Introduction, Aims, and Objectives

Introduction

In this project, I will be looking at three of Singapore's local universities to explore the correlation between a university's ranking, and how it has affected the employment outcome of its fresh graduates across the years 2015 to 2021. Additionally, analysis will be carried out to determine whether it is true that university rankings no longer affects the employment outcomes of its fresh graduates as much as it did in previous years.

Background

University ranking has always been a universal topic of discussion when determining which university is considered more recognised, more prestigious, and better able to guarantee good employment outcomes for its fresh graduates. University ranking systems such as Times Higher Education(THE) **[1]**, QS Universities Rankings **[2]**, Academic Ranking of World Universities(ARWU) **[3]**, and many more, are not only widely used by students, but also by companies to identify potential employees. This is despite the fact that the evaluation criterias used by these systems to determine those rankings are not necessarily a good measure of the technical abilities and employability of each university's fresh graduates.

In Singapore, the general consensus has always been that fresh graduates from a higher ranking local university would be offered better employment opportunities, and a higher starting salary upon graduation, as compared to fresh graduates from a lower ranked local university. This has led to a bias between two fresh graduates where even if both are hired for the same position in the same company, the fresh graduate from the higher ranked university receives better employee benefits and starting salary, even as both offer the same set of skills at the same level of proficiency.

This has also created a trend of students looking to pursue undergraduate studies choosing a university based on its ranking, rather than choosing based on which university would best suit their needs.

However, over the years, there has been an increasing number of discussions made to argue that more companies in Singapore are now evaluating fresh graduates based on their technical abilities. Thus, providing equal opportunities and benefits to all fresh graduate applicants regardless of the university they had graduated from. As such, suggesting that students in Singapore should no longer use a university's ranking as the main factor of consideration when deciding which university to study in.

Field of interest

As a university student myself, I had also taken university rankings into consideration when deciding which university to apply to. This topic piqued my curiosity as to whether it is true that a university's ranking no longer matters as much as it did years ago, with regard to the employment outcome of university fresh graduates from local universities in Singapore.

Hence, I will be carrying out analysis to test the two following hypothesis:

1. There **is** a correlation between university rankings and employment outcome.
2. It is **true** that university rankings no longer affects employment outcomes of local university fresh graduates as much as it did years ago.

Aims and Objectives

Aims

The main aim of my analysis is to explore how a university's ranking affects the employment outcome of its fresh graduates over the years, as well as to determine whether university rankings still affects employment outcomes of fresh graduates as much as it did in previous years.

Objectives

1. Determine the ranking baseline of the three local Singapore universities across the years by comparing the ranking outcomes of two reputable ranking systems for accuracy.
2. Determine employment outcomes by comparing the difference in salary and employment rate of fresh graduates between different universities.
3. Observe any trends between points (1)ranking of each university, and (2)employment outcome to determine if there are any correlations.
4. Compare the employment outcomes across the years 2015 to 2021 to observe whether university ranking still affects employment outcome.
5. Plot graphs to provide better visualisation of data and trends.

Scope of work and Data pipeline

Scope of work

Since the aim of my analysis focuses on university rankings, I will limit my scope such that only Singapore universities that have consistently appeared in the rankings are analysed. Hence, the following three local Singapore universities will be the focus of this analysis:

- National University of Singapore (NUS)
- Nanyang Technological University (NTU)
- Singapore Management University (SMU)

I will also only be looking at the years 2015 to 2021, as these are the years where at least two of the universities above are ranked.

In order to determine the ranking baseline of the three universities listed above, I will be comparing the world university ranking results from QS World University Rankings and THE World University Rankings over the years 2015 to 2021. If there are no major differences between the ranking results for all three universities in each year, that ranking will be my baseline. Otherwise, I will make further comparisions with other ranking systems to determine the most recurrent ranking result.

Employment outcome will be determined by looking at the full-time employment rate and median basic monthly salary of fresh graduates from each university.

(Note: From this point onwards, I will be referring to the three universities(NUS, NTU, SMU) and two ranking systems(QS, THE) using their respective acronyms.)

Data processing pipeline

1. Prepare and clean the two datasets containing university rankings
2. Compare both datasets to find any differences in ranking over the years 2015 to 2021.
3. Set the base rank (the ranking that will be used for the rest of the analysis).
4. Clean and prepare the dataset containing graduate employment details
5. Perform analysis to test the first hypothesis
    - Plot graph to show the difference in overall employment rate and basic monthly median salary of the three universities
    - Identify the trend (which university shows the highest)
    - Reference back to the ranking (does higher rank = higher employment rate and monthly median salary?)
6. Perform analysis to test the second hypothesis
    - Compare the difference in employment rate and median salary between universities over the years 2015 to 2021.
    - Is there a smaller difference in year 2021 as compared to the years before?
7. Summarise all findings and state whether each of the two hypothesis tested out to be true or false.

Data Sources, Description, and Ethics of use

Data Sources

1. (Web Scraping) Singapore region QS World University Rankings all years: https://www.universityrankings.ch/results?ranking=QS&region=World&year=all+years&q=Singapore

2. (Web Scraping) Singapore region THE World University Rankings all years: https://www.universityrankings.ch/results?ranking=Times&region=World&year=all+years&q=Singapore

3. (CSV Export) Graduate Employment Survey: https://data.gov.sg/dataset/graduate-employment-survey-ntu-nus-sit-smu-suss-sutd

Dataset description and relevance

In this project, 3 different sets of data have been used to carry out analysis. Two of which are collected through web scraping.

Data description

1. The first two sets of data contains world university ranking results by **QS rankings and Times Higher Education ranking systems**, containing only results of Singapore universities throughout years 2004 to 2023. Although the original ranking results are avilable for csv export from the official ranking websites, csv export does not work after applying filters to the table (e.g only display Singapore's universities). Therefore, web scraping would be used to scrape the filtered table containing only ranking datas required.

2. The third dataset "graduate-employment-survey" was exported as a csv file from **Data.gov.sg**. This dataset consists of data collected from fresh graduates from six local universities in Singapore: NTU, NUS, SIT, SMU, SUSS, and SUTD. Through an exercise called the **Graduate Employment Survey(GES)** held anually, fresh graduates from the six universities are invited to respond to a survey consisting of questions related to their current employment conditions approximately six months after their final university examination. The dataset contains results from 1st January 2013 to 31st December 2021.

Relevance of data to research topic

- The first two sets of data will be used to determine a ranking baseline for the six local Singapore universities I will be analysing. Comparing between two ranking sources allows for more accuracy in determining a rank.
- The third dataset will allow me to determine employment outcomes of fresh graduates as it contains columns such as employment rate, mean and median basic monthly salary.

