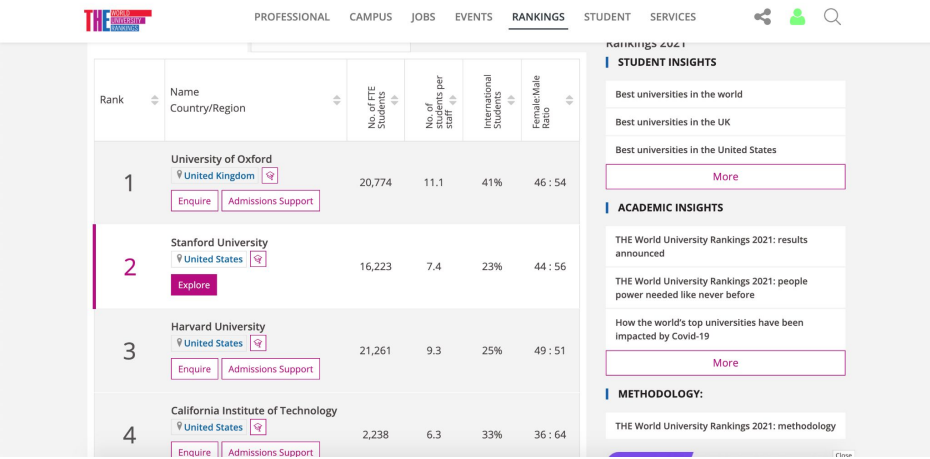
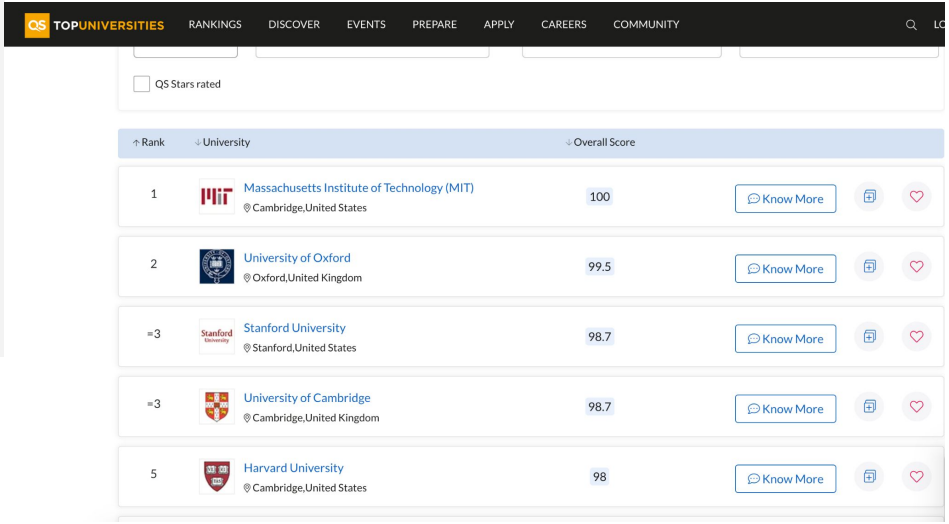
Examining the characteristics of top universities around the world, comparing to University in HK

Lo Kai Yeung

Introduction





Rank	Name Country/Region	No. of FTE Students	No. of students per staff	International Students	Female:Male Ratio
1	University of Oxford United Kingdom Enquire Admissions Support	20,774	11.1	41%	46 : 54
2	Stanford University United States Explore	16,223	7.4	23%	44 : 56
3	Harvard University United States Enquire Admissions Support	21,261	9.3	25%	49 : 51
4	California Institute of Technology United States Enquire Admissions Support	2,238	6.3	33%	36 : 64

Rankings 2021

STUDENT INSIGHTS

- Best universities in the world
- Best universities in the UK
- Best universities in the United States

More

ACADEMIC INSIGHTS

- THE World University Rankings 2021: results announced
- THE World University Rankings 2021: people power needed like never before
- How the world's top universities have been impacted by Covid-19

More

METHODOLOGY:

- THE World University Rankings 2021: methodology

Close

https://www.timeshighereducation.com/world-university-rankings/2021/world-ranking#!/page/0/length/-1/sort_by/rank/sort_order/asc/cols/stats

import all the libraries

In [ ]:
```
from bs4 import BeautifulSoup
from urllib.request import urlopen
from selenium.webdriver import Chrome
import pandas as pd
import numpy as np


import inspect

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

get the html code from the link

In [161]:

