Data Analysis of University Graduates in Singapore

Yadanar Aung (P2214621)
DAAA/FT/1B/02
ST1510 PDAS

About Analysis

Ms Tan is the Data Analyst for the Director of Ministry of Student Development in MOE.

She wants to analyse how **University Graduates' Population** & **Gender Ratio** have changed over the years **2010 to 2020**.

She also wants to analyse how graduates fare in the industry, such as their **gross monthly salaries**.

What is Analysed

Ms Tan decided to analyse the following:

1. The **change in number of University graduates**, for Female & Male, from the years 2010 to 2020
2. The **gender ratio per University** in 2020
3. The **change in gender ratio** per University **course** in 2010 to 2020
4. The **distribution of Median Gross Monthly Salary** for University Graduates
5. The **distribution of Median Gross Monthly Salary Per University**

Datasets Used

1. **graduate-employment-survey-ntu-nus-sit-smu-suss-sutd.csv**

https://data.gov.sg/dataset/graduate-employment-survey-ntu-nus-sit-smu-suss-sutd

2. **universities-intake-enrolment-and-graduates-by-course.csv**

https://data.gov.sg/dataset/universities-intake-enrolment-and-graduates-by-course

3. **graduates-by-institutions.csv**

https://data.gov.sg/dataset/intake-enrolment-and-graduates-by-institutions

Nature of Dataset 1

graduate-employment-survey-ntu-nus-sit-smu-suss-sutd

year	Year of Record Taken (YYYY)
university	Universities in Singapore
school	Schools in Respective Universities
degree	Degree that University School Offers
employment_rate_overall	Employment Rate for University Degree
employment_rate_ft_perm	Employment Rate For Full Time Permanent Position

basic_monthly_mean	Basic Monthly Mean Salary for University Degree
basic_monthly_median	Basic Monthly Median Salary for University Degree
gross_monthly_mean	Gross Monthly Mean Salary for University Degree
gross_monthly_median	Gross Monthly Median Salary for University Degree
gross_mthly_25_percentile	Gross Monthly Q1 Salary for University Degree
gross_mthly_75_percentile	Gross Monthly Q3 Salary for University Degree

Nature of Dataset 2

universities-intake-enrolment-and-graduates-by-course

year	Year of Record Taken (YYYY)
sex	Gender taken into account (Both, Female)
course	Classified University Courses
intake	Number of University intakes in that Year
enrolment	Number of University enrolment in that Year
graduates	Number of University graduates in that Year

Nature of Dataset 3

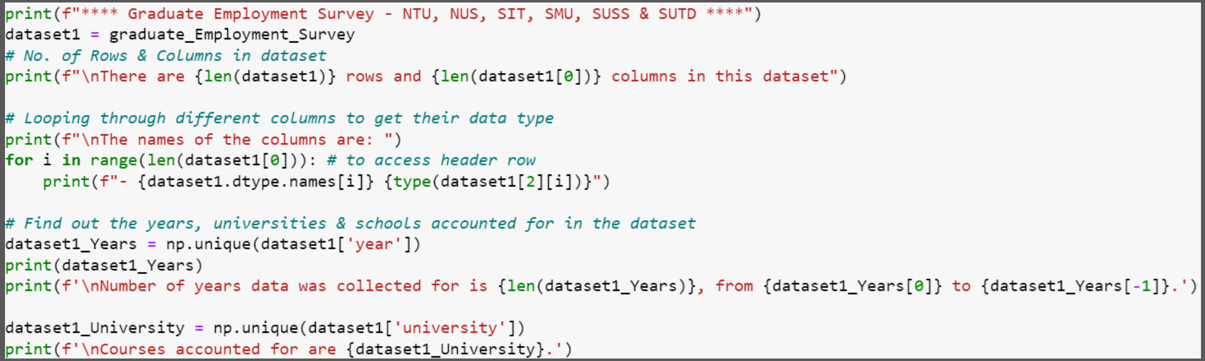
graduates-by-institutions

year	Year of Record Taken (YYYY)
sex	Gender taken into account (Both, Female)
nus	National University Singapore
ntu	Nanyang Technological University
smu	Singapore Management University
sit	Singapore Institute of Technology