Other datasets considered

- I had also considered using QS Asia University Rankings **[4]** which is a ranking system with only Asian Universities. This ranking might provide a more accurate representation of ranks due to the differences between Asian and Western education system and structure. However, this ranking system only started in year 2020 so there is a lack of data, and therefore not suitable to be used for my analysis.

Ethics of use of data

1. Data sources (#1) and (#2) are from **UniversityRankings.ch**, a site showing results of world university rankings from major ranking systems throughout the years, is run by the State Secretariat for Education, Research and Innovation (Switzerland), and swissuniversities. As per the Terms and conditions **[5]**, information on websites run by the Swiss federal authorities is accessible to the public.
2. Data source (#3) "Graduate Employment Survey" dataset was published on **Data.gov.sg** by The Ministry of Education, Singapore. Data.gov.sg is a Singapore Government Agency website containing datasets published by public agencies in Singapore, and made available to the public. Source acknowledgement and Licence of the dataset can be found under the references section **[6]**.

Import libraries and packages

In [1]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
```

Data Acquisition

Web Scraping

In order to determine a ranking baseline, I will be performing web scraping to extract data containing an overview of ranking results of the six local Singapore universities, from two different ranking systems. The website **universityrankings.ch** allows filtering by preferred ranking system, period (years), and a region. The first url I am webscraping from is the search results from filtering by "QS World University Ranking", for the period of "all years", with the keyword "Singapore". The second url is the search result after filtering by "Times World University Rankings", for the period of "all years", with "Singapore" keyword.

- qs_url : QS World University Rankings (filtered by 'Singapore' region)
- the_url: THE World University Rankings (filtered by 'Singapore' region)

Helper function

A helper function send_request() has been defined below to reduce code repetition. The function sends a GET request to the url passed in if the url is valid (webpage exists). If the GET request is successful (status code 200), the html of the webpage is parsed and the page title is printed. The function then returns the parsed html. If the request is unsuccessful, the status code will be returned.

In [2]:
```
def send_request(url):
    try: # check if url is valid
        response = requests.get(url)

    except Exception: # url is invalid
        return "url is not valid."

    else: # url is valid
        if response.status_code == 200:
            print("request successful!")
            soup = BeautifulSoup(response.text, "html.parser")

            # print title
            if soup.title is not None:
                print("Title: ", soup.title.string)
            else:
                print("No title found.")

        else: # unsuccessful request
            soup = response.status_code
            print("request unsuccessful.")
        return soup
```

Carry out web scraping

Pass both urls into the helper function to scrape the table.
If successful, the title should be printed by the function for verifying that the correct html was parsed.

In [3]:
```
# specify both urls to webscrape from
qs_url = "https://www.universityrankings.ch/results?ranking=QS&region=World&year=all+
the_url = "https://www.universityrankings.ch/results?ranking=Times&region=World&year=

# get parsed html if url is valid
qs_soup = send_request(qs_url)
the_soup = send_request(the_url)
```

```
request successful!
Title:  QS Ranking all years - Singapore - Results | UniversityRankings.ch
request successful!
Title:  Times Ranking all years - Singapore - Results | UniversityRankings.ch
```

Convert table into pandas dataframe

By inspecting the source website, it can be seen that the table containing the ranking results is enclosed in a `<table>` tag, with the class "RankingTable". Hence, I will use BeautifulSoup's find() function to extract the table from the parsed html, before using pandas' read_html() function to convert the table into a dataframe. This will be done for both QS ranking(qs_soup) and THE ranking(the_soup).

In [4]:
```
# extract table
qs_table = qs_soup.find("table", { "class" : "RankingTable" })
the_table = the_soup.find("table", { "class" : "RankingTable" })

# convert into df
qs_df = pd.read_html(str(qs_table))[0]
the_df = pd.read_html(str(the_table))[0]
```

In [5]:
```
# show rows and columns of both dataframes
print('QS data: ', qs_df.shape)
# show first 5 rows of both dataframes
display(qs_df.head())

print('THE data: ', the_df.shape)
display(the_df.head())
```

```
QS data:  (47, 6)
```

	Unnamed: 0	World Rank▲▼	Institution▲▼	Country▲▼	Year▲▼	Unnamed: 5
0	1	11 ▲	Nanyang Technological University	NaN	2020	NaN
1	2	11 ▲	Nanyang Technological University	NaN	2018	NaN
2	3	11 =	National University of Singapore	NaN	2023	NaN
3	4	11 =	National University of Singapore	NaN	2022	NaN
4	5	11 =	National University of Singapore	NaN	2021	NaN

```
THE data:  (26, 6)
```

	Unnamed: 0	World Rank▲▼	Institution▲▼	Country▲▼	Year▲▼	Unnamed: 5
0	1	19 ▲	National University of Singapore	NaN	2023	NaN
1	2	21 ▲	National University of Singapore	NaN	2022	NaN
2	3	22 ▲	National University of Singapore	NaN	2018	NaN
3	4	23 ▼	National University of Singapore	NaN	2019	NaN
4	5	24 ▲	National University of Singapore	NaN	2017	NaN

Verify scraped data

In the source webpage, the table containing QS Rankings should contain 47 rows and six columns in total, and the table containing THE Rankings should have 26 rows and six columns in total. Based on the output of the cell above, both 'qs_df' and 'the_df' dataframes have the exact number of rows and columns expected.

However, it can be observed that there are two unnamed column, and two columns containing NaN values. Referring back to the source webpage, the first unnamed column seems to be the index column while the second unnamed column and "Country" column, both contain buttons which are unrelated to my analysis. Hence, I will leave the NaN values to be cleaned later on.

Write scraped data to csv

Since both data that has just been scraped is of university rankings, this means that information in the table will definitely be updated whenever a new year's ranking is released by THE or QS. And it is hard to say whether future updates made to the webpage will cause the webscraping process to fail due to change in html class names or tags.

Therefore, I will write the current two dataframes into csv files so that even if the web scraping fails when the source webpage gets updated in the future, it will not affect my analysis.

In [6]:
```
# write to the path (dataset > scraped)
qs_df.to_csv('dataset\scraped\qs_ranking_scraped.csv', index=False)
the_df.to_csv('dataset\scraped\\the_ranking_scraped.csv', index=False)
```

Cleaning and Processing of data

Before getting to the analysis, cleaning and processing will be done for all datasets so that they will be easy to work with later on. Since both the QS ranking and THE ranking datasets are scraped from the same website, they both have a similar structure. Therefore, I clean both using similar methods.

Cleaning of University Ranking data

In [7]:
```
# load the two csv files created earlier
qs_df = pd.read_csv('dataset/scraped/qs_ranking_scraped.csv')
the_df = pd.read_csv('dataset/scraped/the_ranking_scraped.csv')
```

In [8]:
```
# show summary of dataframe for QS ranking
qs_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47 entries, 0 to 46
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      47 non-null     int64  
 1   World Rank▲ ▼   47 non-null     object 
 2   Institution▲ ▼  47 non-null     object 
 3   Country▲ ▼      0 non-null      float64
 4   Year ▲ ▼        47 non-null     int64  
 5   Unnamed: 5      0 non-null      float64
dtypes: float64(2), int64(2), object(2)
memory usage: 2.3+ KB
```

In [9]:
```
# show summary of dataframe for THE ranking
the_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      26 non-null     int64  
 1   World Rank▲ ▼   26 non-null     object 
 2   Institution▲ ▼  26 non-null     object 
 3   Country▲ ▼      0 non-null      float64
 4   Year ▲ ▼        26 non-null     int64  
 5   Unnamed: 5      0 non-null      float64
dtypes: float64(2), int64(2), object(2)
memory usage: 1.3+ KB
```

Observing the infomation of both dataframes printed above, the following issues can be spotted:

- The columns "Country" and "Unnamed: 5" both has 0 non null values, which means the column only contains NaN values.
- The column headers and some rows contain non-alphanumeric characters.
- There are two unnamed columns.

Cleaning will now be carried out to solve the issues listed.

In [10]:
```
# drop the first unnamed column
qs_df.drop(columns=qs_df.columns[0], axis=1, inplace=True)
the_df.drop(columns=the_df.columns[0], axis=1, inplace=True)

