Summary

I had set out to answer some key and meaningful questions using two very different datasets with one of them being World University Rankings and the other one being World GDP. Both datasets were uncleaned and needed to be cleaned and merged into a csv format to make it easier to analyse data and answer the questions which I had set out and wanted answered.

Python was used for the wrangling the data and merging the two datasets, this was mainly done due to the use of the Pandas library which made it easier to work on the uncleaned data and clean it so that the following questions could be answered:

1. Does a high GDP per capita mean a higher university rank?
2. Do higher populated countries have better universities?
3. Which region and which of the sub regions in that region has the best teaching?

After cleaning all questions were answered and some extra components were added to the answer to get some more interesting information out of the dataset.

Question 1 - Does a high GDP per capita mean a higher university rank?

- Using GDP per capita from the GDP dataset and the total score from University World rankings dataset the result turned out to be true in that generally a higher GDP per capita will indeed result in on average a higher university rank. Along with that there was some interesting information that when taking into consideration only the highest ranked university the higher to middle GDP per capita total scores increase quite a bit while the lower GDP per capita doesn't increase much in the total score

Question 2 – Do higher populated countries have better universities:

- By using population rank and average total score for universities I found that there is no relation between population and university ranking as there is no clear increase on the graph and instead its very random meaning that population and university rankings are not related.

Question 3 – Which region and which of the sub regions in that region has the best teaching?:

- I was curious in knowing which region has the best teaching and out of those regions which of the sub regions have the best teachings. The results were as expected in that Americas has the best average teaching and north America has the best average teaming out of the America however what was interesting is that when looking at the average the deviation between the regions wasn't that big.

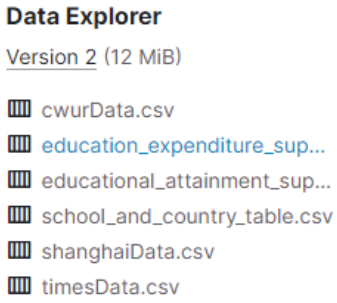
Wrangling Details

Prior to starting the wrangling process ensure that Python pandas library has been downloaded and ensure the code is written in python as the wrangling process will be done in python using the Pandas library.

Dataset 1 – University Rankings

- https://www.kaggle.com/datasets/mylesoneill/world-university-rankings?resource=download

The first dataset is from Kaggle and its about University Rankings throughout the world. When downloading from this source it comes with six 6 different datasets:



However only three of those datasets are relevant to this project the cwurData.csv, shangaiData.csv and timesData.csv and each of these datasets represent University Rankings but done by different sources and of them times.csv which is by Times is considered to be the most influential and widely observed dataset so using that information I had decided to use times.csv as my dataset.

The dataset used is in a csv tabular format and has 14 columns and 2604 rows, the columns are as follows:

Column	Description
world_rank	Overall ranking of the university
university_Name	Name of the university
country	The country which the university is in
teaching	A score given to the university based on the learning environment
international	A score given to the university based on its international outlook
research	A score given to the university based on its research volume, income, and reputation
citations	A score given to the university based on its research influence
income	A score given to the university based on its industry income
total_score	The score given to a university based on its teaching, international, research, citation, and income score which is used to determine the rank
num_students	Number of students at the university
student_staff_ratio	Student to staff ratio (number of students divided by the number of staff)
international_students	Percentage of students which are international
female_male_ratio	Female student to male student ratio

year	Year of the rankings (2011 – 2016 included)

When going through the first few rows of data I realised that there were some missing values in the income column and the international column. I knew that the values of the income and international columns helped in calculating the total score so if those values were missing then how was the total score calculated? So I did some research and found the times methodology used in ranking the University's in 2015 – 2016 (https://www.timeshighereducation.com/news/ranking-methodology-2016). Here is what I found from that:

Total score is made up using five categories and each of those categories were given a weight towards the ranking:

- Teaching = 30%
- Research = 30%
- Citations = 30%
- International = 7.5%
- Income = 2.5%

Not all five categories helped equally in creating the total score so using that information I decided to remove any records which were missing values for the major categories of:

- Teaching
- Research
- Citations

I feel that if any of those 3 categories don't have a value then the specific ranking loses a lot of credibility in terms of overall ranking. I removed them using the inbuilt methods of the panda's library in python. In python anything that was considered null would be 'nan' and anything that was meant to be blank but wasn't cleared was given a '-' as seen below

The following code is used to remove any record where teaching, research or citation is null or has a '-' for its value.

```
# drop any record if teaching, international or citation is empty
df_csv = df_csv.drop(df_csv[(df_csv.teaching == "-")].index)  # drops row if the record has a - as a value for teaching
df_csv = df_csv.drop(df_csv[(df_csv.research == "-")].index)
df_csv = df_csv.drop(df_csv[(df_csv.citations == "-")].index)
df_csv = df_csv.dropna(
    subset=['teaching', 'research', 'citations'])  # drops any records in which teaching, research or citiations is null
```

Once the major categories were sorted I dealt with the minor categories, Income and international. As the minor categories don't have a massive impact to the overall score compared to the major categories I felt that it would be best to just let them through.

There were many records which doesn't have a value in the total score column which is bad for the overall dataset as then we cant really validate that the University Ranking is correct either. So to deal with this I manually added total scores for any records which were missing it using the Times Methodology. I had already removed records without a value in the Teaching, Research or Citation columns so the most significant performance indicators will all be present when doing the calculation while there may be some Income and International columns records which don't have a value so set the minor categories without values to 0.

```
df_csv.loc[df_csv.citations == "-", 'international'] = 0  # if the record has a - for its citation value then change to a 0
df_csv.loc[df_csv.income == "-", 'income'] = 0  # if the record has a - for its income value then change to a 0
df_csv.loc[pd.isnull(df_csv.citations), 'international'] = 0  # if the record has a nan for its citation value then change to a 0
df_csv.loc[pd.isnull(df_csv.income), 'income'] = 0  # if the record has a nan for its income value then change to a 0
```

As the International and Income columns don't have a massive impact on the total score calculation, the margin of error (**include percentage if possible)** will only be slightly off compared to what Time's value should be. The following calculation is used based on the category weights to get the total score:

Total score = (0.3 x Teaching) + (0.3 x Research) + (0.3 x Citation) + (0.075 x International) + (0.025 x income)

Using the following code the total_score was replaced with the estimated total_score

```
# for loop used to find all empty total scores and replace them with an estimated total score
for x in range(len(df_csv['total_score'].values)):
    # the following code will calculate the total score if the total score is empty and the calculation is done based on the University Rankings performance weights
    # it rounds the total to 1dp and i used the df_csv.info to check for the object data type and converted it to float if needed
    if df_csv['total_score'].values[x] == '-' or pd.isnull(df_csv['total_score'].values[x]):
        df_csv['total_score'].values[x] = (round(
            (df_csv['teaching'].values[x] * 0.3) +
            (df_csv['research'].values[x] * 0.3) +
            (df_csv['citations'].values[x] * 0.3) +
            (float(df_csv['international'].values[x]) * 0.075) +
            (float(df_csv['income'].values[x]) * 0.025), 1
        ))
```

As international and income weren't already a float datatype they had to be converted to match the datatype of research, teaching, citations and total_score. The total score was rounded to the nearest 1dp as when observing the other total scores there always had 1 dp so I felt that it would be best if the estimated total_score was as close to the actual total_score format as possible.

