U.S. Universities - Regional Performance

By Sarina Noone

Using datasets from the Integrated Postsecondary Education Data System, or IPEDS, I will explore the range of American 4-year universities (public, private or for-profit), their distribution across the country and look at one basic indicator of student outcomes: years spent in attaining a bachelor's degree.

Hypothesis: Americans tend to hold biases towards elite, east coast universities because of their historical excellence and the reputation of the Ivy League. While based on density and presence of major employers in coastal cities probably leads to a higher number of institutions of higher education, this may not be linked to school performance.

```
import pandas as pd
```

```
#import first dataset with all universit
allunis = pd.read_csv('/home/jovyan/pyth
```

```
#total number of universities in dataset
allunis['UNITID'].count()
```

```
6440
```

```
allunis.head()
```

	UNITID	INSTNM	IALIAS	
0	100654	Alabama A & M University	AAMU	M
1	100663	University of Alabama at Birmingham		Admini Blc
2	100690	Amridge University	Southern Christian University Regions University	120(
3	100706	University of Alabama in Huntsville	UAH University of Alabama Huntsville	301 Sp
4	100724	Alabama State University		

5 rows × 73 columns

```
allunis['SECTOR'].dtypes
```

```
dtype('int64')
```

For the sake of this exploration, we'll limit our scope to "traditional" four-year college experiences. To do this, I have identified the appropriate sector codes in the IPEDS data set to filter out all community colleges, trade schools, certificate programs and graduate/professional degree granting insitutions.

```
sectorcodes = [1,2,3]
fouryearunis = allunis[allunis.SECTOR.is
```

```
#total number of four year universities
```

```
fouryearunis['UNITID'].count()
```

```
2846
```

```
fouryearunis_cleaned = fouryearunis[["UN
fouryearunis_cleaned.head()
```

	UNITID	INSTNM	CITY	STA
0	100654	Alabama A & M University	Normal	
1	100663	University of Alabama at Birmingham	Birmingham	
2	100690	Amridge University	Montgomery	
3	100706	University of Alabama in Huntsville	Huntsville	
4	100724	Alabama State University	Montgomery	

Print to PDF

```
#to contextualize sector number, add a c

def label_sectortype(row):
    if row['SECTOR']==1:
        return "Public 4-Year"
    elif row['SECTOR']==2:
        return "Private 4-Year"
    elif row['SECTOR']==3:
        return "For-Profit 4-Year"
    else:
        return 'Invalid Sector'
```

```
#applying that label to the dataset
fouryearunis_cleaned['sectortype'] = fou
fouryearunis_cleaned.head()
```

```
/tmp/ipykernel_1558/675621173.py:2: Sett
A value is trying to be set on a copy of
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: ht
  fouryearunis_cleaned['sectortype'] = f
```

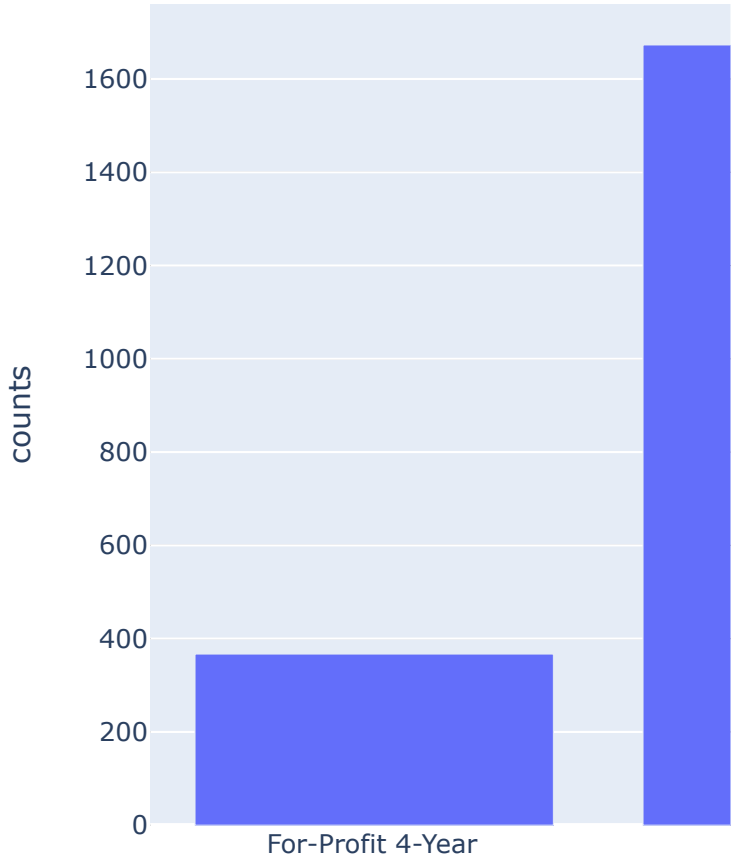
	UNITID	INSTNM	CITY	STA
0	100654	Alabama A & M University	Normal	
1	100663	University of Alabama at Birmingham	Birmingham	
2	100690	Amridge University	Montgomery	
3	100706	University of Alabama in Huntsville	Huntsville	
4	100724	Alabama State University	Montgomery	

```
#to get a sense of the type of universit

fouryear_bytype = fouryearunis_cleaned.g
print(fouryear_bytype)
```

```
          sectortype  counts
0  For-Profit 4-Year     367
1     Private 4-Year    1673
2      Public 4-Year     806
```

```
import plotly.express as px
```

```
fig = px.bar(fouryear_bytype, x='sector
fig.show()
```