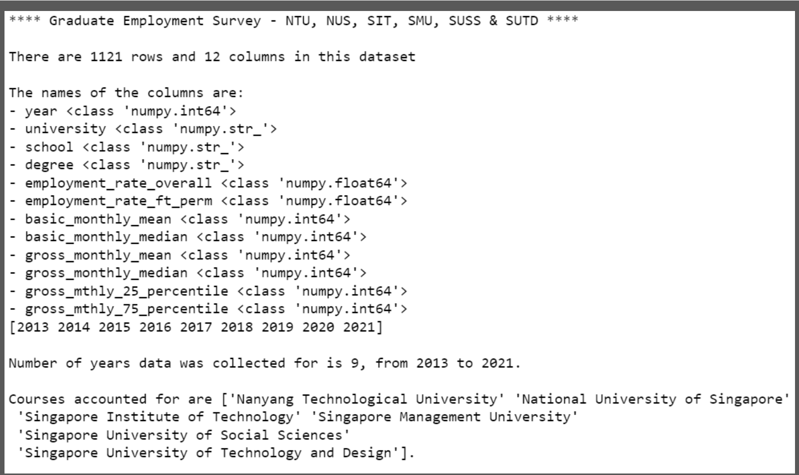
stud	Singapore University of Technology & Design
suss	Singapore University of Social Sciences
nie	National Institute of Education
singapore_polytechnic	Singapore Polytechnic
ngee_ann_polytechnic	Ngee Ann Polytechnic
temasek_polytechnic	Temasek Polytechnic
nanyang_polytechnic	Nanyang Polytechnic

republic_polytechnic	Republic Polytechnic
lasalle_diploma	Lasalle College of the Arts
lasalle_degree	Lasalle College of the Arts
nafa_diploma	Nanyang Academy of Fine Arts
nafa_degree	Nanyang Academy of Fine Arts
ite	Institute of Technical Education

Analysis of Datasets



```
print(f"**** Graduate Employment Survey - NTU, NUS, SIT, SMU, SUSS & SUTD ****")
dataset1 = graduate_Employment_Survey
# No. of Rows & Columns in dataset
print(f"\nThere are {len(dataset1)} rows and {len(dataset1[0])} columns in this dataset")

# Looping through different columns to get their data type
print(f"\nThe names of the columns are: ")
for i in range(len(dataset1[0])): # to access header row
    print(f"- {dataset1.dtype.names[i]} {type(dataset1[2][i])}")

# Find out the years, universities & schools accounted for in the dataset
dataset1_Years = np.unique(dataset1['year'])
print(dataset1_Years)
print(f'\nNumber of years data was collected for is {len(dataset1_Years)}, from {dataset1_Years[0]} to {dataset1_Years[-1]}.')

dataset1_University = np.unique(dataset1['university'])
print(f'\nCourses accounted for are {dataset1_University}.')
```