# drop columns containing all NaN rows
qs_df.dropna(axis=1, how='all', inplace=True)
the_df.dropna(axis=1, how='all', inplace=True)

# rename column headers
col = ['Rank', 'University', 'Year']
qs_df.columns = col
the_df.columns = col

print('QS: ')
display(qs_df.head())
print('THE: ')
display(the_df.head())
```

```
QS:
```

	Rank	University	Year
0	11 ▲	Nanyang Technological University	2020
1	11 ▲	Nanyang Technological University	2018
2	11 =	National University of Singapore	2023
3	11 =	National University of Singapore	2022
4	11 =	National University of Singapore	2021

```
THE:
```

	Rank	University	Year
0	19 ▲	National University of Singapore	2023
1	21 ▲	National University of Singapore	2022
2	22 ▲	National University of Singapore	2018
3	23 ▼	National University of Singapore	2019
4	24 ▲	National University of Singapore	2017

Rearrange dataframe rows and columns

In [11]:
```
# function to replace university names with synonym
def use_acronym(df, colName):
    uniDict = { 'National University of Singapore' : 'NUS',
                'Nanyang Technological University' : 'NTU',
                'Singapore Management University' : 'SMU',
                'Singapore Institute of Technology' : 'SIT',
                'Singapore University of Technology and Design' : 'SUTD',
                'Singapore University of Social Sciences' : 'SUSS' }

    df = df.replace({ colName : uniDict })
    return df
```

In [12]:
```
# re-arrange the columns in the order -> Year, University, Rank
arranged = ['Year', 'University', 'Rank']
qs_df = qs_df.reindex(columns=arranged)
the_df = the_df.reindex(columns=arranged)

# replace all non alphanumeric characters, non-spaces, and non-dashes with whitespace
exp = '[^a-zA-Z0-9\-\ ]'
qs_df = qs_df.replace(exp, '', regex=True)
the_df = the_df.replace(exp, '', regex=True)

qs_df = use_acronym(qs_df, 'University')
the_df = use_acronym(the_df, 'University')

print('QS: ')
display(qs_df.head())
print('THE: ')
display(the_df.head())
```

QS:

	Year	University	Rank
0	2020	NTU	11
1	2018	NTU	11
2	2023	NUS	11
3	2022	NUS	11
4	2021	NUS	11

THE:

	Year	University	Rank
0	2023	NUS	19
1	2022	NUS	21
2	2018	NUS	22
3	2019	NUS	23
4	2017	NUS	24

From the output of 'qs_df' dataframe above, it can be observed that SMU's "Rank" column contain rows with hypens to indicate a range of rank. According to the FAQ page on QS Universities' website **[7]**, it is stated that universities ranked lower than a certain rank will be gouped into a range of ranking instead of being given a single rank.

Since the rank of SMU can be seen to be lower compared to NUS and NTU, it will not affect the result of my analysis no matter which number is used within the stated range. Hence, I will modify the "Rank" column such that rows containing a hypen will have its values replaced by only the first three characters. This will prevent issues from arising when sorting the rows later on.

In [13]:
```
# filter out the rows where its 'Rank' value contains a hyphen
cond = qs_df['Rank'].str.contains('-')
display(qs_df[cond])

# modify those rows by setting its new value to only the first 3 characters
qs_df.loc[cond, 'Rank'] = qs_df.loc[cond, 'Rank'].str[:3]
display(qs_df[cond])
```

	Year	University	Rank
40	2017	SMU	431-440
41	2018	SMU	441-450
44	2022	SMU	511-520
45	2021	SMU	511-520
46	2023	SMU	561-570

	Year	University	Rank
40	2017	SMU	431
41	2018	SMU	441
44	2022	SMU	511
45	2021	SMU	511
46	2023	SMU	561

As mentioned in the scope, I will only be focusing on the years from 2015 to 2021. Hence, rows where the year is out of this range will be dropped.

In [14]:
```
# only years between 2015(inclusive) and 2021(inclusive)
cond_qs = (qs_df['Year'] >= 2015) & (qs_df['Year'] <= 2021)
cond_the = (the_df['Year'] >= 2015) & (the_df['Year'] <= 2021)

qs_df = qs_df[cond_qs]
the_df = the_df[cond_the]

# sort by 'year' column followed by 'rank' column
qs_df.sort_values(by=['Year', 'Rank'], ascending=True, inplace=True)
the_df.sort_values(by=['Year', 'Rank'], ascending=True, inplace=True)

# display unique years to check
print('QS: ', qs_df['Year'].unique())
print('THE: ', the_df['Year'].unique())
```

```
QS:  [2015 2016 2017 2018 2019 2020 2021]
THE:  [2015 2016 2017 2018 2019 2020 2021]
```

Cleaning of Graduate Employment Survey data

One last dataset required for my analysis is the "graduate-employment-survey.csv" which I had exported from *Data.gov.sg*. and saved under the dataset folder.

In [15]:
```
# load GES dataset
# text encoding in 'latin-1' so that characters such as 'ä' in 'universität münchen'
ges_df = pd.read_csv('dataset/graduate-employment-survey.csv', encoding='latin-1')
ges_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1121 entries, 0 to 1120
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   year                       1121 non-null   int64 
 1   university                 1121 non-null   object
 2   school                     1121 non-null   object
 3   degree                     1121 non-null   object
 4   employment_rate_overall    1121 non-null   object
 5   employment_rate_ft_perm    1121 non-null   object
 6   basic_monthly_mean         1121 non-null   object
 7   basic_monthly_median       1121 non-null   object
 8   gross_monthly_mean         1121 non-null   object
 9   gross_monthly_median       1121 non-null   object
 10  gross_mthly_25_percentile  1121 non-null   object
 11  gross_mthly_75_percentile  1121 non-null   object
dtypes: int64(1), object(11)
memory usage: 105.2+ KB
```

First, I will drop columns that will not be used in my analysis, followed by renaming columns. When exporting, all spaces were replaced with underscores, so I wil change them back to spaces. Afterwhich I will replace all 'na' in school column with NaN, and call the use_acronym() function defined earlier, to replace all university names to their acronym.

In [16]:

```
# drop columns not required
drop_col = ['employment_rate_overall', 'gross_monthly_mean', 'gross_monthly_median']
ges_df.drop(columns=drop_col, axis=1, inplace=True)

# rename column
ges_df.rename(columns={ 'employment_rate_ft_perm': 'employment rate' }, inplace=True)

# replace all underscores with space
ges_df.columns = ges_df.columns.str.replace('_', ' ')

# replace NaN with na
ges_df['school'] = ges_df['school'].replace('na', np.NaN)

# rename university names to their acronym
ges_df = use_acronym(ges_df, 'university')

ges_df.head()
```

Out[16]:

	year	university	school	degree	employment rate	basic monthly mean	basic monthly median	gross mthly 25 percentile	gross mthly 75 percentile
0	2013	NTU	College of Business (Nanyang Business School)	Accountancy and Business	96.1	3701	3200	2900	4000
1	2013	NTU	College of Business (Nanyang Business School)	Accountancy (3-yr direct Honours Programme)	95.7	2850	2700	2700	2900
2	2013	NTU	College of Business (Nanyang Business School)	Business (3-yr direct Honours Programme)	85.7	3053	3000	2700	3500
3	2013	NTU	College of Business (Nanyang Business School)	Business and Computing	87.5	3557	3400	3000	4100
4	2013	NTU	College of Engineering	Aerospace Engineering	95.3	3494	3500	3100	3816

In [17]:
```
# drop rows not within the year 2015(inclusive) and 2021(inclusive)
years = (ges_df['year'] >= 2015) & (ges_df['year'] <= 2021)
ges_df = ges_df[years]
print(ges_df['year'].unique())
```

```
[2015 2016 2017 2018 2019 2020 2021]
```

In [18]:
```
# print before and after to verify
print('Before drop:', ges_df.shape)

# drop rows where the university is not NUS, NTU or SMU.
todrop = ges_df[~ges_df['university'].isin(['NUS', 'NTU', 'SMU'])]
ges_df = ges_df.drop(todrop.index)

print('After drop:', ges_df.shape)
print('Expected no. of rows: ', 927-275)
```

```
Before drop: (927, 9)
After drop: (652, 9)
Expected no. of rows:  652
```

In [19]:
```
# show rows where employment rate is 'na'
na_rows = ges_df['employment rate'] == 'na'
ges_df.loc[na_rows].head()
```

Out[19]:

	year	university	school	degree	employment rate	basic monthly mean	basic monthly median	gross mthly 25 percentile	gross mthly 75 percentile
199	2015	NTU	College of Engineering	Aerospace Engineering and Economics **	na	na	na	na	na
201	2015	NTU	College of Engineering	Business and Computer Engineering **	na	na	na	na	na
203	2015	NTU	College of Engineering	Chemical And Biomolecular Engineering and Econ...	na	na	na	na	na
209	2015	NTU	College of Engineering	Environmental Engineering and Economics **	na	na	na	na	na
211	2015	NTU	College of Engineering	Information Engineering And Media and Economic...	na	na	na	na	na

It can be seen from the output above that if 'employment_rate_ft_perm' is 'na', the other columns will also be 'na'. Hence, I will drop those rows as they do not contain the main information required for my analysis.

In [20]:
```
print('Before drop:', ges_df.shape)

# drop the rows
ges_df.drop(ges_df.loc[na_rows].index, inplace=True)
print('After drop:', ges_df.shape)
```

```
Before drop: (652, 9)
After drop: (593, 9)
```

Now that all 'na' values have been dealt with, the next step will be to convert the datatype of each column to a suitable type such that I am able to use it as I intend to later on (e.g. perform mathematical calculations). To do this, I will attempt to convert the values in all columns into numeric datatype (int, float). I will also set the error parameter to 'ignore' so that values that are incompatible for conversion will retain their original datatype.

In [21]:
```
ges_df = ges_df.apply(pd.to_numeric, errors='ignore')
ges_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 593 entries, 194 to 1108
Data columns (total 9 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   year                       593 non-null    int64
 1   university                 593 non-null    object
 2   school                     593 non-null    object
 3   degree                     593 non-null    object
 4   employment rate            593 non-null    float64
 5   basic monthly mean         593 non-null    int64
 6   basic monthly median       593 non-null    int64
 7   gross mthly 25 percentile  593 non-null    int64
 8   gross mthly 75 percentile  593 non-null    int64
dtypes: float64(1), int64(5), object(3)
memory usage: 46.3+ KB
```

Together with the dataset, I have also exported a .txt file containing a description of the dataset and other additional information, which can be found in the dataset folder. Based on information in this file,

- Each SMU degree has two rows, one containing respondants who achieved 'Cum Laude and Above' only, and one overall regardless of award. I will keep the overall result and drop the other.
- Degrees containing '^' in its name means that data is from a small sample size. I will drop these rows as it might cause inaccuracy in my analysis results.

In [22]:
```
# drop rows with '^' and '*'
ges_df = ges_df.drop(ges_df[ges_df['degree'].str.contains('\^')].index)
ges_df = ges_df.drop(ges_df[ges_df['degree'].str.contains('Cum Laude and above')].ind
ges_df.head()
```

Out[22]:

	year	university	school	degree	employment rate	basic monthly mean	basic monthly median	gross mthly 25 percentile	gross mthly 75 percentile
194	2015	NTU	College of Business (Nanyang Business School)	Accountancy and Business	97.3	4225	3500	3000	4850
195	2015	NTU	College of Business (Nanyang Business School)	Accountancy (3-yr direct Honours Programme)	96.5	3182	2850	2850	3125
196	2015	NTU	College of Business (Nanyang Business School)	Business (3-yr direct Honours Programme)	87.6	3343	3100	2900	3700
197	2015	NTU	College of Business (Nanyang Business School)	Business and Computing	100.0	4036	4184	3800	4876
198	2015	NTU	College of Engineering	Aerospace Engineering	86.0	3699	3650	3450	4000

Exploratory data analysis

Determine Ranking Baseline

Before getting to my first hypothesis, I will first have to determine a ranking baseline by comparing the ranking results of all six universities in QS ranking and THE ranking. This ranking baseline will be the ranking I will use to test both my hypothesis later on.

In order to easily make comparison, a function is defined to add a new national rank column that is derived from the world ranking of each university for each unique year. This function takes in a dataframe that should already be arranged by year followed by world rank in ascending order, and the desired column name for the new column.

In [23]:
```
# function to derive "National rank" column
def get_nat_rank(df):
    rankList = []
    for year in df['Year'].unique():
        num = 1
        cond = df[df['Year'] == year]['University']