It's not surprising to see that the number of Private Four-Year universities far exceeds the number of for-profit and public universities combined.

Next up, we'll see how these institutions are spread across the country. To do this, I will add context to the IPEDS dataset's OBEREG data flag to indicate which part of the country is

represented.

```
#to contextualize regions in column OBEF

def label_region(row):
    if row['OBEREG']==1:
        return "New England"
    elif row['OBEREG']==2:
        return "Mid Atlantic"
    elif row['OBEREG']==3:
        return "Great Lakes"
    elif row['OBEREG']==4:
        return "Plains"
    elif row['OBEREG']==5:
        return "Southeast"
    elif row['OBEREG']==6:
        return "Southwest"
    elif row['OBEREG']==7:
        return "Rocky Mountains"
    elif row['OBEREG']==8:
        return "Far West"
    elif row['OBEREG']==9:
        return "US Territories"
    else:
        return 'N/A or Other'
```

```
#applying that label to the dataset
fouryearunis_cleaned['region'] = fourye
fouryearunis_cleaned.head()
```

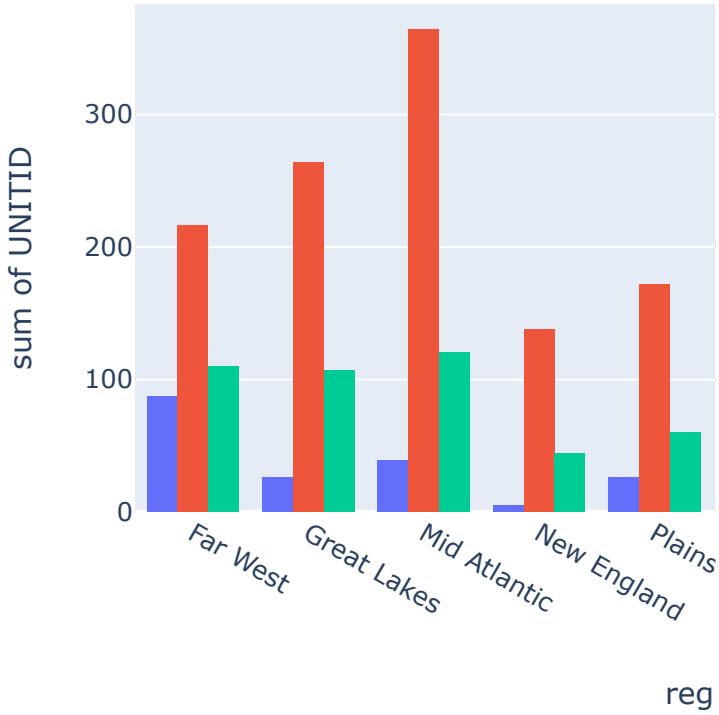
```
/tmp/ipykernel_1558/1056732486.py:2: Set


A value is trying to be set on a copy of
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: ht
```