```
driver = Chrome("./chromedriver")
driver.get("https://www.timeshighereducation.com/world-university-rankings/2021/world-ranking#!/page/0/length/-1/sort_b
bs = BeautifulSoup(driver.page_source, 'html.parser')
search_result_list = bs.find_all('tr', {"role": "row"})
```

Rank	Name Country/Region	No. of FTE Students	No. of students per staff	International Students	Female:Male Ratio
1	University of Oxford United Kingdom Enquire Admissions Support	20,774	11.1	41%	46 : 54
2	Stanford University United States Explore	16,223	7.4	23%	44 : 56
3	Harvard University United States Enquire Admissions Support	21,261	9.3	25%	49 : 51
4	California Institute of Technology United States Enquire Admissions Support	2,238	6.3	33%	36 : 64

```
name_list = []
country_list = []
stats_number_students = []
stats_student_staff_ratio = []
stats_pc_intl_students = []
stats_female_male_ratio = []

for i in range(1, 251):
    name_list.append(search_result_list[i].find('a', {'class': 'ranking-institution-title'}).get_text())
    country_list.append(search_result_list[i].find('div', {'class': 'location'}).get_text())
    stats_number_students.append(search_result_list[i].find('td', {'class': 'stats stats_number_students'}).get_text())
    stats_student_staff_ratio.append(search_result_list[i].find('td', {'class': 'stats stats_student_staff_ratio'}).get
    stats_pc_intl_students.append(search_result_list[i].find('td', {'class': 'stats stats_pc_intl_students'}).get_text(
    stats_female_male_ratio.append(search_result_list[i].find('td', {'class': 'stats stats_female_male_ratio'}).get_tex


uni_df = pd.DataFrame({'name': name_list, 'Country': country_list, 'No. of students': stats_number_students, "Students
```

create lists for storing the data including, university name, country, number of students, number of students per staff, the percentage of international students and female:male ratio

create a data frame for storing the data

Missing Data

```
In [164]: uni_df = uni_df.replace(to_replace = ['n/a', 'n/' ,'/a', '/', 'n', 'a'], value = np.nan)

          uni_df.to_csv('uni_df.csv')
```

```python
uni_df['International Student Int'] = float(0)
uni_df['Female ratio'] = float(0)
uni_df['male ratio'] = float(0)
uni_df['No. of students Int'] = float(0)
uni_df['Students per staff Int'] = float(0)


for i in range(0, 250):
    if (isinstance(uni_df['International Student'][i][:-1], str)):
        uni_df.iloc[i, 6] = float(uni_df['International Student'][i][:-1].replace(' ', ''))

    if (isinstance(uni_df['Female: male ratio'][i], str)):
        uni_df.iloc[i, 7] = float(uni_df['Female: male ratio'][i][0:2].replace(' ', ''))

    if (isinstance(uni_df['Female: male ratio'][i], str)):
        uni_df.iloc[i, 8] = float(uni_df['Female: male ratio'][i][-2:].replace(' ', ''))

    if (isinstance(uni_df['No. of students'][i], str)):
        uni_df.iloc[i, 9] = float(uni_df['No. of students'][i].replace(',', ''))

    uni_df.iloc[i, 10] = float(uni_df['Students per staff'][i])

uni_df.to_csv('uni_df2.csv')
```

Examples:

20,774

41%

46 : 54

creating extra columns for storing float type values

Missing Data

