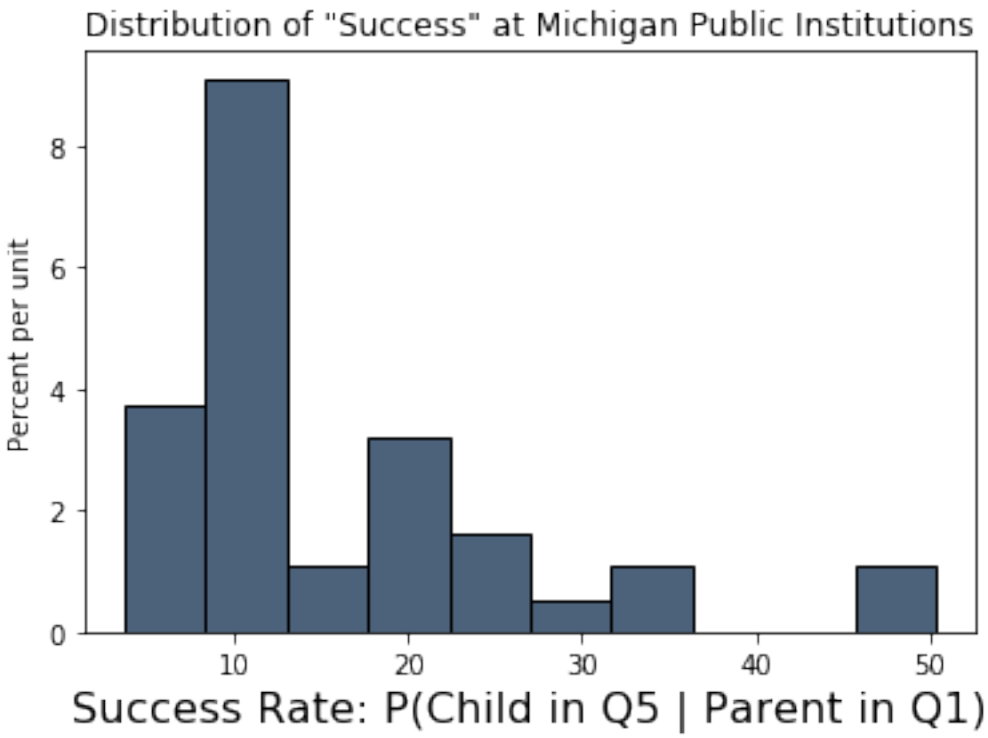
1.4.1 Success Rates

EXERCISE 1.1: Plot a histogram of `success` across institutions.

Hint: Look at the datascience documentation here for how to use `.hist` to graph a histogram.

[7]:
```
#create histogram of success column
mi_pub_mobility.hist("success")

#the code below will label the axes and title of your histogram
plt.title('Distribution of "Success" at Michigan Public Institutions')
plt.xlabel('Success Rate: P(Child in Q5 | Parent in Q1)')
```

[7]:
```
Text(0.5, 0, 'Success Rate: P(Child in Q5 | Parent in Q1)')
```



Notice that one percent of the institutions have a substantially larger `success` rate than the rest. This type of data point(s), one that does not fit the overall pattern of the data, is often referred to as an **outlier**.

EXERCISE 1.2: What is that outlier?

To find this, we can filter the table to look at rows where `success` is sufficiently large. Alternatively you can order the table by success and select the top row. The function `.where()` and `are.above(<value>)` may be useful.

Set `success_outlier` to the name of the outlier institution.

[9]:
```
success_outlier = mi_pub_mobility.where("success", are.above(40)).column("name")
print(success_outlier)
```

```
['Michigan Technological University' 'University Of Michigan - Ann Arbor']
```

EXERCISE 1.3: Look at some descriptive statistics of `success`.

Compute the mean, standard deviation, 25th, 50th (median) and 75th percentiles of the column `success`.

[10]:
```
#note: the function np.std(x) takes an array x and calculates the standard␣
↪deviation.
```

```
#note: the function np.percentile(x, A) takes an array x and calculates␣
 ↪percentiles of x corresponding
#to the values of an array A, ranging from 0-100. Your call of np.percentile␣
 ↪will take
#the form: np.percentile(x, [a1, a2, a3])

success_mean = np.mean(mi_pub_mobility.column("success"))
success_std = np.std(mi_pub_mobility.column("success"))
success_percentiles = np.percentile(mi_pub_mobility.column("success"), [25, 50,␣
 ↪75])

#note: success_percentiles should be an array of 3 values

print('mean:', success_mean)
print('standard deviation:', success_std)
print('percentiles:', success_percentiles)
```