        for uni in cond:
            rankList.append(num)
            num += 1

    df['National Rank'] = rankList
    return df
```

In [24]:
```
# use function above to add 'national rank' column
qs_df = get_nat_rank(qs_df)
qs_df['Ranking System'] = 'QS'

the_df = get_nat_rank(the_df)
the_df['Ranking System'] = 'THE'

display(qs_df.head())
display(the_df.head())
```

	Year	University	Rank	National Rank	Ranking System
18	2015	NUS	22	1	QS
28	2015	NTU	39	2	QS
10	2016	NUS	12	1	QS
13	2016	NTU	13	2	QS
9	2017	NUS	12	1	QS

	Year	University	Rank	National Rank	Ranking System
7	2015	NUS	25	1	THE
21	2015	NTU	61	2	THE
8	2016	NUS	26	1	THE
20	2016	NTU	55	2	THE
4	2017	NUS	24	1	THE

In [25]:
```
# display frequency table to see which universities are ranked each year
display(pd.crosstab(qs_df['Year'], qs_df['University']))
```

```
display(pd.crosstab(the_df['Year'], the_df['University']))
```

University	NTU	NUS	SMU
Year			
2015	1	1	0
2016	1	1	0
2017	1	1	1
2018	1	1	1
2019	1	1	1
2020	1	1	1
2021	1	1	1

University	NTU	NUS
Year		
2015	1	1
2016	1	1
2017	1	1
2018	1	1
2019	1	1
2020	1	1
2021	1	1

Based on the frequency table above:

- NTU and NUS are ranked throughout 2015 to 2021 for both QS and THE
- SMU is ranked from 2017 to 2021 in QS, and not ranked on THE

In order to better visialise and compare the national rankings, I will create a new dataframe by concatenating 'qs_df2' and 'the_df2', before sorting the new dataframe by Year followed by National Rank.

In [26]:
```
# combine both rankings into one dataframe and sort
qs_df2 = qs_df.drop(['Rank'], axis=1)
the_df2 = the_df.drop(['Rank'], axis=1)
comb_df = pd.concat([qs_df2, the_df2])
comb_df.sort_values(by=['Year', 'National Rank'], ascending=True, inplace=True)
comb_df.head()
```

Out[26]:

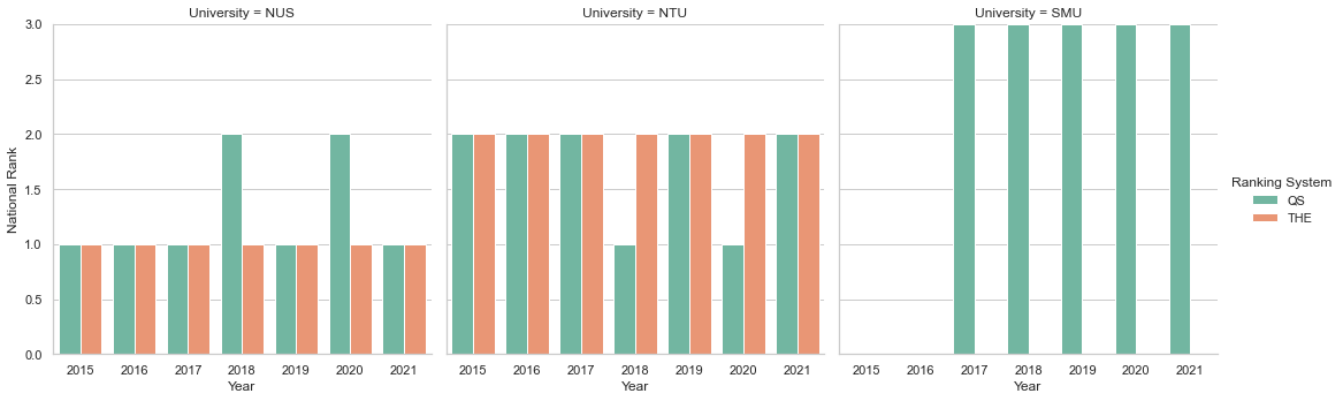
	Year	University	National Rank	Ranking System
18	2015	NUS	1	QS
7	2015	NUS	1	THE
28	2015	NTU	2	QS
21	2015	NTU	2	THE
10	2016	NUS	1	QS

Plot graph to cross-compare rankings

Using the new dataframe created, I used seaborn catplot function to plot three bar graphs. Each

plot represents the national ranking of QS and THE, for each unique university across the years 2015 to 2021.

In [27]:
```
# plot bar graph
sns.set_theme(style="whitegrid")
ax = sns.catplot(data=comb_df, kind="bar", palette="Set2",
                 x="Year", y="National Rank",
                 hue="Ranking System", col="University")

ax.set(ylim=(0, 3))