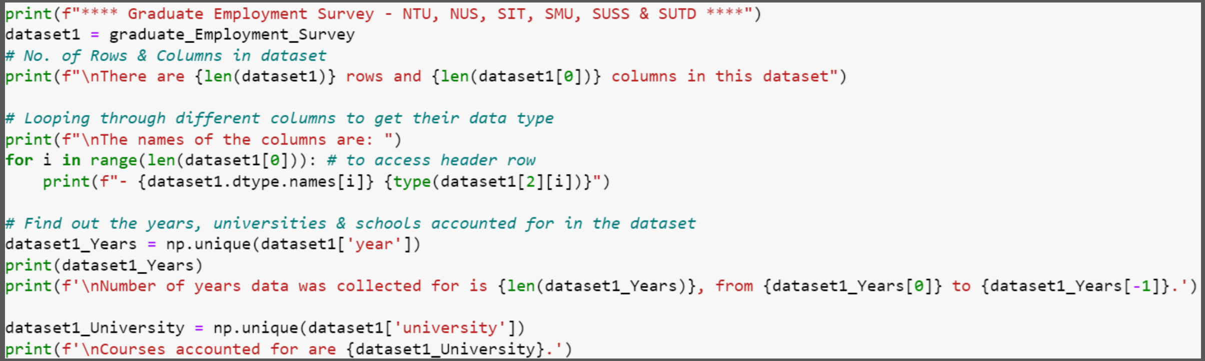


```
**** Graduate Employment Survey - NTU, NUS, SIT, SMU, SUSS & SUTD ****

There are 1121 rows and 12 columns in this dataset

The names of the columns are:
- year <class 'numpy.int64'>
- university <class 'numpy.str_'>
- school <class 'numpy.str_'>
- degree <class 'numpy.str_'>
- employment_rate_overall <class 'numpy.float64'>
- employment_rate_ft_perm <class 'numpy.float64'>
- basic_monthly_mean <class 'numpy.int64'>
- basic_monthly_median <class 'numpy.int64'>
- gross_monthly_mean <class 'numpy.int64'>
- gross_monthly_median <class 'numpy.int64'>
- gross_mthly_25_percentile <class 'numpy.int64'>
- gross_mthly_75_percentile <class 'numpy.int64'>
[2013 2014 2015 2016 2017 2018 2019 2020 2021]

Number of years data was collected for is 9, from 2013 to 2021.

Courses accounted for are ['Nanyang Technological University' 'National University of Singapore'
 'Singapore Institute of Technology' 'Singapore Management University'
 'Singapore University of Social Sciences'
 'Singapore University of Technology and Design'].
```

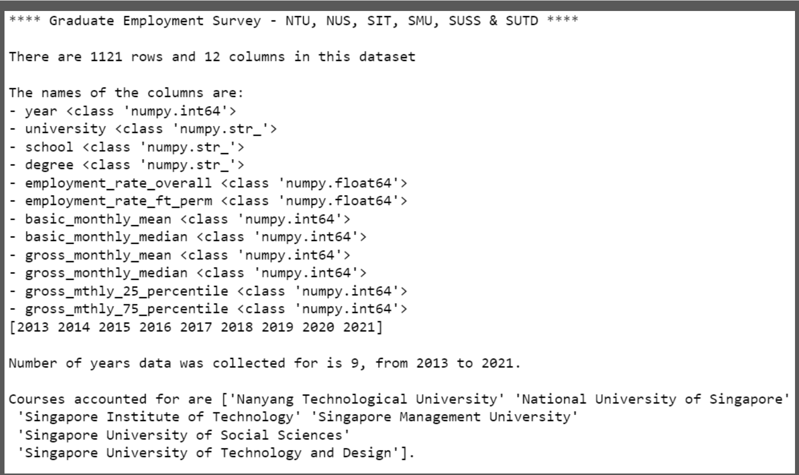
Performed Analysis to Find Out:

- No. of Rows & Columns:
  - **len()** function
- Names & Data Type of Columns:
  - For loop
  - **.dtype.name**
  - **type()**
- No. of unique values per Col:
  - **np.unique()**
    - Year
    - Sex
    - University