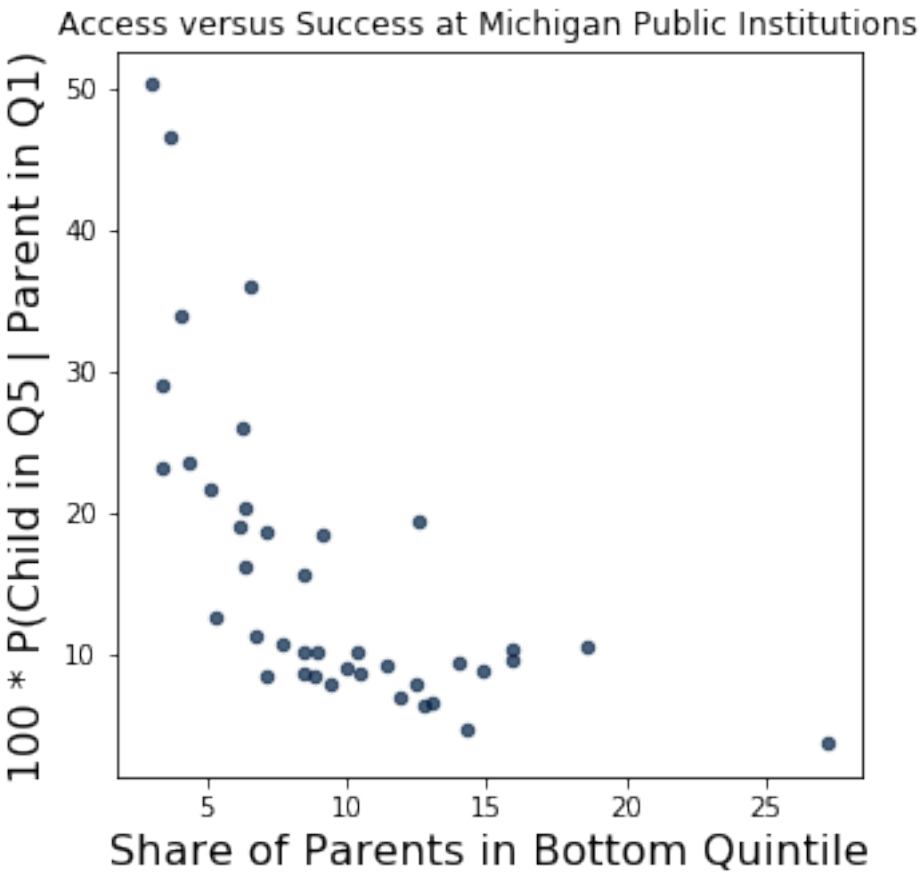
```
mean: 15.7554551825
standard deviation: 10.782666437290377
percentiles: [ 8.69226215 10.4798045  19.6466405 ]
```

EXERCISE 1.4: Next, let's examine the relationship between `access` and `success`. Create a scatterplot with `access` on the horizontal axis and `success` on the vertical access. Try the function `scatter` that can be called on a table.

Table Functions Reference

[12]:
```
#create scatter plot
mi_pub_mobility.scatter("access", "success")

#the code below will label the axes and title of your scatter plot
plt.title('Access versus Success at Michigan Public Institutions')
plt.xlabel('Share of Parents in Bottom Quintile')
plt.ylabel('100 * P(Child in Q5 | Parent in Q1)')
```

[12]:
```
Text(0, 0.5, '100 * P(Child in Q5 | Parent in Q1)')
```



Interestingly, despite the clear relationship between `access` and `success` you've noted above, there is still a lot of variation in `access` among institutions with similar `success` rates. You can see that from the following figure (which includes all US colleges and universities, not just public Michigan schools):

Among schools at the 75th percentile of `success`, the stadard deviation is relatively large at 6.88%. This suggests an interesting policy question: how are institutions producing students of similar 'quality' (as measured by earnings) yet providing very different levels of access? What can be learned from the more accessible colleges and universities?

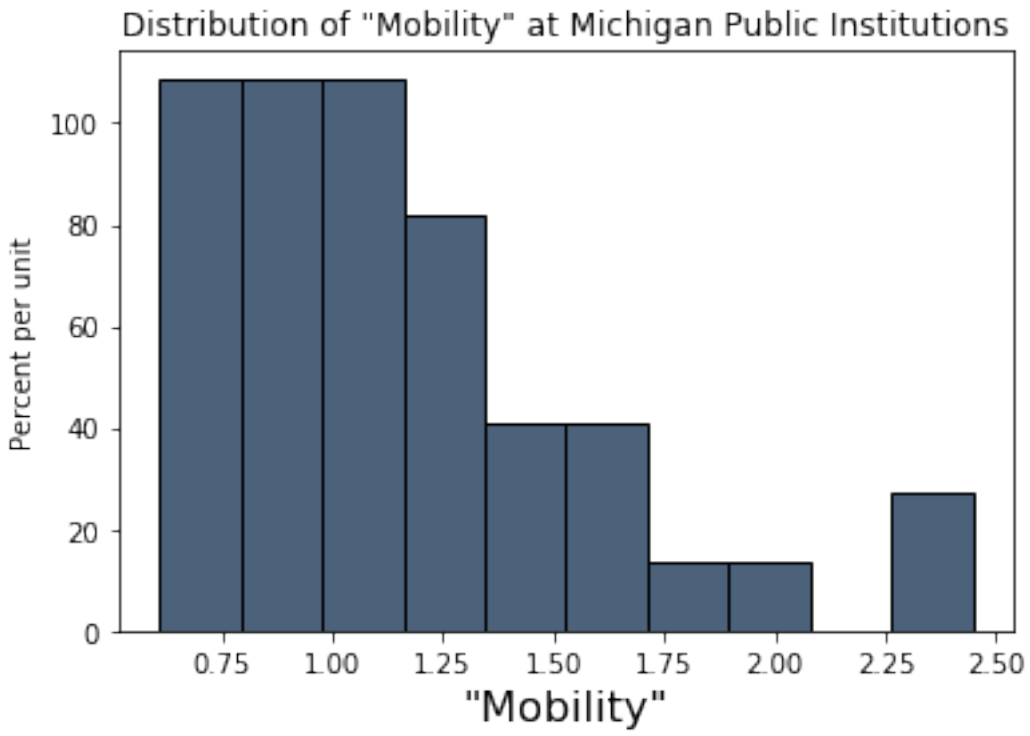
1.4.2 Mobility Rates

Finally, let's investigate `mobility` rates. Recall that `mobility` measures the percentage of students enrolled that are both 'low income' and later have earnings in the top quintile (top 20%) of the student income distribution.