# label x and y-axis
ax.set(xlabel="Year")
ax.set(ylabel="National Rank");
```



Final Ranking Baseline

The following can be observed form the output above:

- NUS ranked first nationally (both QS and THE rank) in most years except 2018 and 2020.
- NTU ranked second nationally (both QS and THE rank) in most years except 2018 and 2020.
- NUS ranked second nationally in QS, and first in THE in the years 2018 and 2020.
- NTU ranked first nationally in QS, and second in THE in the years 2018 and 2020.
- SMU ranked third nationally from 2017 to 2021 based on QS rank.

In order to resolve the discrepancy in results for NUS and NTU in 2018 and 2020, I will look at one other ranking system, Academic Ranking of World Universities (ARWU). Based on ARWU ranking results, NUS ranked first regionally while NTU ranked second in both 2018 **[8]** and 2020 **[9]**. This aligns with the results of THE ranking based on the graph.

As such, the ranking baseline that will be used to test my hypothesis will be:

1. NUS
2. NTU
3. SMU

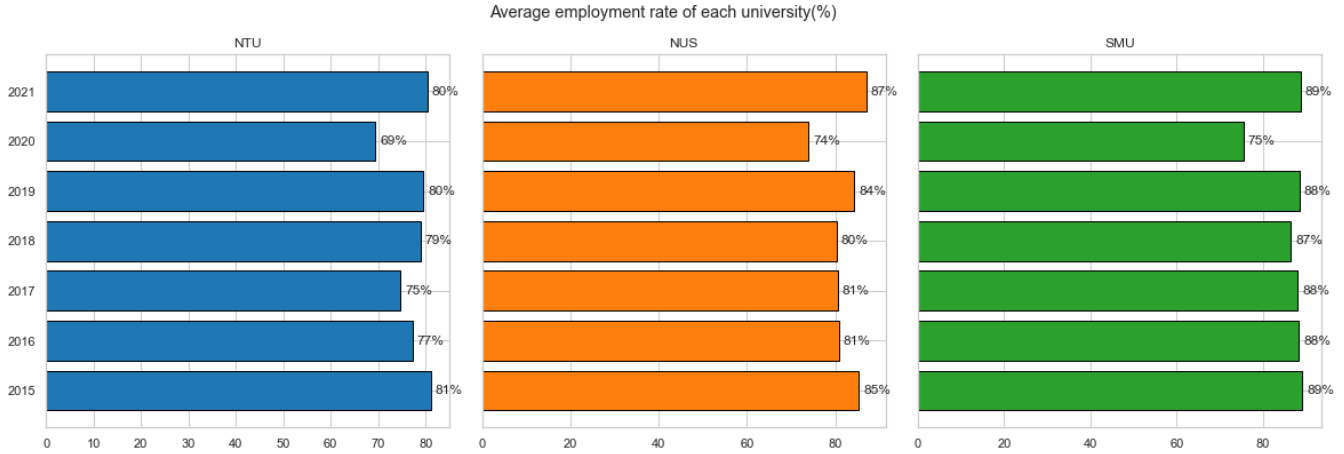
Analysis (Hypothesis 1)

With the ranking baseline determined, I will now test my first hypothesis.
Hypothesis 1: There is a correlation between university rankings and employment outcome.

Employment rate

Plotting of a horizontal bar graph to show the average overall employment rate of each university, for each year.

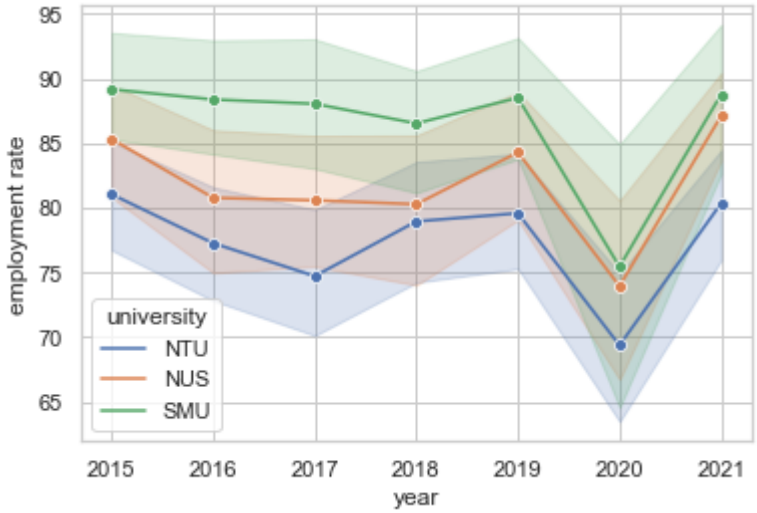
In [28]:
```
fig, ax = plt.subplots(ncols=3, figsize=(20,6), sharey=True)

# set color iterator
palette = itertools.cycle(sns.color_palette(palette='tab10'))

# loop through each uni and plot graph
loop = 0

# plot graph
for uni in ges_df['university'].unique():
    # group by year and find the mean of employment rate
    uni_df = ges_df[ges_df['university'] == uni]
    uni_avg = uni_df.groupby('year').mean()['employment rate'].sort_values()

    # plot horizontal bar graph
    bar = ax[loop].barh(uni_avg.index, uni_avg, color=next(palette), edgecolor='black

    # set label and title
    ax[loop].bar_label(bar, fmt='%.0f%%', padding=4)
    ax[loop].set_title(uni)

    loop+=1

# set main figure title
fig.suptitle('Average employment rate of each university(%)')
plt.subplots_adjust(wspace=0.08)
plt.show()
```



Based on the linegraph above, it can be observed that throughout the years 2015 to 2021, SMU showed the highest number of average employment rate, followed by NUS, and NTU. It can also be seen that the employment rate of all three universitie are generally between the range of 70% to 90%.

Plotting of linegraph to better show the difference in employment rate over the years

In [29]:
```
# plot linegraph
sns.lineplot(data=ges_df, x='year', y='employment rate',
             hue='university', marker='o');
```



Based on the linegraph above, it can be seen clearly that throughout the years 2015 to 2021, SMU showed the highest number of average employment rate, followed by NUS, and NTU. It can also been seen that all three universities show a similar pattern where in 2020 there is a large dip.

In [30]:
```
# print number of employment rate records available for each uni in each year
ntu_df = ges_df[ges_df['university'] == 'NTU']
nus_df = ges_df[ges_df['university'] == 'NUS']
smu_df = ges_df[ges_df['university'] == 'SMU']

for year in ges_df['year'].unique():
    nus = nus_df[nus_df['year'] == year]['employment rate']
    ntu = ntu_df[ntu_df['year'] == year]['employment rate']
    smu = smu_df[smu_df['year'] == year]['employment rate']
    print(year,
          '- NUS:', nus.count(),
          ' | NTU:', ntu.count(),
          ' | SMU:', smu.count())
```

