# 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 university data incl name, state, geo region
allunis = pd.read_csv('/home/jovyan/python-public-policy/Sarina_Noone_Final/hd2020.csv')
```

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

```
6440
```

```
allunis.head()
```

| | UNITID | INSTNM | IALIAS | ADDR | CITY | STABBR | ZIP | FIPS | OBEREG | CHFNM | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | Alabama A & M University | AAMU | 4900 Meridian Street | Normal | AL | 35762 | 1 | 5 | Dr. Andrew Hugine, Jr. | ... |
| **1** | 100663 | University of Alabama at Birmingham | | Administration Bldg Suite 1070 | Birmingham | AL | 35294-0110 | 1 | 5 | Ray L. Watts | ... |
| **2** | 100690 | Amridge University | Southern Christian University Regions University | 1200 Taylor Rd | Montgomery | AL | 36117-3553 | 1 | 5 | Michael C.Turner | ... |
| **3** | 100706 | University of Alabama in Huntsville | UAH University of Alabama Huntsville | 301 Sparkman Dr | Huntsville | AL | 35899 | 1 | 5 | Darren Dawson | ... |
| **4** | 100724 | Alabama State University | | 915 S Jackson Street | Montgomery | AL | 36104-0271 | 1 | 5 | Quinton T. Ross | ... |

Back to top

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.isin(sectorcodes)]
```

```
#total number of four year universities to confirm filtered list?
```

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

```
2846
```

```
fouryearunis_cleaned = fouryearunis[["UNITID","INSTNM","CITY","STABBR","OBEREG","LONGITUD","LATITUDE","SECTOR"]]
fouryearunis_cleaned.head()
```

| | UNITID | INSTNM | CITY | STABBR | OBEREG | LONGITUD | LATITUDE | SECTOR |
|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | Alabama A & M University | Normal | AL | 5 | -86.568502 | 34.783368 | 1 |
| **1** | 100663 | University of Alabama at Birmingham | Birmingham | AL | 5 | -86.799345 | 33.505697 | 1 |
| **2** | 100690 | Amridge University | Montgomery | AL | 5 | -86.174010 | 32.362609 | 2 |
| **3** | 100706 | University of Alabama in Huntsville | Huntsville | AL | 5 | -86.640449 | 34.724557 | 1 |
| **4** | 100724 | Alabama State University | Montgomery | AL | 5 | -86.295677 | 32.364317 | 1 |

```
#to contextualize sector number, add a column explaining sector code

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'] = fouryearunis_cleaned.apply(label_sectortype, axis=1)
fouryearunis_cleaned.head()
```

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

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
  fouryearunis_cleaned['sectortype'] = fouryearunis_cleaned.apply(label_sectortype, axis=1)
```

| | UNITID | INSTNM | CITY | STABBR | OBEREG | LONGITUD | LATITUDE | SECTOR | sectortype |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | Alabama A & M University | Normal | AL | 5 | -86.568502 | 34.783368 | 1 | Public 4-Year |
| **1** | 100663 | University of Alabama at Birmingham | Birmingham | AL | 5 | -86.799345 | 33.505697 | 1 | Public 4-Year |
| **2** | 100690 | Amridge University | Montgomery | AL | 5 | -86.174010 | 32.362609 | 2 | Private 4-Year |
| **3** | 100706 | University of Alabama in Huntsville | Huntsville | AL | 5 | -86.640449 | 34.724557 | 1 | Public 4-Year |
| **4** | 100724 | Alabama State University | Montgomery | AL | 5 | -86.295677 | 32.364317 | 1 | Public 4-Year |

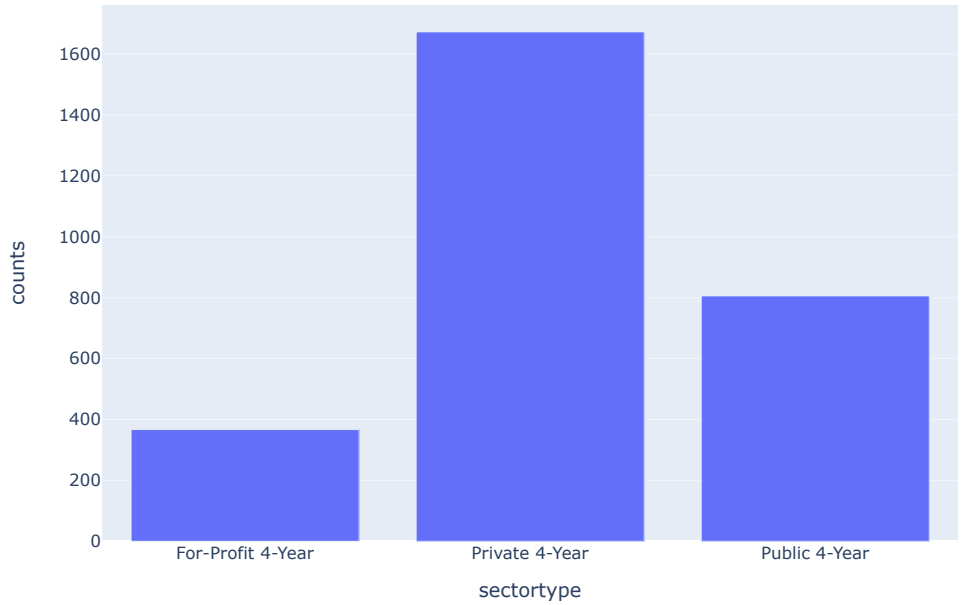
```
#to get a sense of the type of universities represented in this set