EXERCISE 1.5: Plot a histogram of `mobility` across institutions. Follow the example above on `access`

[13]:
```
#create histogram of mobility column
mi_pub_mobility.hist("mobility")

#the code below will label the axes and title of your histogram
plt.title('Distribution of "Mobility" at Michigan Public Institutions')
plt.xlabel('"Mobility"')
```

[13]:
```
Text(0.5, 0, '"Mobility"')
```



You should see again points that separate themselves from the distribution. What institution(s) is that? Set `mobility_outlier` to the name of the institution.

[16]:
```
mobility_outlier = mi_pub_mobility.where("mobility", are.above(2.2))
print(mobility_outlier)
```

```
super_opeid | name                                | type    | tier             |
iclevel    | state | count    | access   | success | mobility
2326        | University Of Michigan - Dearborn | Public | Selective public |
Four-year | MI    | 700.333 | 6.59789 | 36.1298 | 2.3838
2329        | Wayne State University            | Public | Selective public |
Four-year | MI    | 1782.33 | 12.6008 | 19.4228 | 2.44744
```

EXERCISE 1.6: Let's compute the same summary statistics for `mobility`: the mean, standard deviation, and the same percentile values.

[17]:
```
mobility_mean = np.mean(mi_pub_mobility.column("mobility"))
mobility_std = np.std(mi_pub_mobility.column("mobility"))
mobility_percentiles = np.percentile(mi_pub_mobility.column("mobility"), [25,␣
↪50, 75])

print('mean:', mobility_mean)
print('standard deviation:', mobility_std)
print('percentiles:', mobility_percentiles)
```

```
mean: 1.16829738775
standard deviation: 0.4392575677417631
percentiles: [0.84137765 1.0329767  1.34960568]
```

For the sake of comparison, here are `access`, `success`, and `mobility` for Michigan Tech.

[19]:
```
mi_pub_mobility.where('name', are.equal_to('Michigan Technological␣
↪University')).select(['name', 'access', 'success', 'mobility'])
```

[19]:
```
name                              | access  | success | mobility
Michigan Technological University | 3.74191 | 46.6819 | 1.7468
```

1.5 2. How Does Access Vary Over Time?

In this section we will study how low income access to Michigan public institutions has changed over time. Over the last 40 years, public spending on higher education in Michigan has decreased dramatically.

EXERCISE 2.1: We will begin by loading a new dataset, which is described in more detail below.

[22]:
```
mobility_panel = Table.read_table('data/mrc_q2.csv')

#restrict to California public and private (non-profit) institutions
mi_mobility_panel = mobility_panel.where('state', are.equal_to('MI')).
↪where('type', are.contained_in(make_array('Public', 'Private Non-profit')))

#drop missing values
mi_mobility_panel = mi_mobility_panel.where('access', are.above(0))

mi_mobility_panel.show(5)
```

```
<IPython.core.display.HTML object>
```

These data are **longitudinal data** (also known as **panel data**), which means they follow the same object over time with repeated observations. In this case, the data follow institutions over time.

These particular longitudinal data are organized by **cohort**. In general, a cohort is a group of individuals that share some common factor, of a year of birth or year of matriculation. In this case,

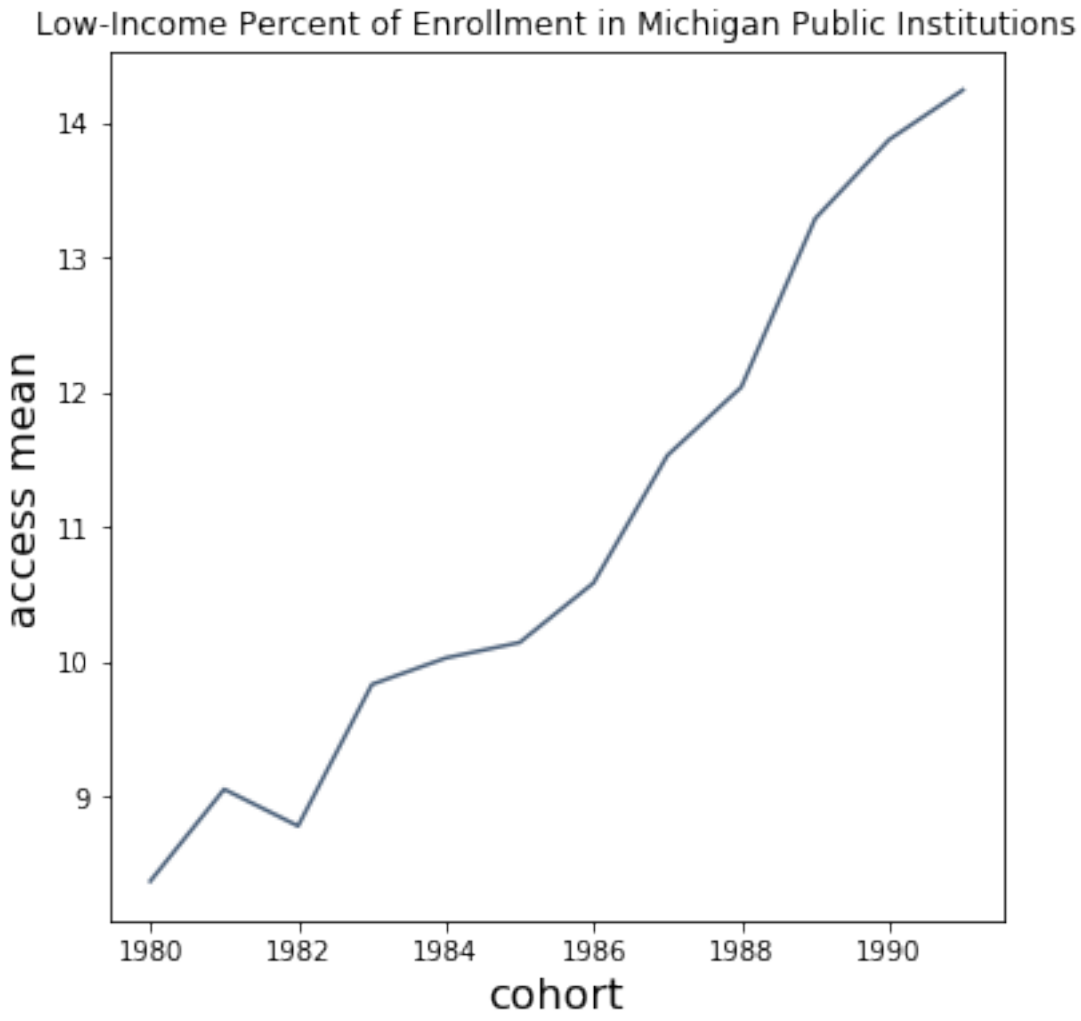
cohorts are defined by the student's year of birth. For each institution, there is now a separate row of data for students born in each year, ranging from 1980 to 1991.