Analysis of Dataset 1



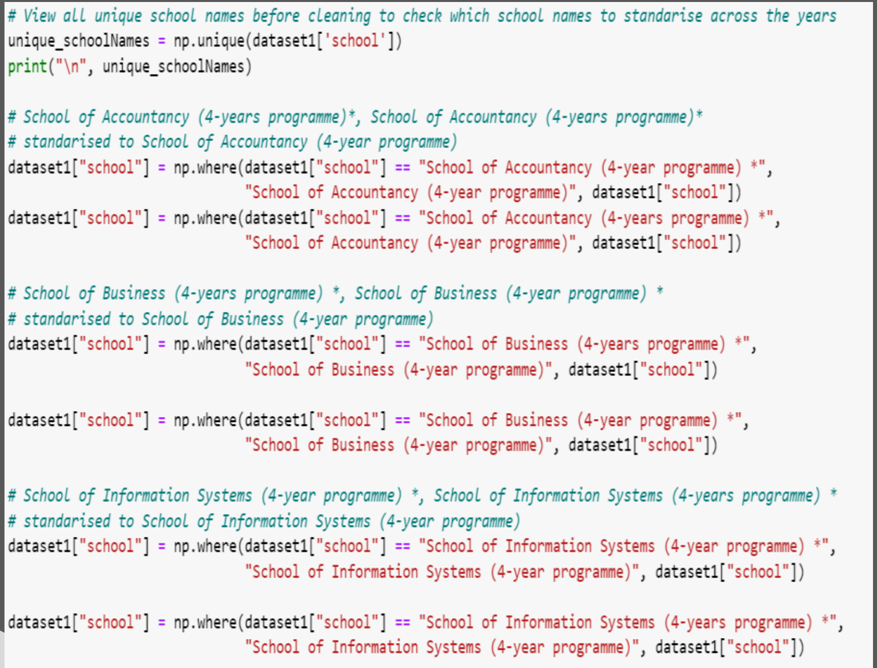
```python
print(f"**** Graduate Employment Survey - NTU, NUS, SIT, SMU, SUSS & SUTD ****")
dataset1 = graduate_Employment_Survey
# No. of Rows & Columns in dataset
print(f"\nThere are {len(dataset1)} rows and {len(dataset1[0])} columns in this dataset")

# Looping through different columns to get their data type
print(f"\nThe names of the columns are: ")
for i in range(len(dataset1[0])): # to access header row
    print(f"- {dataset1.dtype.names[i]} {type(dataset1[2][i])}")

# Find out the years, universities & schools accounted for in the dataset
dataset1_Years = np.unique(dataset1['year'])
print(dataset1_Years)
print(f'\nNumber of years data was collected for is {len(dataset1_Years)}, from {dataset1_Years[0]} to {dataset1_Years[-1]}.')

dataset1_University = np.unique(dataset1['university'])
print(f'\nCourses accounted for are {dataset1_University}.')
```



```
**** Graduate Employment Survey - NTU, NUS, SIT, SMU, SUSS & SUTD ****

There are 1121 rows and 12 columns in this dataset

The names of the columns are:
- year <class 'numpy.int64'>
- university <class 'numpy.str_'>
- school <class 'numpy.str_'>
- degree <class 'numpy.str_'>
- employment_rate_overall <class 'numpy.float64'>
- employment_rate_ft_perm <class 'numpy.float64'>
- basic_monthly_mean <class 'numpy.int64'>
- basic_monthly_median <class 'numpy.int64'>
- gross_monthly_mean <class 'numpy.int64'>
- gross_monthly_median <class 'numpy.int64'>
- gross_mthly_25_percentile <class 'numpy.int64'>
- gross_mthly_75_percentile <class 'numpy.int64'>
[2013 2014 2015 2016 2017 2018 2019 2020 2021]

Number of years data was collected for is 9, from 2013 to 2021.

Courses accounted for are ['Nanyang Technological University' 'National University of Singapore'
 'Singapore Institute of Technology' 'Singapore Management University'
 'Singapore University of Social Sciences'
 'Singapore University of Technology and Design'].
```

- Using **np.unique()**:
  - Find unique values for column:
    - School
      - Check if there is varied naming conventions

Data Cleaning (Dataset 1)



```
# View all unique school names before cleaning to check which school names to standarise across the years
unique_schoolNames = np.unique(dataset1['school'])
print("\n", unique_schoolNames)

# School of Accountancy (4-years programme)*, School of Accountancy (4-years programme)*
# standarised to School of Accountancy (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Accountancy (4-year programme) *",
                              "School of Accountancy (4-year programme)", dataset1["school"])
dataset1["school"] = np.where(dataset1["school"] == "School of Accountancy (4-years programme) *",
                              "School of Accountancy (4-year programme)", dataset1["school"])

# School of Business (4-years programme) *, School of Business (4-year programme) *
# standarised to School of Business (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Business (4-years programme) *",
                              "School of Business (4-year programme)", dataset1["school"])

dataset1["school"] = np.where(dataset1["school"] == "School of Business (4-year programme) *",
                              "School of Business (4-year programme)", dataset1["school"])

# School of Information Systems (4-year programme) *, School of Information Systems (4-years programme) *
# standarised to School of Information Systems (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Information Systems (4-year programme) *",
                              "School of Information Systems (4-year programme)", dataset1["school"])

dataset1["school"] = np.where(dataset1["school"] == "School of Information Systems (4-years programme) *",
                              "School of Information Systems (4-year programme)", dataset1["school"])
```