fouryear_bytype = fouryearunis_cleaned.groupby('sectortype').UNITID.size().reset_index(name='counts')
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='sectortype', y='counts')
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 OBEREG, add a column with more detail

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'] = fouryearunis_cleaned.apply(label_region, axis=1)
fouryearunis_cleaned.head()
```

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


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

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
```

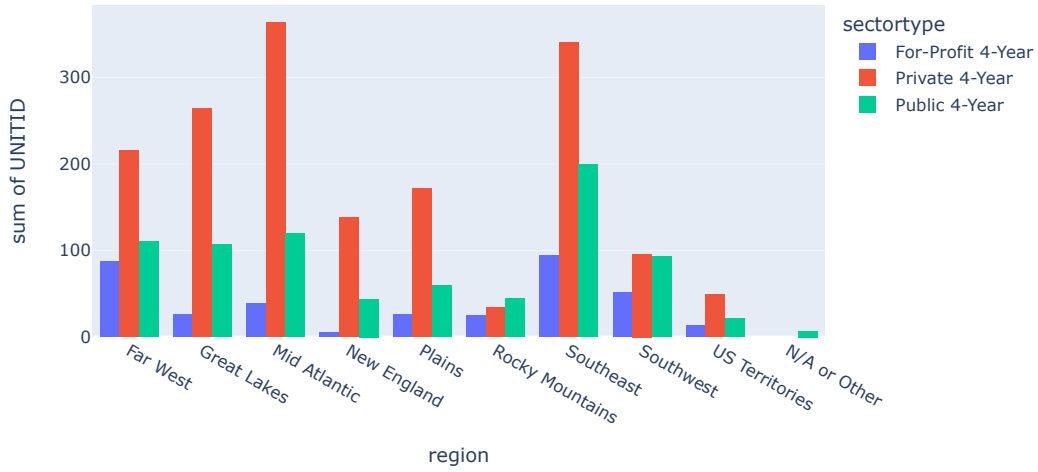
| | UNITID | INSTNM | CITY | STABBR | OBEREG | LONGITUD | LATITUDE | SECTOR | sectortype | regio |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | Alabama A & M University | Normal | AL | 5 | -86.568502 | 34.783368 | 1 | Public 4-Year | Southea |
| **1** | 100663 | University of Alabama at Birmingham | Birmingham | AL | 5 | -86.799345 | 33.505697 | 1 | Public 4-Year | Southea |
| **2** | 100690 | Amridge University | Montgomery | AL | 5 | -86.174010 | 32.362609 | 2 | Private 4-Year | Southea |
| **3** | 100706 | University of Alabama in Huntsville | Huntsville | AL | 5 | -86.640449 | 34.724557 | 1 | Public 4-Year | Southea |
| **4** | 100724 | Alabama State University | Montgomery | AL | 5 | -86.295677 | 32.364317 | 1 | Public 4-Year | Southea |

```
fouryearunis_fullset = fouryearunis_cleaned.groupby(['region','sectortype'])['UNITID'].agg('count').reset_index()
print(fouryearunis_fullset)
```