The column `count` records the number of students from each cohort that were included in the underlying data.

EXERCISE 2.2: Let's measure `access` over time (by cohort), averaging across all public institutions.

[23]:
```
#plot `access` by cohort
#note: you will cover the group function later this week in Data 8. The code␣
 ↪below collapses the data into cohort-level averages.
mi_mobility_panel_public = mi_mobility_panel.where('type', are.
 ↪equal_to('Public')).group('cohort', collect = np.mean)

#When plotting we must first select the columns we want to plot
mi_mobility_panel_public.select(make_array('cohort','access mean')).
 ↪plot(column_for_xticks='cohort')
plt.title('Low-Income Percent of Enrollment in Michigan Public Institutions')
```

[23]:
```
Text(0.5, 1.0, 'Low-Income Percent of Enrollment in Michigan Public
Institutions')
```



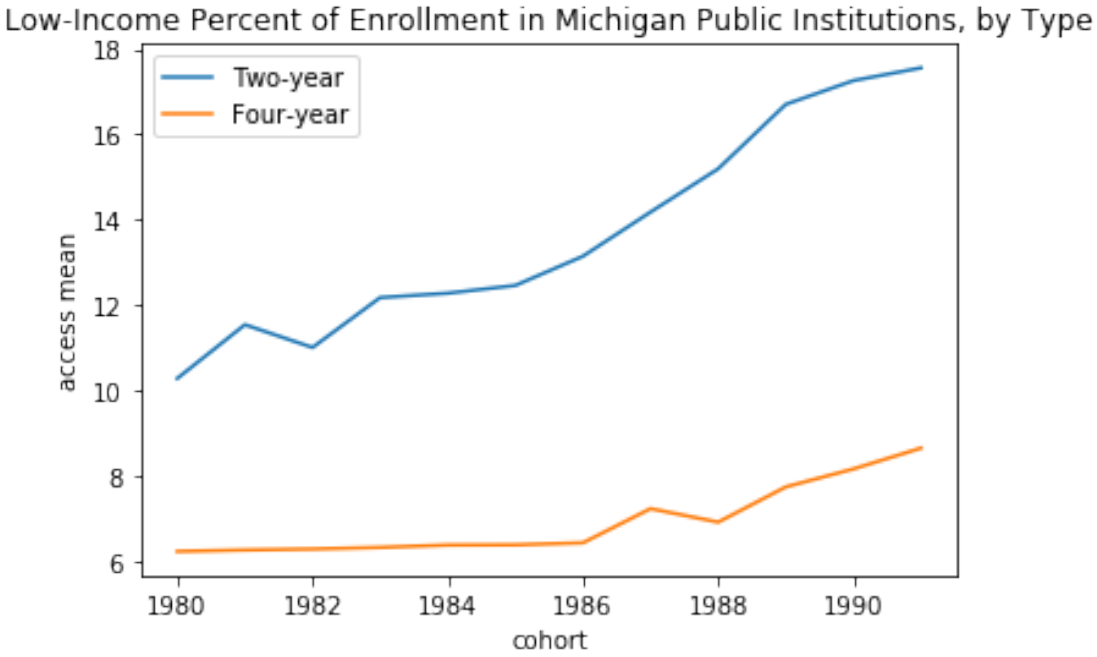
EXERCISE 2.3: Now, let's separate this figure by institution type.

(Note: to overlay plots, we had to go outside the datascience package. Here, I used matplotlib, what creates the Table.plot charts. If you're curious, you can learn more here.)