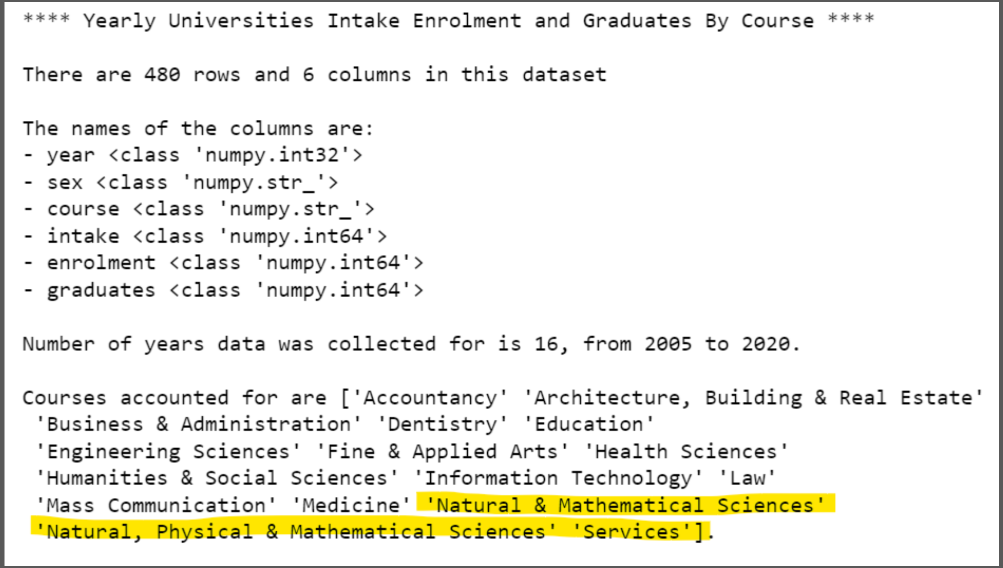
As there are varied naming conventions in the column (“school”):

- Needs to be standardised to make extraction of statistics more accurate

Use **numpy boolean indexing** & **np.where()**

Eg. Standardised “School of Accountancy (4-year programme) *” & “School of Accountancy (4-years programme) *” to “School of Accountancy (4-year programme)”

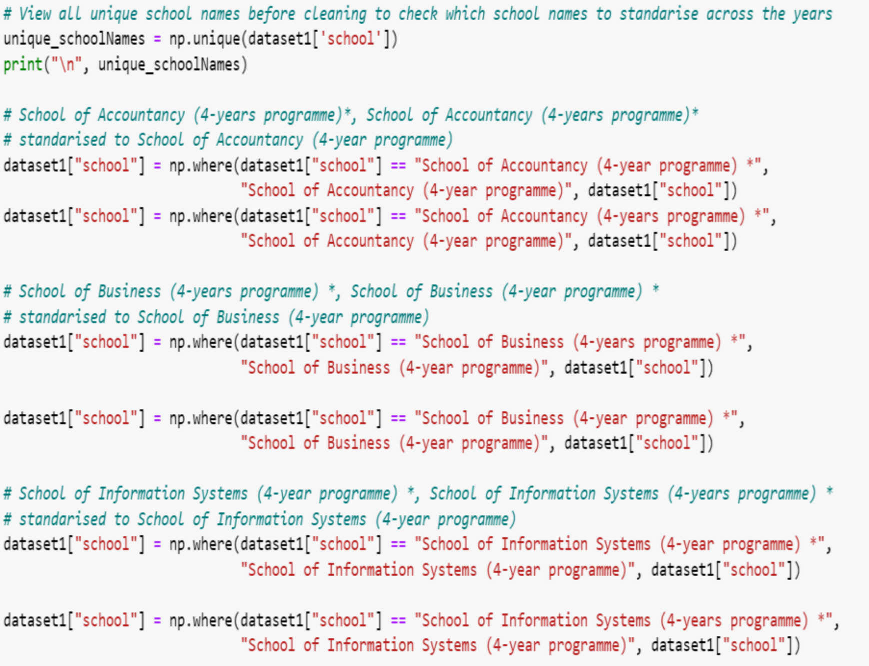
Analysis of Dataset 2



```
**** Yearly Universities Intake Enrolment and Graduates By Course ****

There are 480 rows and 6 columns in this dataset

The names of the columns are:
- year <class 'numpy.int32'>
- sex <class 'numpy.str_'>
- course <class 'numpy.str_'>
- intake <class 'numpy.int64'>
- enrolment <class 'numpy.int64'>
- graduates <class 'numpy.int64'>

Number of years data was collected for is 16, from 2005 to 2020.

Courses accounted for are ['Accountancy' 'Architecture, Building & Real Estate'
 'Business & Administration' 'Dentistry' 'Education'
 'Engineering Sciences' 'Fine & Applied Arts' 'Health Sciences'
 'Humanities & Social Sciences' 'Information Technology' 'Law'
 'Mass Communication' 'Medicine' 'Natural & Mathematical Sciences'
 'Natural, Physical & Mathematical Sciences' 'Services'].
```