| | region | sectortype | UNITID |
|---|---|---|---|
| 0 | Far West | For-Profit 4-Year | 87 |
| 1 | Far West | Private 4-Year | 216 |
| 2 | Far West | Public 4-Year | 110 |
| 3 | Great Lakes | For-Profit 4-Year | 26 |
| 4 | Great Lakes | Private 4-Year | 264 |
| 5 | Great Lakes | Public 4-Year | 107 |
| 6 | Mid Atlantic | For-Profit 4-Year | 39 |
| 7 | Mid Atlantic | Private 4-Year | 364 |
| 8 | Mid Atlantic | Public 4-Year | 120 |
| 9 | N/A or Other | Public 4-Year | 7 |
| 10 | New England | For-Profit 4-Year | 5 |
| 11 | New England | Private 4-Year | 138 |
| 12 | New England | Public 4-Year | 44 |
| 13 | Plains | For-Profit 4-Year | 26 |
| 14 | Plains | Private 4-Year | 172 |
| 15 | Plains | Public 4-Year | 60 |

```
import plotly.express as px

df = fouryearunis_fullset
fig = px.histogram(df, x="region", y="UNITID",
             color='sectortype', barmode='group',
             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.groupby(['region'],sort=False).count()
sorted_regional = total_regional.sort_values('UNITID',ascending=False)['UNITID']
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 outcome measures such as enrollment, degree attainment
uni_outcomes = pd.read_csv('/home/jovyan/python-public-policy/Sarina_Noone_Final/om2020.csv')
```

```
uni_outcomes.head()
```

| | UNITID | OMCHRT | XOMRCHRT | OMRCHRT | XOMEXCLS | OMEXCLS | XOMACHRT | OMACHRT | XOMCERT4 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | 10 | R | 969 | R | 4 | R | 965 | R |
| **1** | 100654 | 11 | R | 788 | R | 4 | R | 784 | R |
| **2** | 100654 | 12 | R | 181 | R | 0 | R | 181 | R |
| **3** | 100654 | 20 | R | 106 | R | 1 | R | 105 | R |
| **4** | 100654 | 21 | R | 80 | R | 1 | R | 79 | 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.OMCHRT.isin(PellCodes)]
uni_students.head()
```

| | UNITID | OMCHRT | XOMRCHRT | OMRCHRT | XOMEXCLS | OMEXCLS | XOMACHRT | OMACHRT | XOMCERT4 |
|---|---|---|---|---|---|---|---|---|---|
| **12** | 100654 | 50 | R | 1277 | R | 5 | R | 1272 | R |
| **27** | 100663 | 50 | R | 3526 | R | 5 | R | 3521 | R |
| **41** | 100690 | 50 | R | 147 | R | 0 | R | 147 | R |
| **56** | 100706 | 50 | R | 1573 | R | 0 | R | 1573 | R |
| **71** | 100724 | 50 | R | 1874 | R | 1 | R | 1873 | R |

5 rows × 54 columns

```
#to contextualize student population served (OMCHRT) number, add a descriptor column

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_students.apply(label_studentdetails, axis=1)
uni_students.head()
```

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


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

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
```

| | UNITID | OMCHRT | XOMRCHRT | OMRCHRT | XOMEXCLS | OMEXCLS | XOMACHRT | OMACHRT | XOMCERT4 |
|---|---|---|---|---|---|---|---|---|---|
| **12** | 100654 | 50 | R | 1277 | R | 5 | R | 1272 | R |
| **27** | 100663 | 50 | R | 3526 | R | 5 | R | 3521 | R |
| **41** | 100690 | 50 | R | 147 | R | 0 | R | 147 | R |
| **56** | 100706 | 50 | R | 1573 | R | 0 | R | 1573 | R |
| **71** | 100724 | 50 | R | 1874 | R | 1 | R | 1873 | R |