```
uni_df = uni_df.replace(to_replace = 0, value = np.nan)
```

	name	Country	No. of students	Students per staff	International Student	Female: male ratio	International Student Int	Female ratio	male ratio	No. of students Int	Students per staff Int
0	University of Oxford	United Kingdom	20,774	11.1	41%	46 : 54	41.0	46.0	54.0	20774.0	11.1
1	Stanford University	United States	16,223	7.4	23%	44 : 56	23.0	44.0	56.0	16223.0	7.4
2	Harvard University	United States	21,261	9.3	25%	49 : 51	25.0	49.0	51.0	21261.0	9.3
3	California Institute of Technology	United States	2,238	6.3	33%	36 : 64	33.0	36.0	64.0	2238.0	6.3
4	Massachusetts Institute of Technology	United States	11,276	8.4	34%	39 : 61	34.0	39.0	61.0	11276.0	8.4
...	...	...	...	...	...	...	...	...	...	...	...
245	Virginia Polytechnic Institute and State Unive...	United States	34,155	18.4	14%	43 : 57	14.0	43.0	57.0	34155.0	18.4
246	University of Waterloo	Canada	32,804	22.8	21%	47 : 53	21.0	47.0	53.0	32804.0	22.8
247	Western University	Canada	29,865	22.8	18%	56 : 44	18.0	56.0	44.0	29865.0	22.8
248	University of the Witwatersrand	South Africa	27,839	25.8	7%	55 : 45	7.0	55.0	45.0	27839.0	25.8
249	University of Wollongong	Australia	18,517	30.5	29%	52 : 48	29.0	52.0	48.0	18517.0	30.5

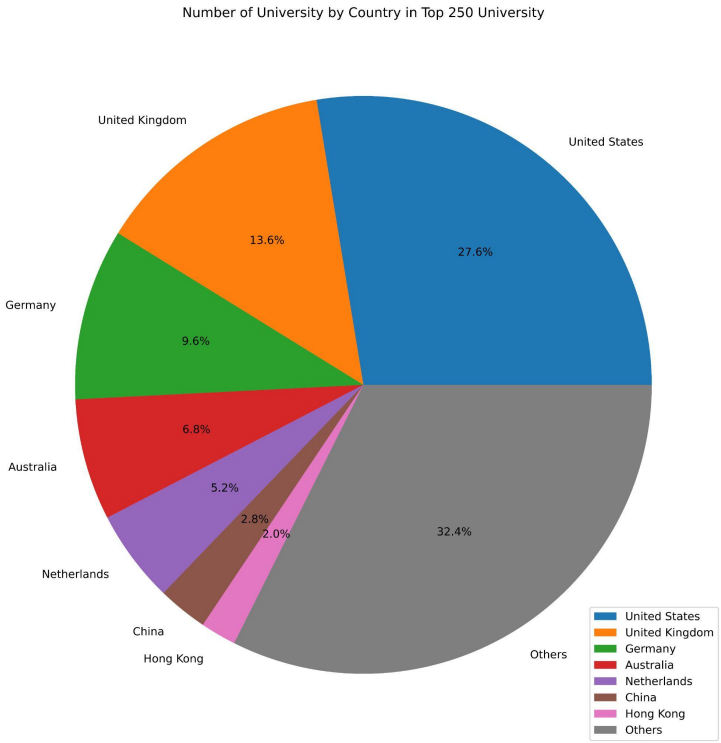
250 rows × 11 columns

Data visualization

A pie chart: showing countries distribution

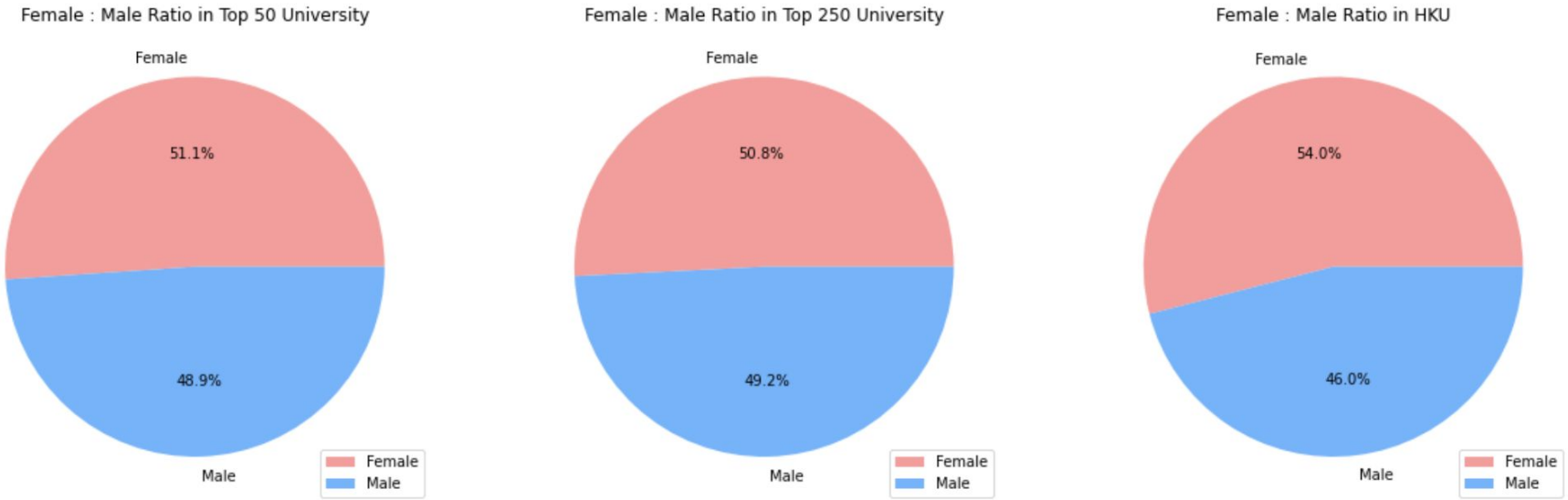
A pie chart: showing the female: male ratio

A bar chart: Compare the number of students per staff and percentage of international students with universities in Hong Kong

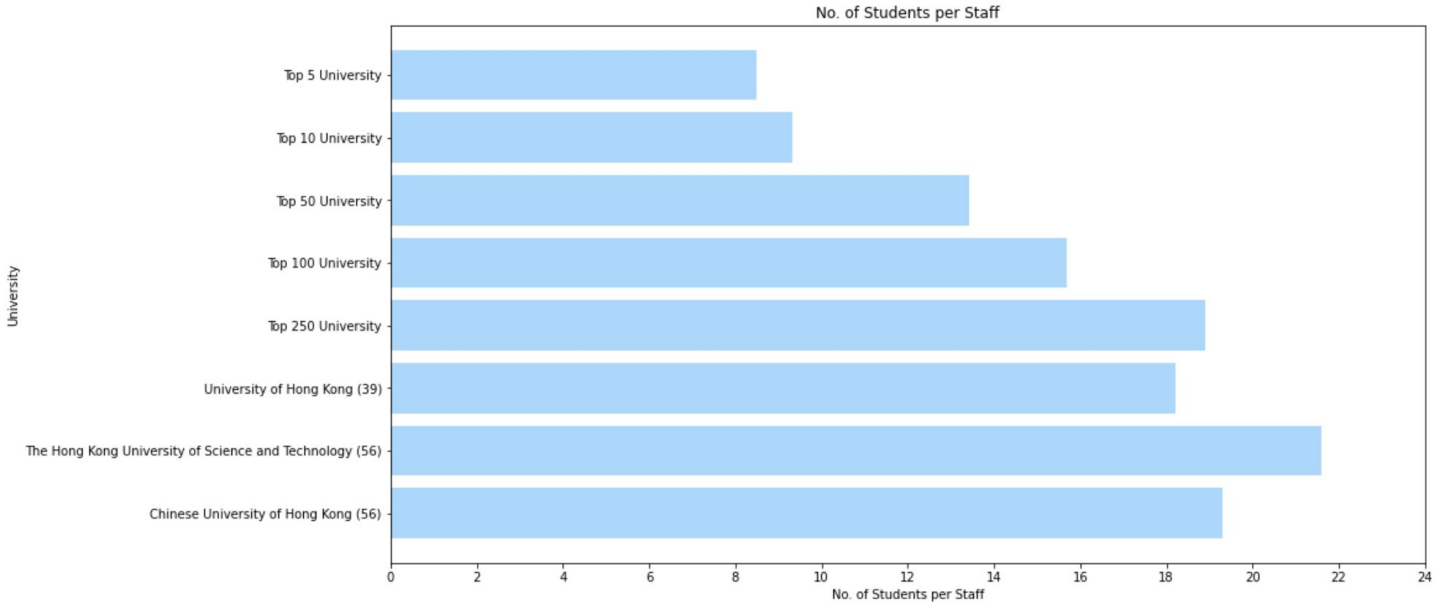