Dataset 2

- https://www.kaggle.com/datasets/darknez/gdp-among-world?select=GDPfinal.json

The second dataset is also from Kaggle, this dataset is about the world GDP for countries in 2020. The dataset that was used is the GDPfinal.json which contains the world GDP in a json format. The dataset contains a single array with 184 objects in it and each of the 184 objects represents a country and contains 18 string properties which are the following:

Object Property	Description
Country	Country Name
Population Rank	The population rank in the world

Growth Rate	Rate at which the country is growing
World Percentage	Percentage of the world population in the country
Density	Population density
Land Area	Land area which is cultivated and established
2020 Population Rank	2020 population rank
2020 World Percentage	2020 world percentage
2020 Growth Rate	Growth in 2020
Area	Overall area of the country
Capital City	Capital city of country
Region	The region the country is in
Subregion	Which part of the region the country is in
Anthem	
Government	The ruling government of that country
GDP(IMF)	GDP value calculation via the International Monetary Fund
GDP(UN)	GDP value calculated via the United Nations
GDP Per Capita	Economic output of a nation per person

From this dataset I mainly needed the GDP records along with region, subregion and population bringing in some interesting research elements which I can use to analyse the data and get some interested results. As such majority of the other properties such as Area, Capital City, Anthem weren't needed for the final result. As the dataset 1 for University rankings is from 2011 – 2016 I decided to ignore the World GDP of 2020 so I removed all 2020 data. I also got rid of Anthem, Government, Capital City, Land Area, and Area as they weren't relevant to the information that I was wanting out from the final result along with the fact that many of records didn't have any data for those properties so majority of these properties were useless. In order to remove those properties the following code was used:

```
del df_json['Anthem']
del df_json['Government']
del df_json['2020 Population Rank']
del df_json['2020 World Percentage']
del df_json['2020 Growth Rate']
del df_json['Capital City']
del df_json['Land Area']
del df_json['Area']
```

Another thing which I observed in this dataset is that there were some GDP Per Capita values which were either null or had a '-' for it and I felt that due to my questions wanting to extract information in terms of GDP per capita and there being no way to create a GDP

Per Capita value I removed any records which didn't have a GDP per capita, the following code was used for this:

```
df_json = df_json.drop(df_json.index[df_json['GDP Per Capita'] == "-"])
df_json = df_json.dropna(subset=['GDP Per Capita'])
```

After the removal of empty GDP per capita records this dataset didn't have much more that was off about it.

Merging

The following line of code was used to merge the two datasets together. The code merges

```
result_dataset = pd.merge(df_csv, df_json, left_on='country', right_on='Country')
```

the csv file json file based on the Country columns which both have and as there is no 'how=', the merge is defaulted to a inner join where the countries that are shared in both the databases are kept. When joining I am mainly interested in having the data which can help to answer the questions I had set out therefore having a University Ranking record which has no country to merge with or vice versa it becomes useless as there is no relationship between the university ranking to a countries GDP hence why inner join was used to ensure that records which have matching values in both datasets are kept.

Following the first merge I noticed that certain universities were missing such as universities from the US. Hence I used the following code which gets a list of all the countries in the

```
# check if all the countries in University Rankings are present in the gdp dataset
csv_countryList = []  # list for all of the countries in the University Ranking database
for i in df_csv['country'].unique():
    csv_countryList.append(i)
csv_countryList.sort()
print(len(csv_countryList))

json_countryList = []  # list for all of the countries in the GDP database
for i in df_json['Country'].unique():
    json_countryList.append(i)
json_countryList.sort()
print(json_countryList)

#Code to find countries not in JSON databse
for i in csv_countryList:
    if i not in json_countryList:
        print("Not in JSON dataset: " + i)
```

university ranking dataset and all the countries in the gdp dataset and compares it so that any countries in the university rankings dataset which are not in the gdp dataset are printed as seen below:

```
Not in JSON dataset: Hong Kong
Not in JSON dataset: Macau
Not in JSON dataset: Republic of Ireland
Not in JSON dataset: Russian Federation
Not in JSON dataset: Unisted States of America
Not in JSON dataset: United States of America
Not in JSON dataset: Unted Kingdom
```