5 rows × 55 columns

```
uni_students_cleaned = uni_students [["UNITID","OMACHRT","OMBACH4","OMBACH6","OMNOAWD","studentdetails"]]
uni_students_cleaned.head()
```

| | UNITID | OMACHRT | OMBACH4 | OMBACH6 | OMNOAWD | studentdetails |
|---|---|---|---|---|---|---|
| **12** | 100654 | 1272 | 118.0 | 349.0 | 875 | Total Students |
| **27** | 100663 | 3521 | 1366.0 | 1899.0 | 1531 | Total Students |
| **41** | 100690 | 147 | 43.0 | 49.0 | 94 | Total Students |
| **56** | 100706 | 1573 | 553.0 | 807.0 | 724 | Total Students |
| **71** | 100724 | 1873 | 239.0 | 553.0 | 1278 | Total Students |

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['OMBACH4']/uni_students_cleaned['OMACHRT'])
uni_students_cleaned['pct_poorly_served']=(uni_students_cleaned['OMNOAWD']/uni_students_cleaned['OMACHRT'])
uni_students_cleaned
```

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


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

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret

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


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

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
```

| | UNITID | OMACHRT | OMBACH4 | OMBACH6 | OMNOAWD | studentdetails | pct_well_served | pct_poorly_s |
|---|---|---|---|---|---|---|---|---|
| **12** | 100654 | 1272 | 118.0 | 349.0 | 875 | Total Students | 0.092767 | 0.6 |
| **27** | 100663 | 3521 | 1366.0 | 1899.0 | 1531 | Total Students | 0.387958 | 0.4 |
| **41** | 100690 | 147 | 43.0 | 49.0 | 94 | Total Students | 0.292517 | 0.6 |
| **56** | 100706 | 1573 | 553.0 | 807.0 | 724 | Total Students | 0.351558 | 0.4 |
| **71** | 100724 | 1873 | 239.0 | 553.0 | 1278 | Total Students | 0.127603 | 0.6 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **48192** | 495031 | 2 | 0.0 | 2.0 | 0 | Total Students | 0.000000 | 0.0 |
| **48196** | 495147 | 2 | NaN | NaN | 2 | Total Students | NaN | 1.0 |
| **48200** | 495183 | 2 | NaN | NaN | 0 | Total Students | NaN | 0.0 |
| **48206** | 495280 | 13 | 0.0 | 2.0 | 5 | Total Students | 0.000000 | 0.3 |
| **48220** | 495767 | 20061 | 9848.0 | 13258.0 | 5915 | Total Students | 0.490903 | 0.2 |

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 | STABBR | OBEREG | LONGITUD | LATITUDE | SECTOR | sectortype | regio |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 100654 | Alabama A & M University | Normal | AL | 5 | -86.568502 | 34.783368 | 1 | Public 4-Year | Southea |
| **1** | 100663 | University of Alabama at Birmingham | Birmingham | AL | 5 | -86.799345 | 33.505697 | 1 | Public 4-Year | Southea |
| **2** | 100690 | Amridge University | Montgomery | AL | 5 | -86.174010 | 32.362609 | 2 | Private 4-Year | Southea |
| **3** | 100706 | University of Alabama in Huntsville | Huntsville | AL | 5 | -86.640449 | 34.724557 | 1 | Public 4-Year | Southea |
| **4** | 100724 | Alabama State University | Montgomery | AL | 5 | -86.295677 | 32.364317 | 1 | Public 4-Year | Southea |