```
others = 0

for i in range(5, 30):
    others = others + uni_df_gp.size().sort_values(ascending=False)[i]


others = others - uni_df_gp.size()['China'] - uni_df_gp.size()['Hong Kong']

others
```

```
labels = ['United States', 'United Kingdom', 'Germany', 'Australia', 'Netherlands', 'China', 'Hong Kong', 'Others']

country   = [uni_df_gp.size()['United States'], uni_df_gp.size()['United Kingdom'], uni_df_gp.size()['Germany'], uni_df

fig,ax = plt.subplots(1,1,figsize=(12,12))

ax.pie(country, labels=labels,autopct='%1.1f%%')
ax.legend(loc='lower right')
ax.set_title('Number of University by Country')
plt.show()

fig.savefig('Number of University by Country.jpg', dpi=1200, bbox_inches='tight')
```

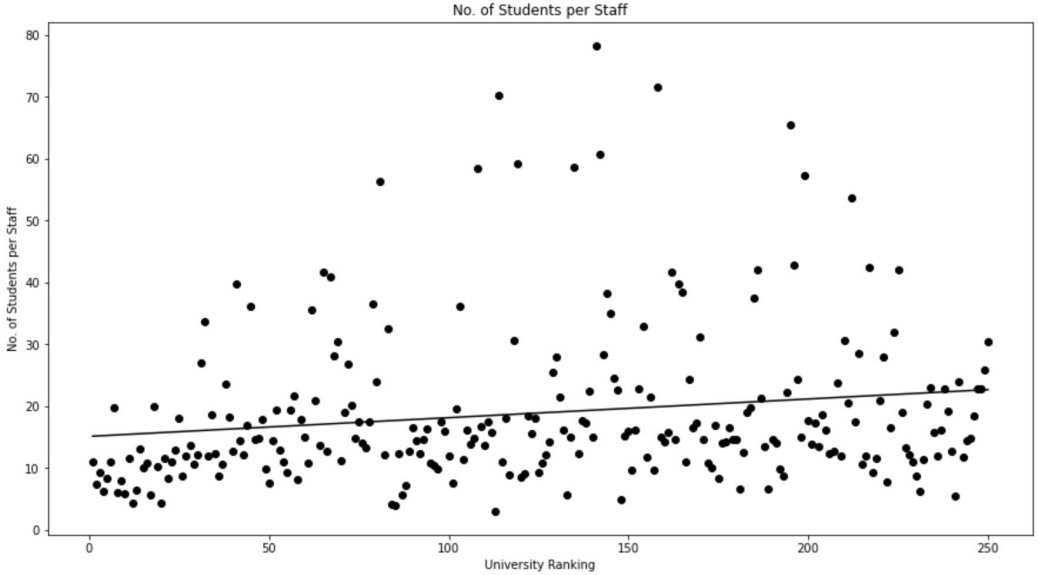



```
labels = ['Female', 'Male']

country   = [uni_df.mean()['Female ratio'], uni_df.mean()['male ratio']]
fig,ax = plt.subplots(1,3,figsize=(20,8))

colorr = ['#ff9999','#66b3ff']

ax1fmr = [uni_df.iloc[38, 8], uni_df.iloc[38, 9]]

mean_female_50_arr = [uni_df[0:50].mean()['male ratio'], uni_df[0:50].mean()['Female ratio']]


ax[0].pie(mean_female_50_arr, labels = labels, autopct = '%1.1f%%', colors = colorr)
ax[0].legend(loc='lower right')
ax[0].set_title('Female : Male Ratio in Top 50 University')


ax[1].pie(country, labels=labels,autopct='%1.1f%%', colors = colorr)
ax[1].legend(loc='lower right')
ax[1].set_title('Female : Male Ratio in Top 250 University')

ax[2].pie(ax1fmr, labels = labels, autopct = '%1.1f%%', colors = colorr)
ax[2].legend(loc='lower right')
ax[2].set_title('Female : Male Ratio in HKU')


plt.show()

fig.savefig('Female : Male Ratio.jpg', dpi=1200, bbox_inches='tight')
```

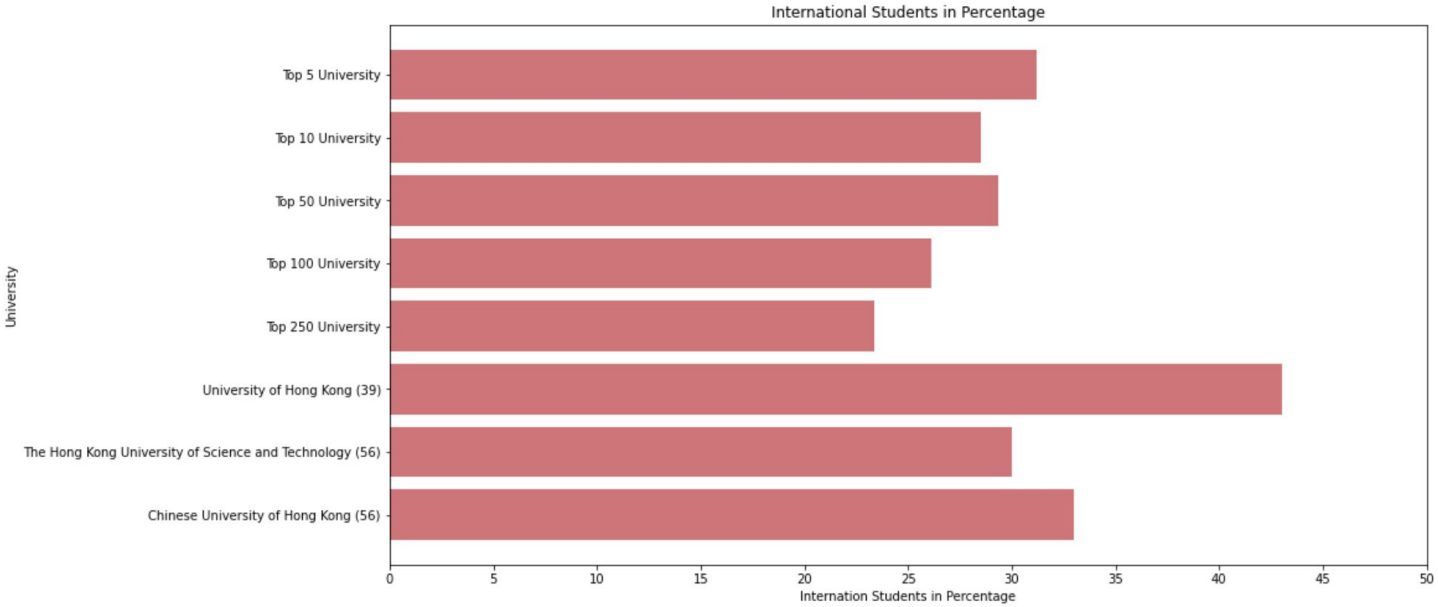