	UNITID	INSTNM	CITY	STA
0	100654	Alabama A & M University	Normal	
1	100663	University of Alabama at Birmingham	Birmingham	
2	100690	Amridge University	Montgomery	
3	100706	University of Alabama in Huntsville	Huntsville	
4	100724	Alabama State University	Montgomery	

```
fouryearunis_fullset = fouryearunis_clea
print(fouryearunis_fullset)
```

```
            region         sectortype
0          Far West  For-Profit 4-Year
1          Far West     Private 4-Year
2          Far West      Public 4-Year
3       Great Lakes  For-Profit 4-Year
4       Great Lakes     Private 4-Year
5       Great Lakes      Public 4-Year
6      Mid Atlantic  For-Profit 4-Year
7      Mid Atlantic     Private 4-Year
8      Mid Atlantic      Public 4-Year
9      N/A or Other      Public 4-Year
10      New England  For-Profit 4-Year
11      New England     Private 4-Year
12      New England      Public 4-Year
13           Plains  For-Profit 4-Year
14           Plains     Private 4-Year
15           Plains      Public 4-Year
16  Rocky Mountains  For-Profit 4-Year
```

```
import plotly.express as px

df = fouryearunis_fullset
fig = px.histogram(df, x="region", y="UN
             color='sectortype', barmode
             height=400)
fig.show()
```



I was honestly surprised to find that the Southeast has nearly the same number of private 4-year universities as the Mid Atlantic region, and almost twice as many public universities. The Great Lakes region surprised me at first, but then I remembered that includes all Chicago/IL schools, Wisconsin, Michigan, etc. To represent these findings a little more simply, I'll run some sums by region next.

```
total_regional =  fouryearunis_cleaned.g
sorted_regional = total_regional.sort_va
print(sorted_regional)
```

```
region
Southeast          633
Mid Atlantic       523
Far West           413
Great Lakes        397
Plains             258
Southwest          241
New England        187
Rocky Mountains    104
US Territories      83
N/A or Other         7
Name: UNITID, dtype: int64
```

Now that we have established a sense of where American universities are located and the range of four year institutions, we'll look broadly at the number of students they serve and what kind of outcomes they generally have. To do so, I'll import a new data set from IPEDS that focuses on student outcomes.

```
#import second dataset with university o
uni_outcomes = pd.read_csv('/home/jovyan
```

```
uni_outcomes.head()
```

	UNITID	OMCHRT	XOMRCHRT	OMRC
0	100654	10	R	
1	100654	11	R	
2	100654	12	R	
3	100654	20	R	
4	100654	21	R	

5 rows × 54 columns

Reviewing the IPEDS variable list and descriptions, I'm most interested in looking at the total number of students an institution serves. The relevant variable is OMCHRT, where a value of 50 = total entering students; 51 = Total entering Pell Grant recipients; and 52 = total entering non-Pell Grant recipients. While it would definitely be interesting to explore different outcomes for students based on their financial aid status, for the sake of this assignment, I'll use the total number of students entering in a cohort (OMCHRT = 50). The value in OMACHRT is the number of students who fit the descriptor in OMCHRT.

To assess outcomes, we'll look at data in columns OMBACH4, OMBACH6 and OMNOAWD, which represent, respectively, the number of students who earned a bachelor's degree within four years, within six years or who at 8 years have not earned a degree yet.

```
PellCodes = [50]
uni_students = uni_outcomes[uni_outcomes
uni_students.head()
```

	UNITID	OMCHRT	XOMRCHRT	OMF
12	100654	50	R	
27	100663	50	R	
41	100690	50	R	
56	100706	50	R	
71	100724	50	R	

5 rows × 54 columns

```
#to contextualize student population se

def label_studentdetails(row):
    if row['OMCHRT']==50:
        return "Total Students"
    else:
        return 'Data Unavailable'
```

```
#applying that label to the dataset
uni_students['studentdetails'] = uni_stu
uni_students.head()
```

```
/tmp/ipykernel_1558/3244937734.py:2: Set


A value is trying to be set on a copy of
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: ht
```

	UNITID	OMCHRT	XOMRCHRT	OMF
12	100654	50	R	
27	100663	50	R	
41	100690	50	R	
56	100706	50	R	
71	100724	50	R	

5 rows × 55 columns

```
uni_students_cleaned = uni_students [["U
uni_students_cleaned.head()
```

	UNITID	OMACHRT	OMBACH4	OME
12	100654	1272	118.0	
27	100663	3521	1366.0	
41	100690	147	43.0	
56	100706	1573	553.0	
71	100724	1873	239.0	