[25]:
```
mi_mobility_two_year = mi_mobility_panel.where('type', are.equal_to('Public')
                                               ).where('iclevel', are.
 ↪equal_to('Two-year')).group('cohort', collect = np.mean)

mi_mobility_four_year = mi_mobility_panel.where('type', are.equal_to('Public')
                                                ).where('iclevel', are.
 ↪equal_to('Four-year')).group('cohort', collect = np.mean)

plt.plot(mi_mobility_two_year.column('cohort'), mi_mobility_two_year.
 ↪column('access mean'), label = 'Two-year')
```

```
plt.plot(mi_mobility_four_year.column('cohort'), mi_mobility_four_year.
 ↪column('access mean'), label = 'Four-year')
plt.title('Low-Income Percent of Enrollment in Michigan Public Institutions, by␣
 ↪Type')
plt.legend()
plt.xlabel("cohort")
plt.ylabel("access mean")
```

[25]:
```
Text(0, 0.5, 'access mean')
```



A couple of key takeaways from this figure: * the *level* of `access` is significantly higher at two-year colleges. * both two-year and four-year colleges are seeing increases of `access`

EXERCISE 2.4: Finally, for comparison's sake, let's check how low-income access is evolving at private non-profit 4-year institutions in Michigan.

Perhaps there is some substitution to these institutions, some of which have increased their financial aid offerings over time.

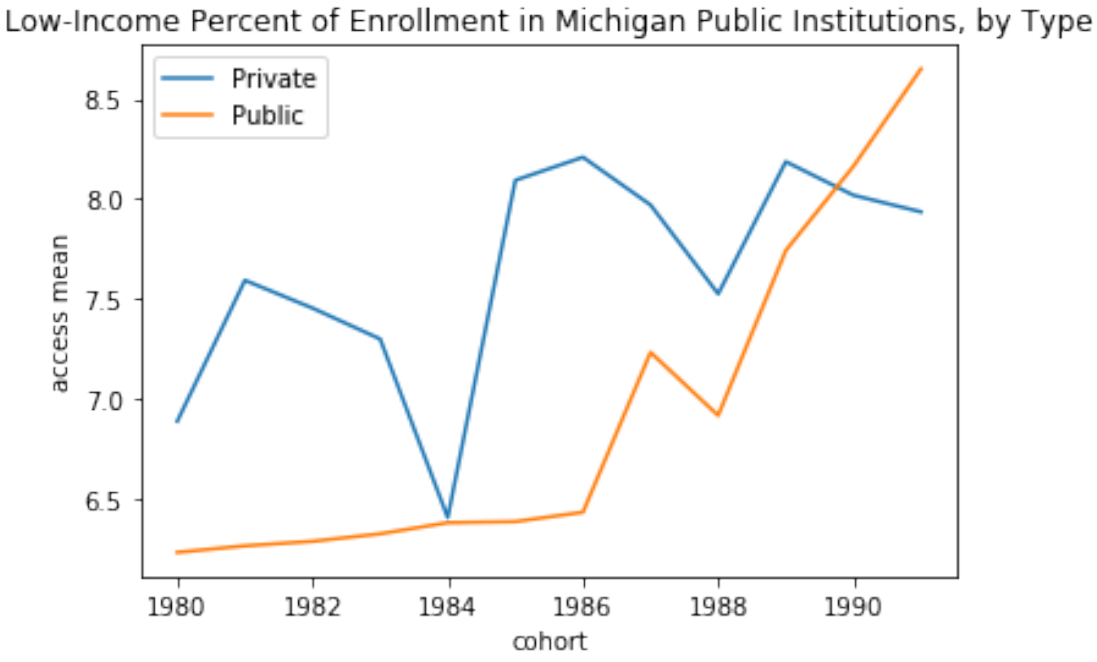
For this exercise you will need to use the following columns:

- **`iclevel`:** indicates whether an institution is a 4-year, 2-year, or less than 2-year college.
- **`type`:** indicates whether an institution is a Public, Private Non-profit, or Private For-profit institution.

[31]:
```
#use similar code as above, except replace public two-year institutions with␣
 ↪private four-year institutions
```

```
mi_mobility_private = mi_mobility_panel.where('type', are.equal_to('Private␣
↪Non-profit')
                                                ).where('iclevel', are.
↪equal_to('Four-year')).group('cohort', collect = np.mean)


mi_mobility_public = mi_mobility_panel.where('type', are.equal_to('Public')
                                                 ).where('iclevel', are.
↪equal_to('Four-year')).group('cohort', collect = np.mean)

plt.plot(mi_mobility_private.column('cohort'), mi_mobility_private.
↪column('access mean'), label = 'Private')
plt.plot(mi_mobility_public.column('cohort'), mi_mobility_public.column('access␣
↪mean'), label = 'Public')
plt.title('Low-Income Percent of Enrollment in Michigan Public Institutions, by␣
↪Type')
plt.legend()
plt.xlabel("cohort")
plt.ylabel("access mean")
```

[31]: Text(0, 0.5, 'access mean')



1.6 3. Creating a College Report Card

The main output of the Chetty et al. (2017) project is a Mobility Report Card for each school included in their data. The Report Card shows the composition of an institution's students by parental income quintile, and success rates by parental income quintile. Report Cards for each institution can be found here.

Above, you can see the Report Card for Michigan Tech. The figure includes a bar chart for the distribution of students by parental income quintile, and a line plot the show success rates by parental income quintile. The figure is effective–it presents a lot of information without too much clutter.

In this section we will create a function that generates a Report Card comparing two institutions.