Seeing the list above I can tell some a clearly spelling mistakes such as Unted Kingdom and Unisted States of America, after searching through the country list in the GDP dataset using ctrl + f, I saw that Ireland, Russia and United States do indeed exist however they are spelt differently in the GDP dataset:

- Republic of Ireland = Ireland in GDP dataset
- Russian Federation = Russia in GDP dataset
- United States of America = United States in GDP dataset

To fix this issue I iterated through all of the country values in the university rankings(CSV) dataset and if the country values were either Republic of Ireland, Russian Federation or United States of America then I changed them to Ireland, Russia, and United States respectively and to do so the following code was used:

```
for x in range(len(df_csv['country'].values)):  # change the name of countries in University Ranking Dataset to match GDP dataset
    if df_csv['country'].values[x] == "Republic of Ireland":
        df_csv['country'].values[x] = "Ireland"

    elif df_csv['country'].values[x] == "Russian Federation":
        df_csv['country'].values[x] = "Russia"

    elif df_csv['country'].values[x] == "Unisted States of America":
        df_csv['country'].values[x] = "United States"

    elif df_csv['country'].values[x] == "United States of America":
        df_csv['country'].values[x] = "United States"

    elif df_csv['country'].values[x] == "Unted Kingdom":
        df_csv['country'].values[x] = "United Kingdom"
```

To deal with the other two countries which weren't present in the GDP dataset of Hong Kong and Macau I decided to do some research to see if they were called anything else but there were no results so I removed any records which contained University Rankings from either of those countries, to do so the following code is used:

```
df_csv = df_csv.drop(df_csv[(df_csv.country == "Hong Kong")].index)  # drop any records of Universities from Hong Kong
df_csv = df_csv.drop(df_csv[(df_csv.country == "Macau")].index)  # drop any records of Universities from Macau
```

One final step I took before merging the document was that I removed the records which were recorded in 2011, 2012, 2013, 2014 and 2015. This was due to the fact that the GDP information for those years wasn't going to change and as I am wanting to find the relationship between the GDP and the University Ranking it wouldn't make a different if it is for 2011 or 2015, I just chose 2016 as that was the most recent year available. The following code was used to remove records of 2011 – 2015:

```
df_csv = df_csv.drop(df_csv[(df_csv.year == 2011)].index)
df_csv = df_csv.drop(df_csv[(df_csv.year == 2012)].index)
df_csv = df_csv.drop(df_csv[(df_csv.year == 2013)].index)
df_csv = df_csv.drop(df_csv[(df_csv.year == 2014)].index)
df_csv = df_csv.drop(df_csv[(df_csv.year == 2015)].index)
```

Once merged there was one small thing which I felt wasn't needed which was that due to both datasets having a country column the merged dataset ended up with two so prior to converting it to a CSV I removed one of the country columns like so:

```
del result_dataset['Country']
```

Converting

Once the dataset was merged there wasn't too much that needed to be done in order to convert it to a CSV file type. One line of code was all it took to convert it to a CSV:

```
result_dataset.to_csv('C:/Users/rinke/OneDrive/Desktop/Uni/2022/Sem 1/Busan 300/A5 - Project/Dataset/finalDataset.csv', encoding='utf-8', index=False)
```

After running that code the finalDataset.csv was created at that file destination I had set.

After converting I opened Excel and loaded the data in from a text/csv, when selected the load options I chose to not detect the data types as if the data types are detected then female_male_ratio column ends up becoming a date format. However once put into a excel worksheet all the columns are treated as a string and to convert each column back to an integer or float would be a pain so instead I changed the format in python so that instead of ff:mm it becomes ff – mm so (33:67 becomes 33 – 67) using the following code:

```
#Code to change the female to male ration format from ff:mm to ff - mm
for x in range(len(df_csv["female_male_ratio"].values)):
    if not pd.isna(df_csv["female_male_ratio"].values[x]) and df_csv["female_male_ratio"].values[x] != "-":
        df_csv["female_male_ratio"].values[x] = (df_csv["female_male_ratio"].values[x].split(':')[0] + "-" + df_csv["female_male_ratio"].values[x].split(':')[1])
```