```
2015 - NUS: 36  | NTU: 33  | SMU: 6
2016 - NUS: 34  | NTU: 34  | SMU: 6
2017 - NUS: 37  | NTU: 35  | SMU: 6
2018 - NUS: 31  | NTU: 34  | SMU: 6
2019 - NUS: 31  | NTU: 34  | SMU: 6
2020 - NUS: 31  | NTU: 36  | SMU: 6
2021 - NUS: 33  | NTU: 34  | SMU: 6
```

Observation (Based on employment rate)

To summarise, in both the line and bar graphs above, SMU showed the highest number of average employment rate, followed by NUS, and NTU across all years. Relating back to the ranking baseline determined earlier, SMU was ranked third, yet achieved the highest average employment rate. On the other hand, NUS and NTU which were ranked first and second respectively has the second and third highest employment rate respectively.

However, in both graphs, the average employment rate was calculated by taking the mean value of the 'employment rate' column after grouping by year and university. This means that the number of records used in the calculation is an important aspect that would affect the accuracy of the results.

Referring at the output of the cell directly above, it can be seen that throughout years 2015 to 2021, NTU and NUS have shown a similar number of records ranging from 31 to 36 that is available, while SMU has a significantly lower number of 6 records. (note: 6 records means that overall employment rate of only 6 different degrees in SMU are available for each year). As such, the mean employment rate calculated and used to plot the graphs above might not be as accurate. Hence, from this point onwards, I will be excluding SMU from my analysis. I will only compare between NUS and NTU since both show a consistent number of records available.

Conclusion

Based on the overall employment rate across the years, it can be observed that fresh graduates from the higher ranked university(NUS) has indeed had a higher employment rate throughout the years 2015 to 2021, as compared to NTU which is ranked lower and showed a lower employment rate. This supports my first hypothesis that there is indeed a correlation between ranking and employment outcome.

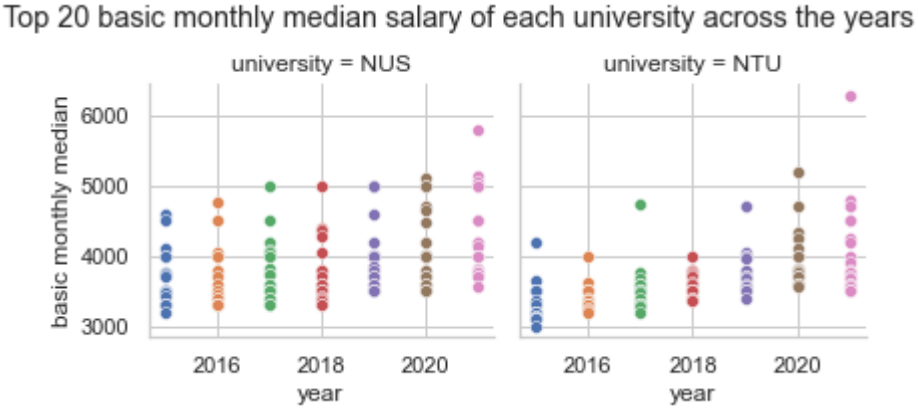
In [31]:
```
# update dataframe to remove SMU
ges_df = ges_df[ges_df['university'] != 'SMU']
ges_df['university'].unique()
```

Out[31]:
```
array(['NTU', 'NUS'], dtype=object)
```

Basic monthly median salary

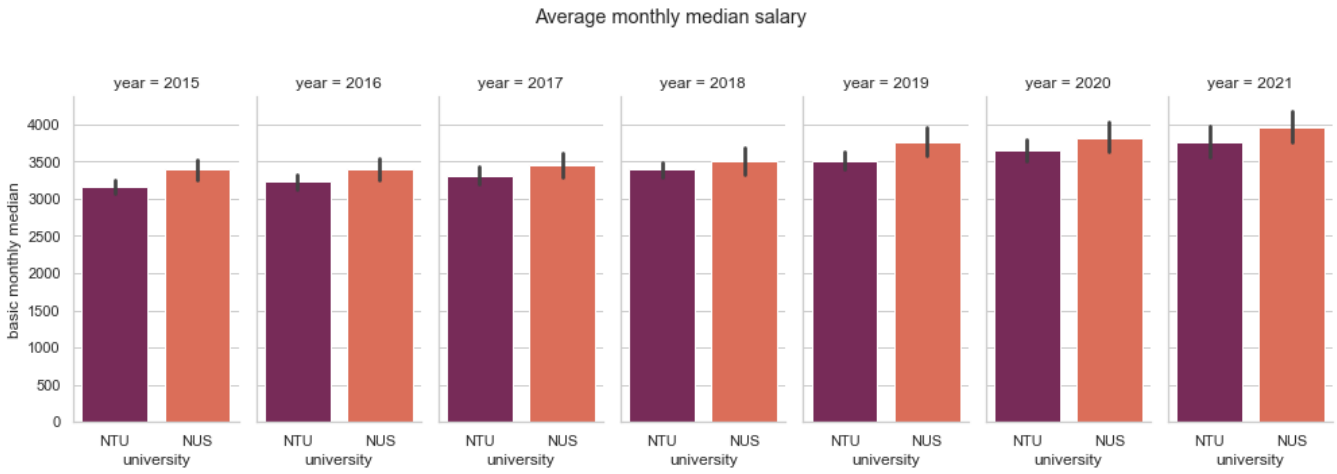
Moving on, I will now plotting a scatter plot to show the top 20 degrees of each university with the highest basic monthly median salary for each year from 2015 to 2021.

In [32]:
```
# group by year, university and get the top 20 monthly median
scatter_df = ges_df.sort_values(['year', 'basic monthly median'],
                                ascending=[True, False]).groupby(['year', 'universit

# plot scatter
g = sns.FacetGrid(scatter_df, col='university', hue='year')
g.map(sns.scatterplot, 'year', 'basic monthly median')