EXERCISE 3.1: For this exercise, it will be easier to work with the first dataset in a different format. Again, we will restrict to public Michigan colleges and universities.

[33]:
```
#read in data
mobility_long = Table.read_table("data/mrc_q3.csv")

#restrict to CA public institutions again
mi_pub_mobility_long = mobility_long.where('type', are.equal_to('Public')).
 ↪where('state', are.equal_to('MI'))

mi_pub_mobility_long.show(5)
```

```
<IPython.core.display.HTML object>
```

Notice that now there are *5* observations per institution. While each row represented an institution in the first table, in this table each row represents an institution by parental income quintile *pair*. The latter is denoted by the column `parq`.

(What we have done is transformed the data from *wide* to *long* format. The details of this are beyond the scope of this lab.)

There are two other columns that require explanation:

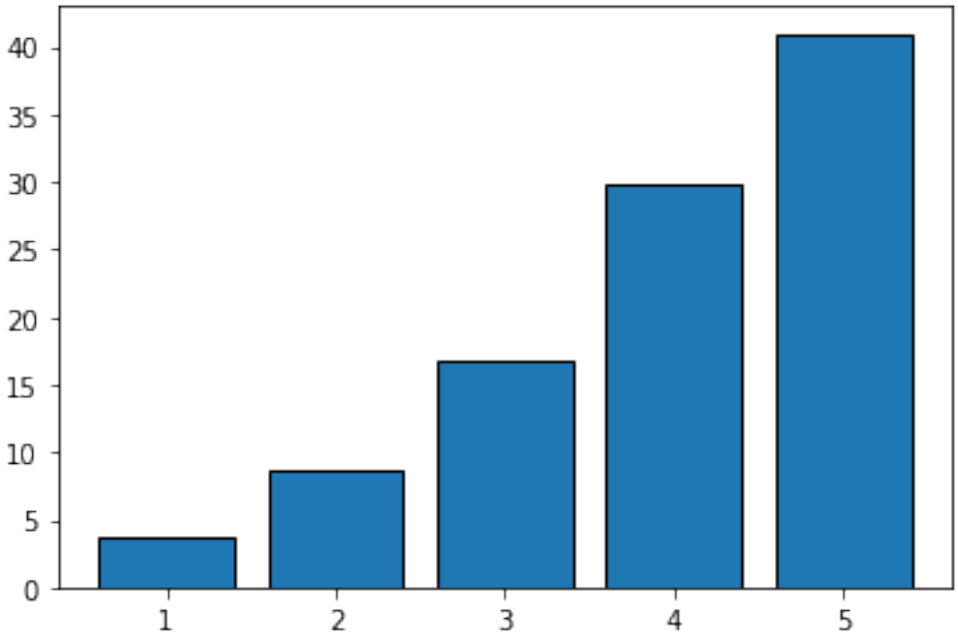
- **`percent`**: this is the percent of students at the institution with parental income in the quintile indicated by `parq`. Across the 5 rows for each institution, these values will sum to 100.
- **`success_by_q`**: this is the 'success rate' for students from a particular institution and parental income quintile. In other words, it is the percentage of students that reach the top quintile of the children's income distribution.

EXERCISE 3.2: First, create the bar chart portion of the Report Card for Michigan Tech.

[34]:
```
#create table with just Berkeley data
mtu_mobility_long = mi_pub_mobility_long.where('name', are.equal_to('Michigan␣
 ↪Technological University'))

#create bar chart
```

```
plt.bar(mtu_mobility_long.column('parq'), mtu_mobility_long.column('percent'),␣
 ↪label = 'Michigan Technological University')
```

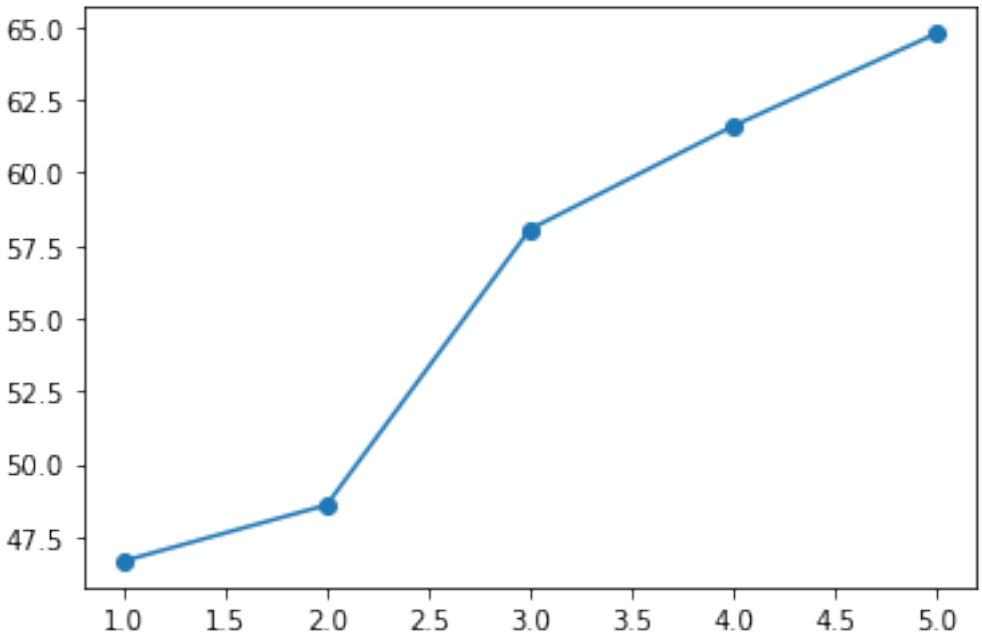
[34]: <BarContainer object of 5 artists>



EXERCISE 3.3: Next, create the line plot portion. Specify which columns labels belong on the x and y axes, take those columns from the relevant table, and use .plot from `matplotlib` to create the scatter plot.