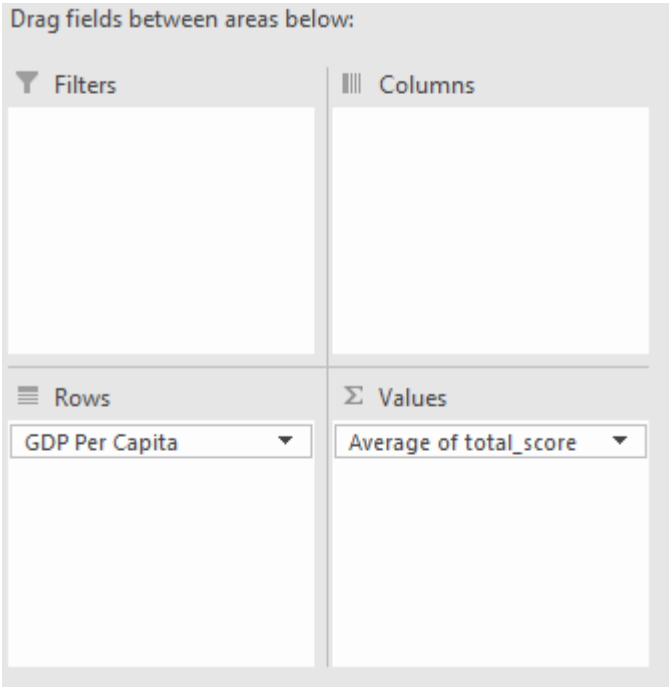
After changing the format of female_male_ratio I used utf-8 and based on the entire dataset to load the data into excel. Once the dataset was loaded into excel there were some values

in the world rank columns which were '=num' so using the find all and replace all in that column I replaced all the = with blank so that only the number was left. I also replaced all the blank space with 123a654b and then replaced 123a654b with blank space to ensure that it was actually blank and will have no affect on the data.

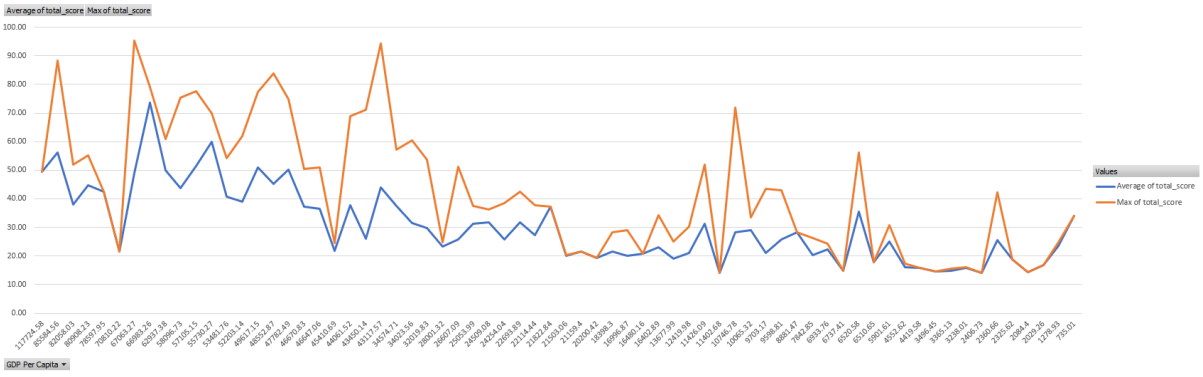
Questions and Answers

Question 1 - Does a high GDP per capita mean a higher university rank?