```
data = [['Chinese University of Hong Kong (56)', uni_df.iloc[55, 11]], ['The Hong Kong University of Science and Techn


df = pd.DataFrame(data, columns=['university', 'no. of students per staff'])

fig, ax = plt.subplots(figsize=(15,5))
ax.set_title('No. of Students per Staff')

ax.barh(np.arange(len(df)), df['no. of students per staff'], color = '#A3D7FE')

ax.set_xlabel('No. of Students per Staff')
ax.set_ylabel('University')

ax.set_yticks(np.arange(len(df)))
ax.set_yticklabels(df['university'])

ax.set_xticks([0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24])

plt.show()

fig.savefig('No of Students per Staff.jpg', dpi=1200, bbox_inches='tight')
```

```
The Correlation between University Ranking and No. of Students per Staff is
0.17059188349267337
```

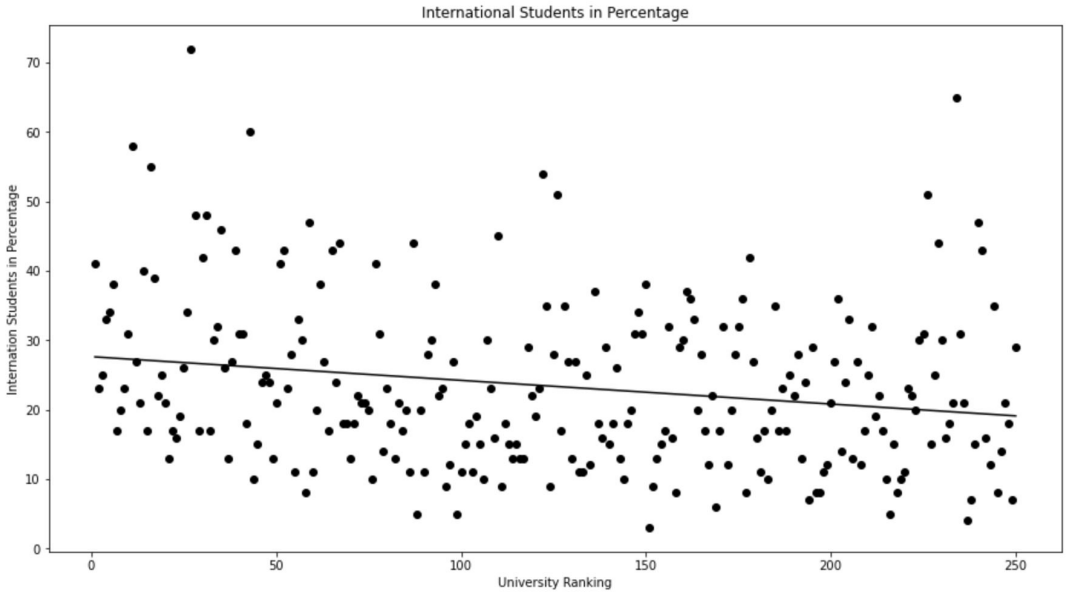



```python
def give_me_a_straight_line(x,y):
    w, b  = np.polyfit(x,y,deg=1)
    line  = w * x + b
    return line

index = range(1, 251)

fig, ax = plt.subplots(figsize=(15,8))
ax.set_title('No. of Students per Staff')


#ax.plot(index, uni_df['Students per staff Int'], 'o', color='black')
ax.scatter(index, uni_df['Students per staff Int'], c = 'k')
ax.plot(index, give_me_a_straight_line(index, uni_df['Students per staff Int']), c='k')

ax.set_xlabel('University Ranking')
ax.set_ylabel('No. of Students per Staff')



print("The Correlation between University Ranking and No. of Students per Staff is")
print(np.corrcoef(uni_df.index, uni_df['Students per staff Int'])[0][1])


plt.show()


fig.savefig('No of Students per Staff line.jpg', dpi=1200, bbox_inches='tight')
```