[35]:
```
#create line plot
plt.plot(mtu_mobility_long.column('parq'), mtu_mobility_long.
 ↪column('success_by_q'), marker='o')
```

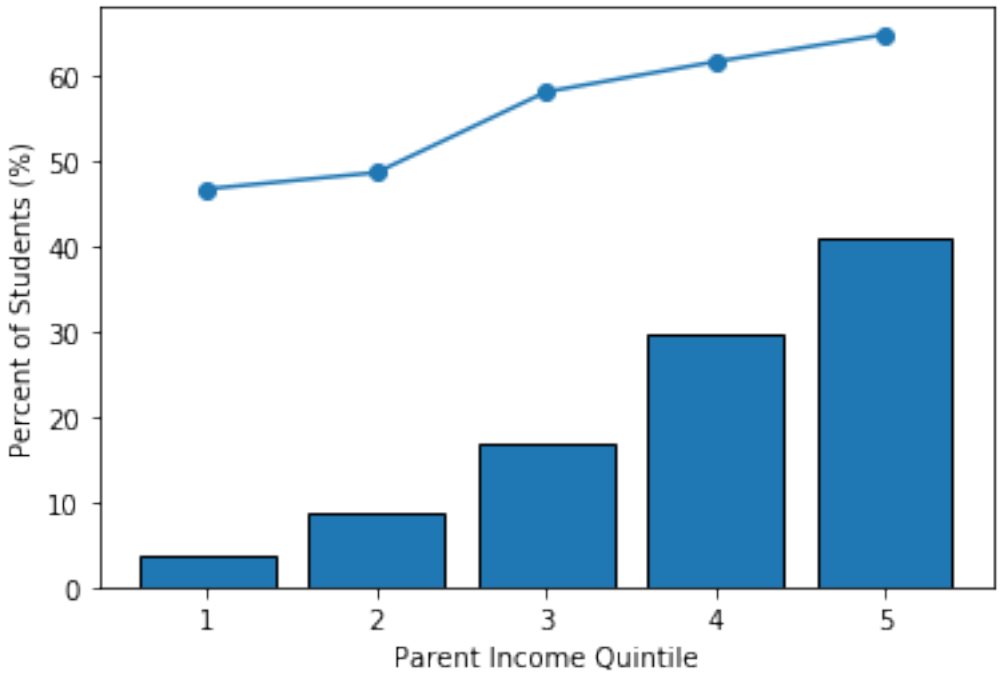
[35]: [<matplotlib.lines.Line2D at 0x1a26d4aa10>]



Notice the difference in vertical scales for the two figures.

EXERCISE 3.4: Let's put the last two pieces together in one figure as in the offical Report Cards.

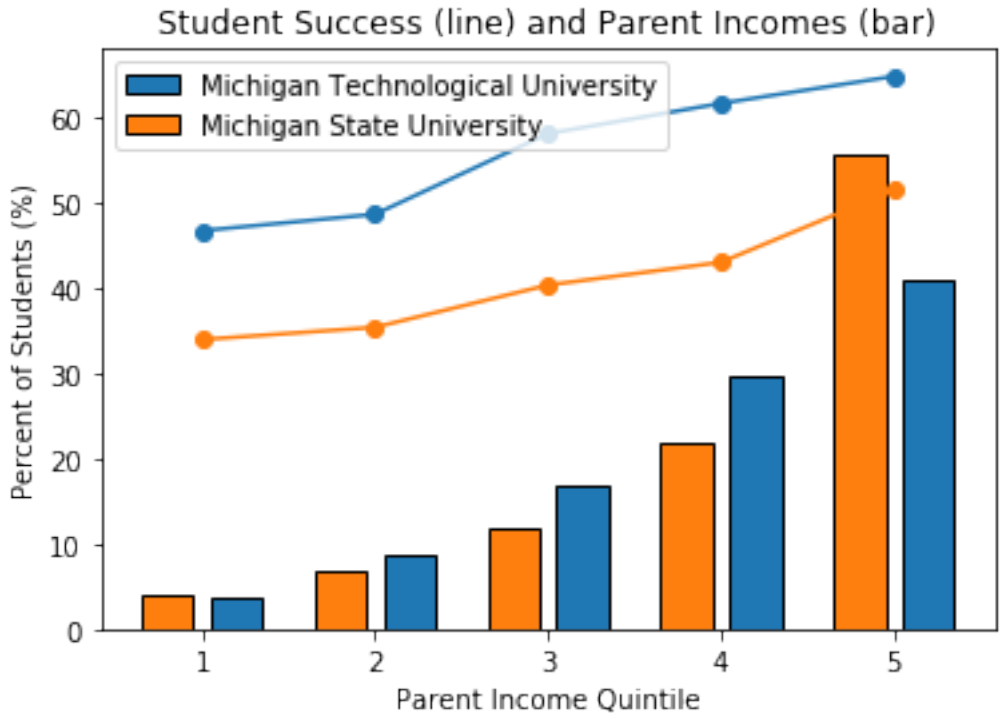
[39]:
```
#copy and paste your code from previous two cells
plt.bar(mtu_mobility_long.column('parq'), mtu_mobility_long.column('percent'),␣
 ↪label = 'Michigan Technological University')
plt.plot(mtu_mobility_long.column('parq'), mtu_mobility_long.
 ↪column('success_by_q'), marker='o')

#and include this last line
plt.xlabel('Parent Income Quintile')
plt.ylabel('Percent of Students (%)')
plt.show()
```



We're almost there! We just need to combine the data from two institutions in one plot. The code below generates a Report Card that compares Michigan Tech and Michigan State.