For each university and each subset of students, we'll calculate the number of students that are "well served" as those who earn their degree within the four years; we'll calculate those "poorly served" as those who do not have a degree after eight years. Each of these will be represented as a percentage of the total subpopulation.

```
uni_students_cleaned.dtypes
```

```
UNITID              int64
OMACHRT             int64
OMBACH4           float64
OMBACH6           float64
OMNOAWD             int64
studentdetails     object
dtype: object
```

```
uni_students_cleaned['pct_well_served']=
uni_students_cleaned['pct_poorly_served
uni_students_cleaned
```

```
/tmp/ipykernel_1558/3501207542.py:1: Set


A value is trying to be set on a copy of
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: ht

/tmp/ipykernel_1558/3501207542.py:2: Set


A value is trying to be set on a copy of
Try using .loc[row_indexer,col_indexer]

See the caveats in the documentation: ht
```

	UNITID	OMACHRT	OMBACH4
12	100654	1272	118.0
27	100663	3521	1366.0
41	100690	147	43.0
56	100706	1573	553.0
71	100724	1873	239.0
...	...	...	...
48192	495031	2	0.0
48196	495147	2	NaN
48200	495183	2	NaN
48206	495280	13	0.0
48220	495767	20061	9848.0

3694 rows × 8 columns

To attempt to put this into context with the data on institution type and region, I will merge these datasets using their unique UNITIDs.

```
finaldata = pd.merge(
    left=fouryearunis_cleaned,
    right=uni_students_cleaned,
    how="left",
    on=None,
    left_on='UNITID',
    right_on='UNITID',
    left_index=False,
    right_index=False,
    sort=True,
    suffixes=("_x", "_y"),
    copy=True,
    indicator=False,
    validate=None,
)
finaldata.head()
```

	UNITID	INSTNM	CITY	STA
0	100654	Alabama A & M University	Normal	
1	100663	University of Alabama at Birmingham	Birmingham	
2	100690	Amridge University	Montgomery	
3	100706	University of Alabama in Huntsville	Huntsville	
4	100724	Alabama State University	Montgomery	

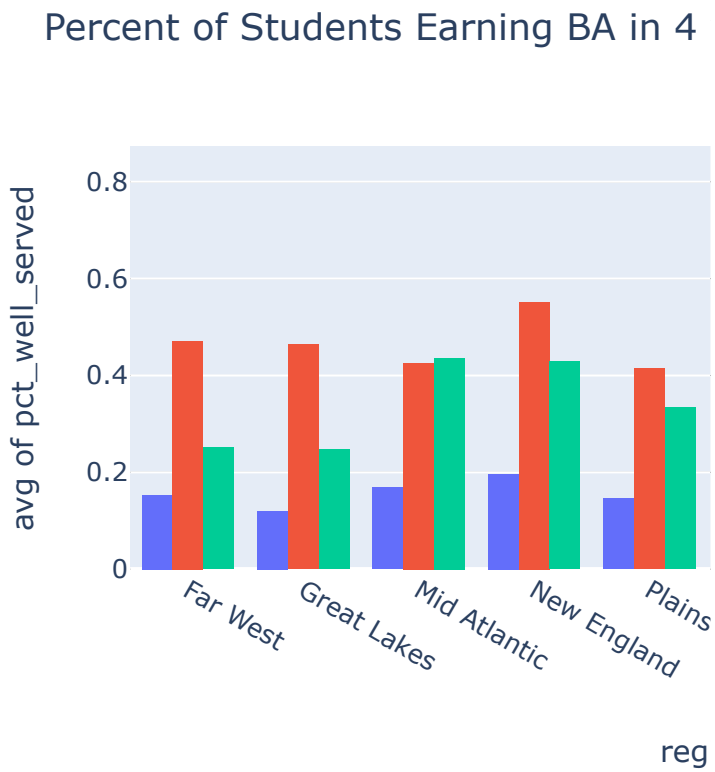
Lastly, I'll try a few visualizations to see if there are any trends in quality of institutions by type or by region.

```
finaldata_grouped = finaldata.groupby([
finaldata_grouped
```

```
/tmp/ipykernel_1558/2067835919.py:1: Fut

Indexing with multiple keys (implicitly
```

	region	sectortype	pct_well_serv
0	Far West	For-Profit 4-Year	0.1528
1	Far West	Private 4-Year	0.4710
2	Far West	Public 4-Year	0.2516
3	Great Lakes	For-Profit 4-Year	0.1202
4	Great Lakes	Private 4-Year	0.4636
5	Great Lakes	Public 4-Year	0.2468
6	Mid Atlantic	For-Profit 4-Year	0.1688
7	Mid Atlantic	Private 4-Year	0.4257
8	Mid Atlantic	Public 4-Year	0.4364
9	N/A or Other	Public 4-Year	0.8285
10	New England	For-Profit 4-Year	0.1967
11	New	Private 4-	0.5516