```
data = [['Chinese University of Hong Kong (56)', uni_df.iloc[55, 7]], ['The Hong Kong University of Science and Techno


df = pd.DataFrame(data, columns=['university', 'internation students in percentage'])

fig, ax = plt.subplots(figsize=(15,5))
ax.set_title('International Students in Percentage')

ax.barh(np.arange(len(df)), df['internation students in percentage'], color = '#DB6F77')

ax.set_xlabel('Internation Students in Percentage')
ax.set_ylabel('University')

ax.set_yticks(np.arange(len(df)))
ax.set_yticklabels(df['university'])

ax.set_xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])

plt.show()

fig.savefig('Internation Students in Percentage.jpg', dpi=1200, bbox_inches='tight')
```

```
The Correlation between University Ranking and Internation Students in Percentage is
-0.20473666744370458
```




```
index = range(1, 251)


fig, ax = plt.subplots(figsize=(15,8))
ax.set_title('International Students in Percentage')


ax.scatter(index, uni_df['International Student Int'], c = 'k')
ax.plot(index, give_me_a_straight_line(index, uni_df['International Student Int']), c='k')

ax.set_xlabel('University Ranking')
ax.set_ylabel('Internation Students in Percentage')

print("The Correlation between University Ranking and Internation Students in Percentage is")
print(np.corrcoef(uni_df.index, uni_df['International Student Int'])[0][1])

plt.show()

fig.savefig('Internation Students in Percentage line.jpg', dpi=1200, bbox_inches='tight')
```

more?

Finding more correlation

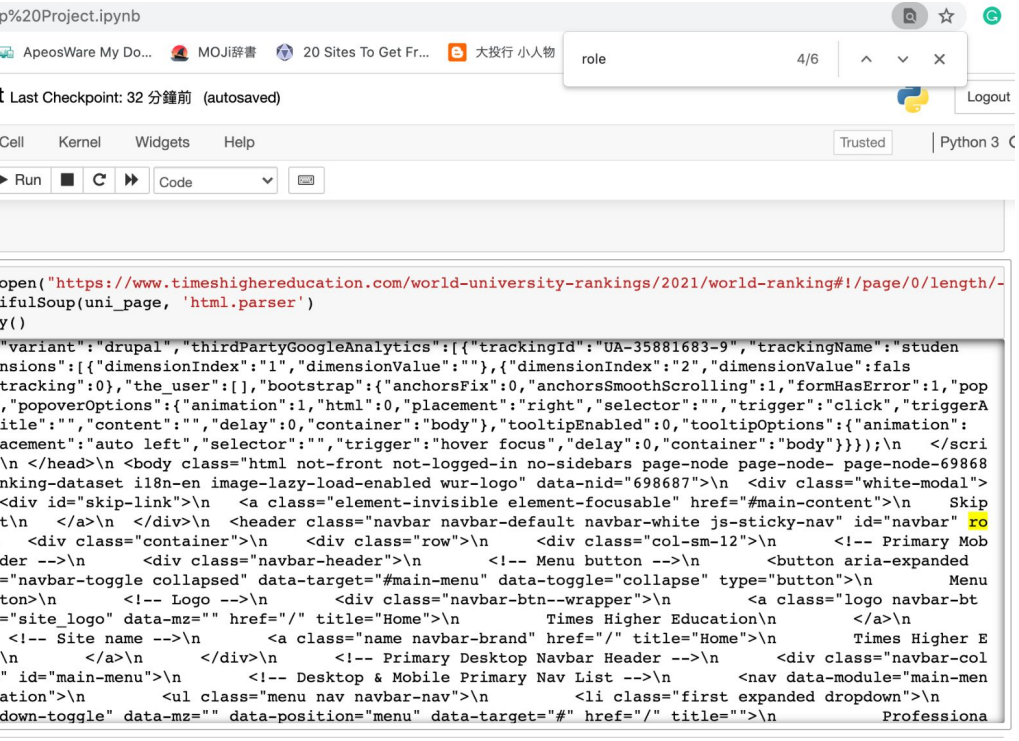
Difficulties/challenges

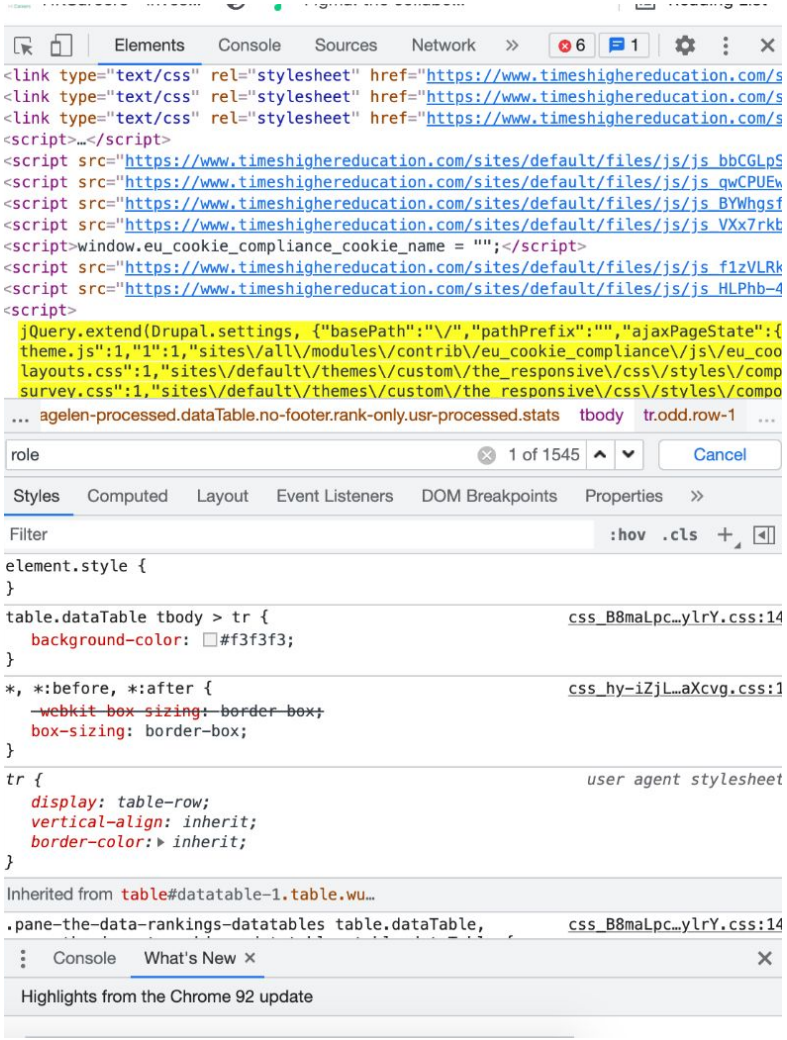
The webpage requires me to sign in