I used a pivot table and a pivot chart to get the answer for this question. In order to analyse this question I created a pivot table using the dataset from the converted CSV using GDP per Capita from the GDP dataset (JSON and total_score from University Rankings (CSV) as total score is directly related to the world ranking for a university. In order to do so I placed the GDP per capita under the Row area and the total_score under the Values area and took the average of the total score like so:



I then included a pivot column line for more clarity which can be seen below:



The x axis contains the GDP per capita ordered from largest on the left to the smallest on the right while the y axis contains the average total score of universities.

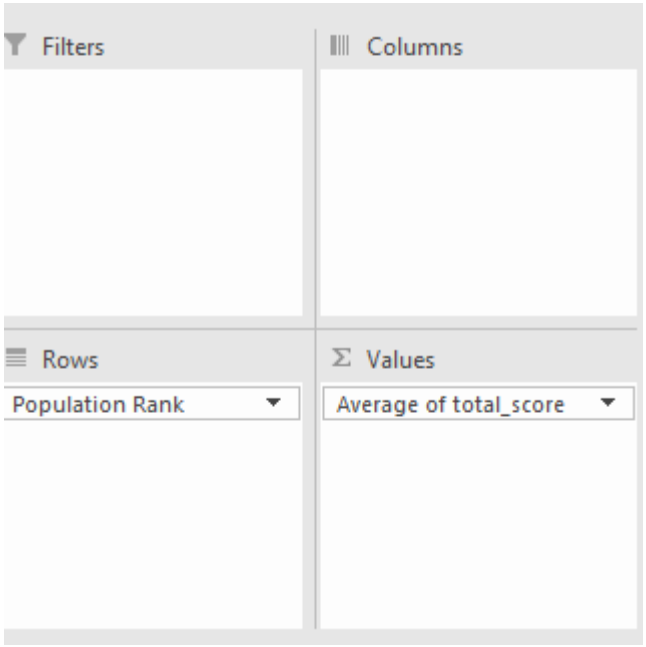
From the results it can be see that generally a higher GDP per capita does result in a better university overall however that may not always be the case as there is a GDP value which is 70810.22 which has an average total score of 21.6.

Sometimes there is the case that certain countries will have top universities which are ranked significantly higher than every other university in that country so I also used the max of the university score at that GDP per capita to see the result and interestingly enough the max total_score doesn't change much for the higher GDP per capita but it does seem to change a decent amount for the lower GDP countries.

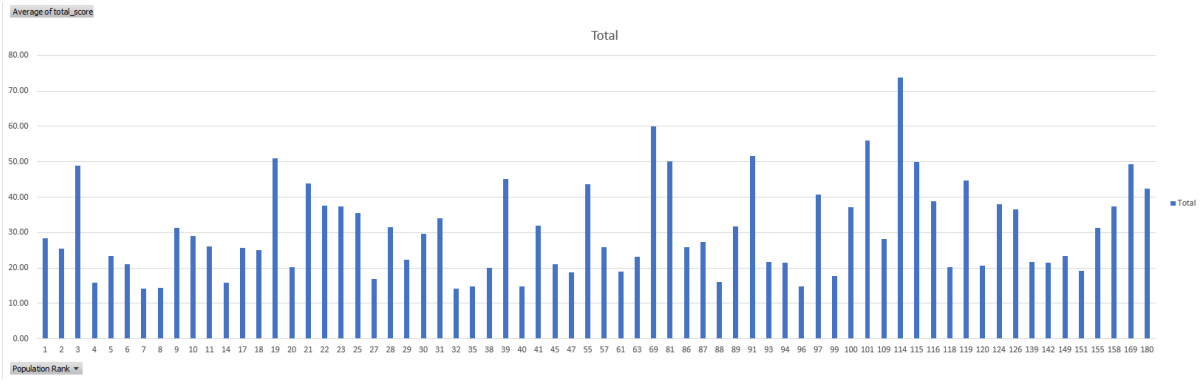
Question 2 – Do higher populated countries have better universities?