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(['region','sectortype'])['pct_well_served','pct_poorly_served'].agg('mean').
finaldata_grouped
```

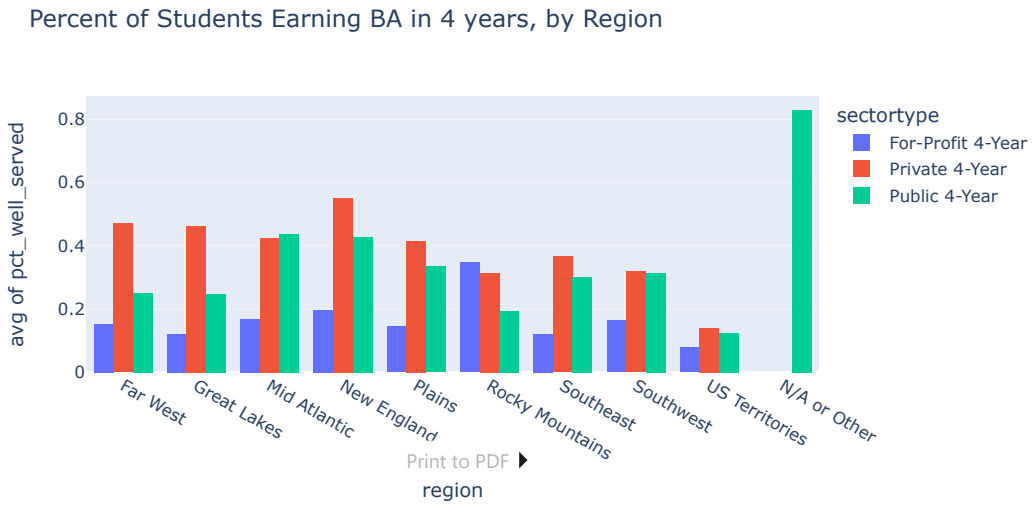
```
/tmp/ipykernel_1558/2067835919.py:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
```

| | region | sectortype | pct_well_served | pct_poorly_served |
|---|---|---|---|---|
| **0** | Far West | For-Profit 4-Year | 0.152861 | 0.424287 |
| **1** | Far West | Private 4-Year | 0.471077 | 0.368449 |
| **2** | Far West | Public 4-Year | 0.251621 | 0.471987 |
| **3** | Great Lakes | For-Profit 4-Year | 0.120212 | 0.542366 |
| **4** | Great Lakes | Private 4-Year | 0.463626 | 0.395726 |
| **5** | Great Lakes | Public 4-Year | 0.246805 | 0.563913 |
| **6** | Mid Atlantic | For-Profit 4-Year | 0.168820 | 0.564898 |
| **7** | Mid Atlantic | Private 4-Year | 0.425702 | 0.401470 |
| **8** | Mid Atlantic | Public 4-Year | 0.436490 | 0.399721 |
| **9** | N/A or Other | Public 4-Year | 0.828581 | 0.144768 |
| **10** | New England | For-Profit 4-Year | 0.196760 | 0.553050 |
| **11** | New England | Private 4-Year | 0.551633 | 0.312201 |
| **12** | New England | Public 4-Year | 0.428574 | 0.396982 |
| **13** | Plains | For-Profit 4-Year | 0.145377 | 0.583044 |
| **14** | Plains | Private 4-Year | 0.414622 | 0.407241 |
| **15** | Plains | Public 4-Year | 0.334296 | 0.485910 |

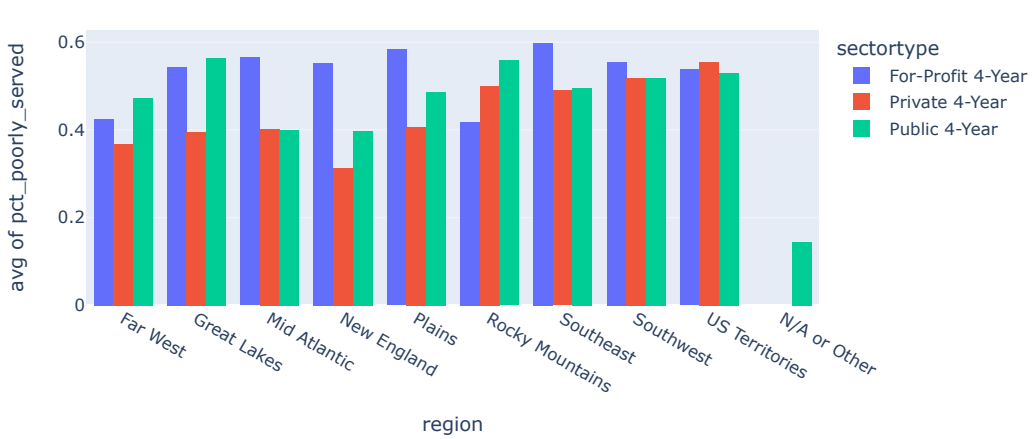
```
import plotly.express as px

df = finaldata_grouped
fig = px.histogram(df, x="region", y="pct_well_served", histfunc='avg',
             color='sectortype', barmode='group',
             height=400,
                  title="Percent of Students Earning BA in 4 years, by Region")
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="pct_poorly_served", histfunc='avg',
             color='sectortype', barmode='group',
             height=400,
                  title="Percent of Students Not Earning a Degree in 8 Years, by Region")
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.