- Using **np.unique()**:
  - Find unique values for column:
    - course
      - Check if there is varied naming conventions

Data Cleaning (Dataset 2)



```python
# View all unique school names before cleaning to check which school names to standarise across the years
unique_schoolNames = np.unique(dataset1['school'])
print("\n", unique_schoolNames)

# School of Accountancy (4-years programme)*, School of Accountancy (4-years programme)*
# standarised to School of Accountancy (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Accountancy (4-year programme) *",
                              "School of Accountancy (4-year programme)", dataset1["school"])
dataset1["school"] = np.where(dataset1["school"] == "School of Accountancy (4-years programme) *",
                              "School of Accountancy (4-year programme)", dataset1["school"])

# School of Business (4-years programme) *, School of Business (4-year programme) *
# standarised to School of Business (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Business (4-years programme) *",
                              "School of Business (4-year programme)", dataset1["school"])

dataset1["school"] = np.where(dataset1["school"] == "School of Business (4-year programme) *",
                              "School of Business (4-year programme)", dataset1["school"])

# School of Information Systems (4-year programme) *, School of Information Systems (4-years programme) *
# standarised to School of Information Systems (4-year programme)
dataset1["school"] = np.where(dataset1["school"] == "School of Information Systems (4-year programme) *",
                              "School of Information Systems (4-year programme)", dataset1["school"])

dataset1["school"] = np.where(dataset1["school"] == "School of Information Systems (4-years programme) *",
                              "School of Information Systems (4-year programme)", dataset1["school"])
```

As there are varied naming conventions in the column ("course"):

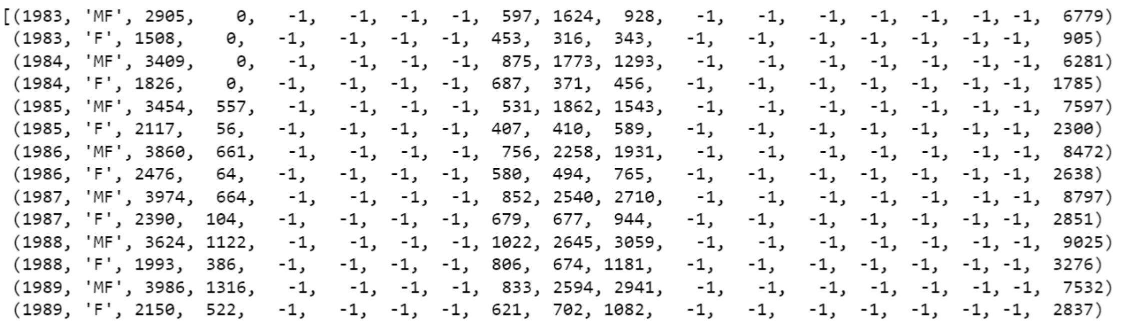
- Needs to be standardised to make extraction of statistics more accurate