I again used a pivot table to analyse the data for this question. For this question I wanted to know if there is a direct relation to population and university score, whether lower populated countries have worse of Universities as there are less people or if higher populated countries have a better Universities due to there being more people.

I used population rank from the GDP dataset and total score from the university ranking dataset. I ordered population rank from smallest to largest to get the country ranked with the highest population at the top and the country with the lowest population at the bottom. The following was used to get the information:



I then inserted a bar graph to get a visual representation to better understand the data:

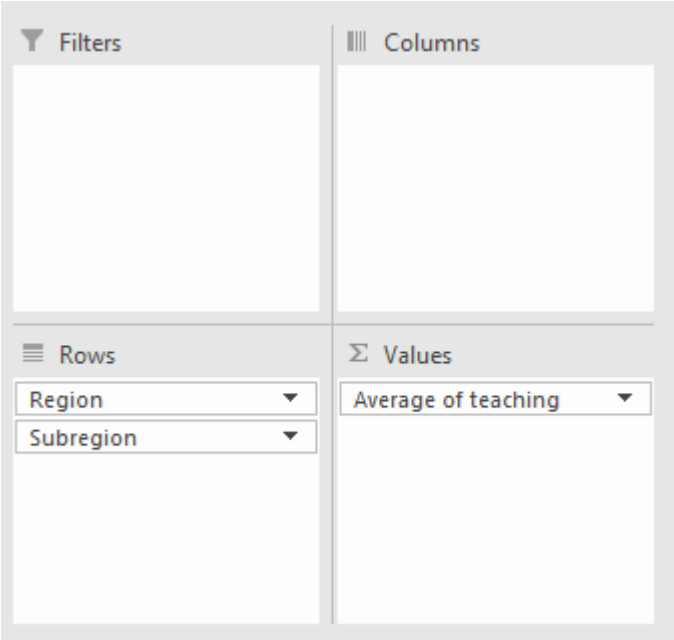


The y axis has the population rank from lowest to highest (population rank of 1 means high population while population rank of 180 means low population)

Looking at the results the data is very scattered showing that there isn't really any sort of relationship between population rank and university score. This means that rather than population there is probably some other fact leading to higher ratings.

Question 3 – Which Region and which of the sub regions in that region has the best teaching?

Generally most students go to University for the teaching so it would be interesting to know what region or rather which sub-region in that region has the best teaching. To answer this I used the following data:



For this question I didn't insert a graph as I felt that it was clear just using the table:

Row Labels	Average of teaching	Max of teaching
Africa	**21.42**	**34.9**
Western Africa	19.15	22.7
Southern Africa	25.60	34.9
Northern Africa	19.80	22.6
Eastern Africa	14.40	15.1
Americas	**38.12**	**95.6**
South America	26.05	59.1
Northern America	40.00	95.6
Central America	33.30	42.7
Asia	**27.85**	**81.4**
Western Asia	22.64	45.8
Southern Asia	28.88	51.8
South-Eastern Asia	26.49	71.7
Eastern Asia	28.90	81.4
Europe	**30.62**	**88.2**
Western Europe	36.91	77
Southern Europe	24.38	54.2
Northern Europe	31.11	88.2
Eastern Europe	25.55	75.4
Oceania	**27.97**	**62**
Australia and New Zealand	27.97	62
Grand Total	**31.54**	**95.6**

I also added the max teaching values just because of interest and wanting to see some more information. Based on the results its clear that on average America has the best teaching and on the max scale America still has the best teaching but what is note worthy is that on average scale the variation between the regions isn't that big compared to the variation of sub regions such as north America has a average of 40 and a max of 95 compared to eastern Africa which has an average of 14.40 and a max of 15.1. Overall this answer was as expected but the max values were a little surprising in that sense.

Link to download the Worksheet and Code

https://drive.google.com/drive/folders/15zZ9d6VZdHn1kctWFtSb1X8A5_6yo-BZ?usp=sharing