g.fig.subplots_adjust(top=.8)
g.fig.suptitle('Top 20 basic monthly median salary of each university across the year
plt.show()
```



Based on the scatter plot above, fresh graduates from NUS generally showed a higher basic monthly median salary as compared to fresh graduated from NTU. To take a closer look at the difference, I will plot several bar graph to compare the top 3 degrees with the higest basic monthly median salary for each university from 2015 to 2021.

In [33]:
```
g = sns.FacetGrid(ges_df, col='year', height=5, aspect=.4)
g.map(sns.barplot, 'university', 'basic monthly median', palette='rocket', order=['NT
g.fig.subplots_adjust(top=0.8)
g.fig.suptitle('Average monthly median salary');
```



Observation (Based on monthly median salary)

Based on both the scatter and bar charts above, it can be observed that fresh graduates from the higher ranking university, NUS, showed higher basic monthly median salary across all years from 2015 to 2021 as compared to fresh graduates from NTU. This supports my first hypothesis that there is indeed a correlation between university ranking and employment outcome such that fresh graduates from a higher ranked university earns a higher basic monthly median salary.

Analysis (Hypothesis 2)

With the first hypothesis shown to be true, I will now test my second hypothesis.
Hypothesis 2: It is true that university rankings no longer affects employment outcomes of local university fresh graduates as much as it did years ago.

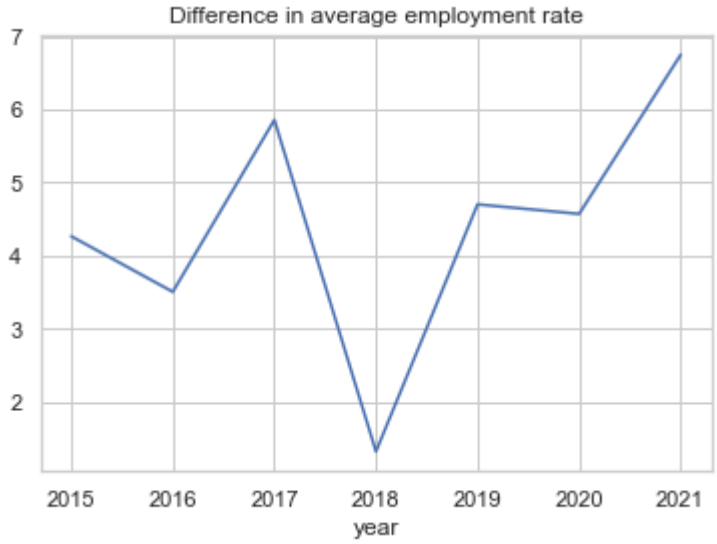
Change in difference in employment rate

Calculate and plot the difference in average employment rate between NUS and NTU across the years, so that it can be used to visualise the trend.

In [34]:
```
# calculate the difference in employment rate each year
empRateDiff = (nus_df.groupby(['year'])['employment rate'].mean()
- ntu_df.groupby(['year'])['employment rate'].mean())
empRateDiff
```

Out[34]:
```
year
2015    4.265909
2016    3.505882
2017    5.855521
2018    1.324194
2019    4.701233
2020    4.571057
2021    6.746791
Name: employment rate, dtype: float64
```

In [35]:
```
# plot the difference
ax = empRateDiff.plot(x='year', title='Difference in average employment rate')
```



Observation (based on difference in overall employment rate)

Based on the line graph above, it is difficult to say whether there was an increase or decrese in the difference in employment rate, since the lines are staggered which means there is no consistency in the data. However, it can be seen that the difference shown in the years after 2019 have generally been higher than in yeat 2015. This does not support my hypothesis since there is an even bigger difference in employment rate between NUS and NTU in recent years, as compared to in 2015.

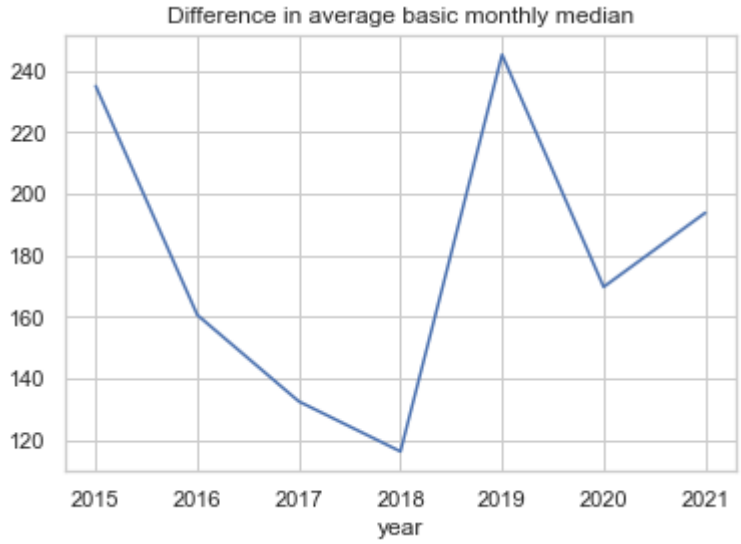
Change in difference in monthly median salary

Calculate and plot the difference in basic monthly median between NUS and NTU across the years, so that it can be used to visualise the trend.

In [36]:
```
# calculate the difference in basic monthly median each year
medianDiff = (nus_df.groupby(['year'])['basic monthly median'].mean()
- ntu_df.groupby(['year'])['basic monthly median'].mean())
medianDiff
```

Out[36]:
```
year
2015    235.116162
2016    160.588235
2017    132.429344
2018    116.151803
2019    245.303605
2020    169.716846
2021    193.836007
Name: basic monthly median, dtype: float64
```

In [37]:
```
# plot the difference
ax = medianDiff.plot(x='year', title='Difference in average basic monthly median')
```



Observation(Based on change in basic monthly median)

Similar to the previous line graph, this like graph also does not show a consistent trend. However, it can be observed that from the years 2016 to 2018, and 2020 to 2021, the difference was lower than in year 2015. This support my hypothesis since it means that the fresh graduates from NUS and NTU are starting to earn a basic monthly median salary more similar to each other even as the ranking of the respective university remains the same.

Conclusion

To summarise the observation of my analysis, I first found a ranking baseline based by comparing between two ranking systems, and also cross cheked with another due to descrepancies in ranking between the first two. The final ranking baseline was Rank 1: NUS, Rank 2: NTU, Rank 3: SMU.

Next, I tested my first hypothesis (that there is a correlation between university rankings and employment outcome) which according to my analysis showed to be true. Graduates from the higher ranked university, NUS, showed a higher employment rate and monthly median salary across all years from 2015 to 2021. As compared to graduates from NTU. It was also found that there was a lack of SMU data, therefore, SMU was excluded from the analysis.

The second hypothesis (that university rankings no longer affects employment outcomes of local university fresh graduates as much as it did years ago) however turned out to be inconclusive as there was no concrete trend shown when observing the change in the difference in employment rate and monthly median salary between NUS and NTU for each year.

Ideas for further development

This analysis can be futher improved by comparing between even more universities instead of just basing the scope on Singapore's local universities. By broaderning the scope, more interesting analysis can be made by comparing between different countries and looking at the difference shown by each country or region.

References

Data Sources

- https://www.universityrankings.ch/results?ranking=QS&region=World&year=all+years&q=Singapore
- https://www.universityrankings.ch/results?ranking=Times&region=World&year=all+years&q=Singapore
- https://data.gov.sg/dataset/graduate-employment-survey-ntu-nus-sit-smu-suss-sutd

Ranking Systems

[1] THE - Times Higher Education. “World University Rankings.” Times Higher Education (THE), 29-Nov-2022. [Online]. Available: https://www.timeshighereducation.com/world-university-rankings.

[2] QS Quacquarelli Symonds Limited. “QS universities rankings.” Top Universities. [Online]. Available: https://www.topuniversities.com/university-rankings.

[3] ShanghaiRanking Consultancy. “Academic Ranking of World universities.” ShanghaiRanking. [Online]. Available: http://www.shanghairanking.com/.

[4] QS Quacquarelli Symonds Limited. “QS Asia University Rankings 2023 - Overall.” Top Universities. [Online]. Available: https://www.topuniversities.com/university-rankings/asia-university-rankings/2023.

Terms of use of data

[5] Swiss Government. “Terms and conditions.” Der Bundesrat admin.ch - Startseite. [Online]. Available: https://www.admin.ch/gov/en/start/terms-and-conditions.html#1938362905.

[6] Source acknowledgement and Licence: This report contains information from "Graduate Employment Survey - NTU, NUS, SIT, SMU, SUSS & SUTD" accessed on 3/12/2022 from Data.gov.sg which is made available under the terms of the Singapore Open Data Licence version 1.0

[7] S. Collier, “World University Rankings – Frequently asked questions,” Top Universities, 13-Oct-2022. [Online]. Available: https://www.topuniversities.com/university-rankings-articles/world-university-rankings/world-university-rankings-frequently-asked-questions

cross references

[8] “2018 academic ranking of world universities,” ShanghaiRanking's Academic Ranking of World Universities. [Online]. Available: https://www.shanghairanking.com/rankings/arwu/2018.

[9] “2020 Academic Ranking of World universities,” ShanghaiRanking's Academic Ranking of World Universities. [Online]. Available: https://www.shanghairanking.com/rankings/arwu/2020