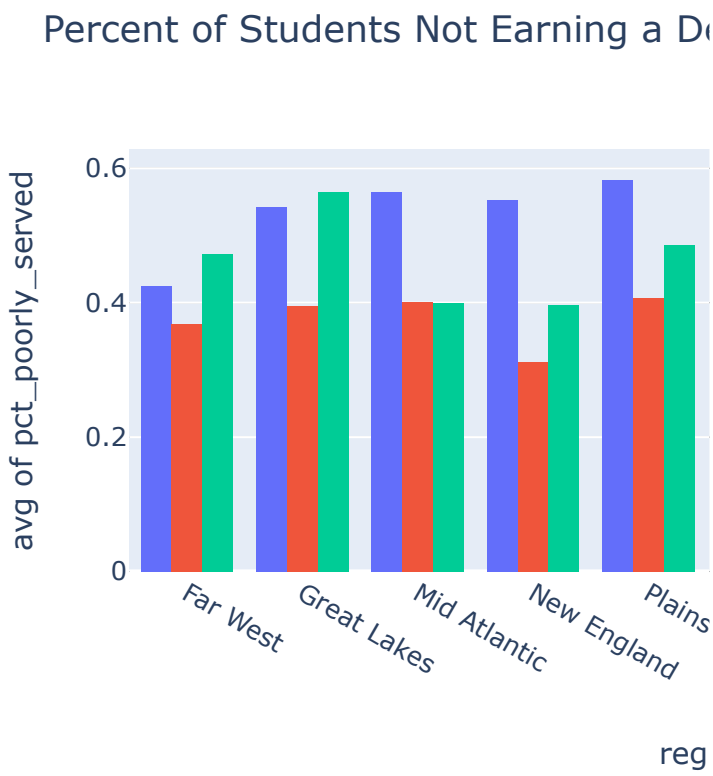
```
import plotly.express as px

df = finaldata_grouped
fig = px.histogram(df, x="region", y="pc
             color='sectortype', barmode
             height=400,
                  title="Percent of Stud
fig.show()
```



The outlier on the right under “N/A or Other region” for public 4-year institution is likely representative of US Armed Forces academies which had their own classification in IPEDS for regions. It is also worth noting that in calculating the size of the student cohort and degree attainment, students who are called into active duty, injured or deceased are excluded from the data set, which would also impact the military

colleges' performance data.

```
df = finaldata_grouped
fig = px.histogram(df, x="region", y="p
             color='sectortype', barmode
             height=400,
                  title="Percent of Stu
fig.show()
```



These visualizations show generally little variation in terms of academic outcomes for the students served. There may be a slightly higher percentage of students who are "well-served" by Mid-Atlantic private universities, but this may, of course, be conflated with the academic competitiveness of gaining admissions to certain schools and a student's past performance and aptitude.

Altogether, this study on U.S. universities and regional performance reveals the real depth of educational data and complexity in comparing school-to-school. As we know, student experiences vary based on their PK-12 educational preparation, household support and income, community resources and so many other factors that are out of the hands of the learner.

A more rigorous study could leverage IPEDS data on students' SAT scores or high school GPAs, family income, post-college job placement or more to gauge the quality or impact of the university on student outcomes. It would also be interesting to dive into the specifics of one region, for example looking within the Mid Atlantic to surface deeper variation. The four-year university dataset included 2,846 colleges which are difficult to compare.

Personal note: I was glad to have the chance to engage with IPEDS data through this assignment as I will be graduating this month and working in postsecondary education consulting. This was my first foray into using this robust data set myself, though I've read countless studies that leverage the data. I know this is a very amateur first step to exploring here, but appreciated the chance to get more familiar with it and learn how to read the descriptions for variables a little more closely.