```
imdb_page =
urlopen("https://www.timeshighereducation.com/world-university-rankings/2021/world-ranking#!/page/0/length/-1/sort_by/rank/sort_order/asc/cols/stats")  # open the URL
```

-> Html code is not completed

Difficulties/challenges





Difficulties/challenges

```
driver = Chrome("./chromedriver")

driver.get("https://www.timeshighereducation.com/world-university-rankings/2021/world-ranking#!/page/0/length/-1/sort_by/rank/sort_order/asc/cols/stats")

bs = BeautifulSoup(driver.page_source, 'html.parser')

search_result_list = bs.find_all('tr', {"role": "row"})
```

Difficulties/challenges

```
type(np.nan)
```

```
float
```

Dealing with missing value

Try to convert the string to integer

-> NaN is not string

-> use if-statement to skip them

```
for i in range(0, 250):
    if (isinstance(uni_df['International Student'][i][:-1], str)):
        uni_df.iloc[i, 6] = float(uni_df['International Student'][i][:-1].replace(' ', ''))

    if (isinstance(uni_df['Female: male ratio'][i], str)):
        uni_df.iloc[i, 7] = float(uni_df['Female: male ratio'][i][0:2].replace(' ', ''))

    if (isinstance(uni_df['Female: male ratio'][i], str)):
        uni_df.iloc[i, 8] = float(uni_df['Female: male ratio'][i][-2:].replace(' ', ''))

    if (isinstance(uni_df['No. of students'][i], str)):
        uni_df.iloc[i, 9] = float(uni_df['No. of students'][i].replace(',', ''))

    uni_df.iloc[i, 10] = float(uni_df['Students per staff'][i])
```

Difficulties/challenges

When finding the correlation between university ranking and number of student per staff

Some universities have the same ranking but different number of student per staff

-> same x value, but calculate different y value

-> treat them with different ranking, Duke University: 20, Tsinghua University: 21

=20	Duke University United States Enquire Admissions Support	15,489	4.3	21%	49 : 51
=20	Tsinghua University China Explore	37,484	11.6	13%	34 : 66

Thank You