[41]:
```
#create report card that compares two institutions
bar_width = 0.3  # default: 0.8

school1 = mi_pub_mobility_long.where('name', are.equal_to('Michigan␣
↪Technological University'))
school2 = mi_pub_mobility_long.where('name', are.equal_to('Michigan State␣
↪University'))

#create the bar charts
plt.bar(school1.column('parq') + bar_width/2 + .05, school1.column('percent'),␣
↪bar_width, label = 'Michigan Technological University')
plt.bar(school2.column('parq') - bar_width/2 - .05, school2.column('percent'),␣
↪bar_width, label = 'Michigan State University')

#create the line plots
plt.plot(school1.column('parq'), school1.column('success_by_q'), marker='o')
plt.plot(school2.column('parq'), school2.column('success_by_q'), marker='o')

plt.legend()
plt.xlabel('Parent Income Quintile')
plt.ylabel('Percent of Students (%)')
plt.title('Student Success (line) and Parent Incomes (bar)')
```

```
plt.show()
```



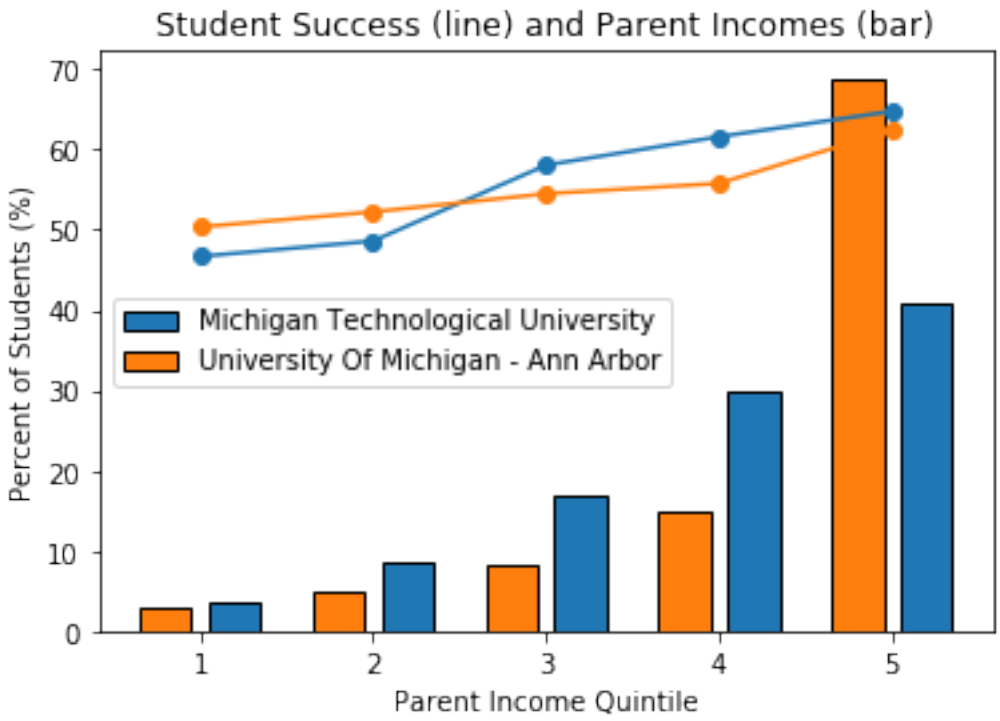
EXERCISE 3.5: Create a function that takes two institution names as arguments and returns a Report Card that compares the two.

[42]:
```
#turn into function
#hint: you should first copy the code from the cell above and then make some␣
 ↪minor changes so that
#'Michigan Tech' and 'Michigan State' are replaced by the names for the function␣
 ↪arguments.

def report_card(a, b):

    school1 = mi_pub_mobility_long.where('name', are.equal_to(a))
    school2 = mi_pub_mobility_long.where('name', are.equal_to(b))

    #create the bar charts
    plt.bar(school1.column('parq') + bar_width/2 + .05, school1.
 ↪column('percent'), bar_width, label = a)
    plt.bar(school2.column('parq') - bar_width/2 - .05, school2.
 ↪column('percent'), bar_width, label = b)

    #create the line plots
```

```
    plt.plot(school1.column('parq'), school1.column('success_by_q'), marker='o')
    plt.plot(school2.column('parq'), school2.column('success_by_q'), marker='o')

    plt.legend()
    plt.xlabel('Parent Income Quintile')
    plt.ylabel('Percent of Students (%)')
    plt.title('Student Success (line) and Parent Incomes (bar)')

    return plt.show()
```

EXERCISE 3.6: Generate a report card using two institutions of your choosing. Describe the comparison.

[48]:
```
report_card('Michigan Technological University', 'University Of Michigan - Ann␣
↪Arbor')
```



If you're interested in these data, you can play around with this data exploration tool put together by the New York Times.

Congratulations, you've finished the exercise.

[ ]:
```

```