Use **numpy boolean indexing** & **np.where()**

Eg. Standardised "Natural & Mathematical Sciences) *" & "Natural, Physical & Mathematical Sciences" to "Natural, Physical & Mathematical Sciences"

Analysis of Dataset 3

- Print(dataset3):
  - Values of **-1** for the column of intake, enrolment, graduates
    - Which is when certain universities where <u>not opened</u> yet in those years

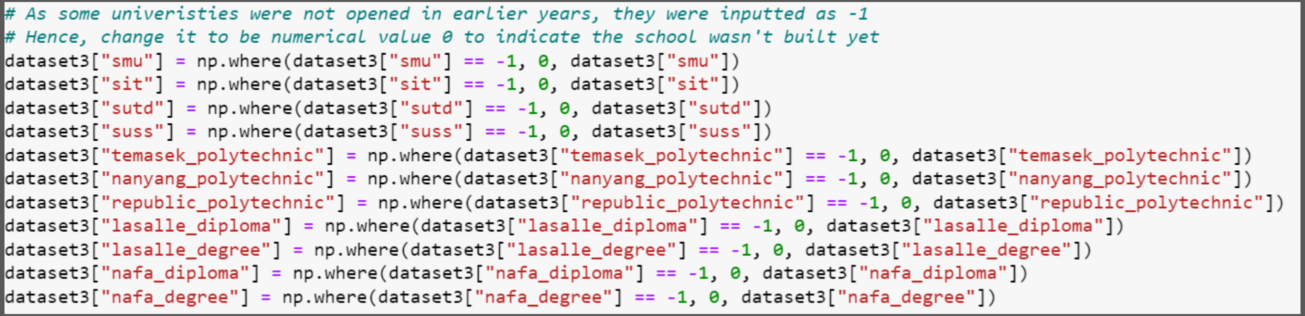


```
[(1983, 'MF', 2905,    0,   -1,   -1,  -1,  -1,  597, 1624,  928,   -1,   -1,   -1,  -1,  -1,  -1, -1,  6779)
 (1983, 'F', 1508,    0,   -1,   -1,  -1,  -1,  453,  316,  343,   -1,   -1,   -1,  -1,  -1,  -1, -1,   905)
 (1984, 'MF', 3409,    0,   -1,   -1,  -1,  -1,  875, 1773, 1293,   -1,   -1,   -1,  -1,  -1,  -1, -1,  6281)
 (1984, 'F', 1826,    0,   -1,   -1,  -1,  -1,  687,  371,  456,   -1,   -1,   -1,  -1,  -1,  -1, -1,  1785)
 (1985, 'MF', 3454,  557,   -1,   -1,  -1,  -1,  531, 1862, 1543,   -1,   -1,   -1,  -1,  -1,  -1, -1,  7597)
 (1985, 'F', 2117,   56,   -1,   -1,  -1,  -1,  407,  410,  589,   -1,   -1,   -1,  -1,  -1,  -1, -1,  2300)
 (1986, 'MF', 3860,  661,   -1,   -1,  -1,  -1,  756, 2258, 1931,   -1,   -1,   -1,  -1,  -1,  -1, -1,  8472)
 (1986, 'F', 2476,   64,   -1,   -1,  -1,  -1,  580,  494,  765,   -1,   -1,   -1,  -1,  -1,  -1, -1,  2638)
 (1987, 'MF', 3974,  664,   -1,   -1,  -1,  -1,  852, 2540, 2710,   -1,   -1,   -1,  -1,  -1,  -1, -1,  8797)
 (1987, 'F', 2390,  104,   -1,   -1,  -1,  -1,  679,  677,  944,   -1,   -1,   -1,  -1,  -1,  -1, -1,  2851)
 (1988, 'MF', 3624, 1122,   -1,   -1,  -1,  -1, 1022, 2645, 3059,   -1,   -1,   -1,  -1,  -1,  -1, -1,  9025)
 (1988, 'F', 1993,  386,   -1,   -1,  -1,  -1,  806,  674, 1181,   -1,   -1,   -1,  -1,  -1,  -1, -1,  3276)
 (1989, 'MF', 3986, 1316,   -1,   -1,  -1,  -1,  833, 2594, 2941,   -1,   -1,   -1,  -1,  -1,  -1, -1,  7532)
 (1989, 'F', 2150,  522,   -1,   -1,  -1,  -1,  621,  702, 1082,   -1,   -1,   -1,  -1,  -1,  -1, -1,  2837)
```

Data Cleaning (Dataset 3)

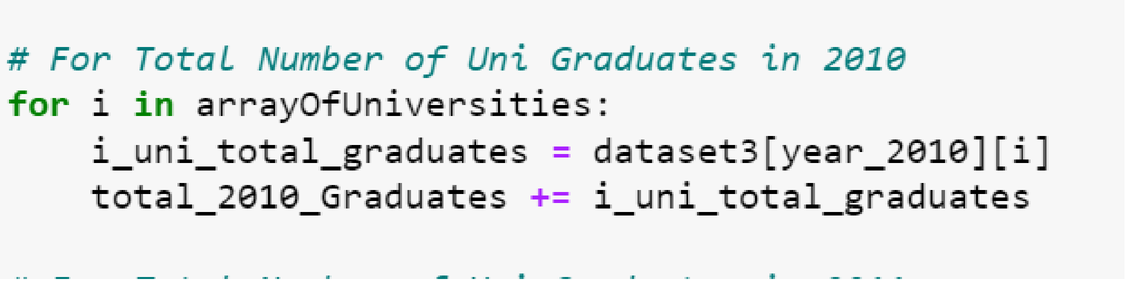
As there were negatives values for columns that were for counting:

- It would mess up our calculations
- Use **boolean indexing** & **np.where()**
  - Replaced all values of -1 to 0 to indicate that the University had not opened in those year



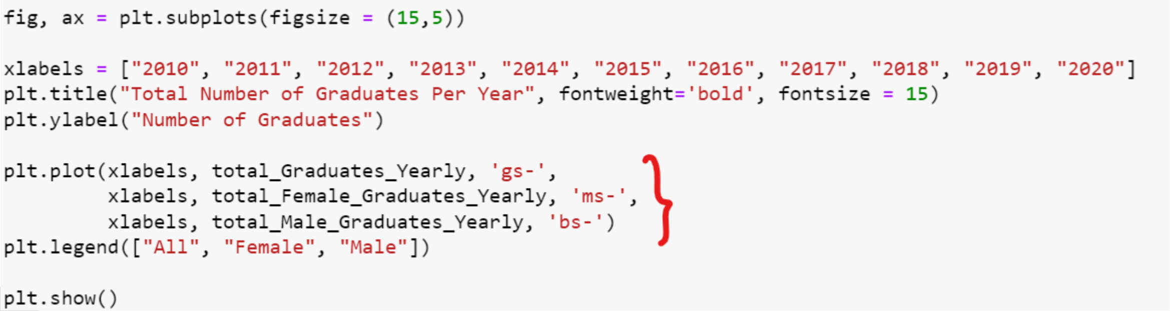
```
# As some univeristies were not opened in earlier years, they were inputted as -1
# Hence, change it to be numerical value 0 to indicate the school wasn't built yet
dataset3["smu"] = np.where(dataset3["smu"] == -1, 0, dataset3["smu"])
dataset3["sit"] = np.where(dataset3["sit"] == -1, 0, dataset3["sit"])
dataset3["sutd"] = np.where(dataset3["sutd"] == -1, 0, dataset3["sutd"])
dataset3["suss"] = np.where(dataset3["suss"] == -1, 0, dataset3["suss"])
dataset3["temasek_polytechnic"] = np.where(dataset3["temasek_polytechnic"] == -1, 0, dataset3["temasek_polytechnic"])
dataset3["nanyang_polytechnic"] = np.where(dataset3["nanyang_polytechnic"] == -1, 0, dataset3["nanyang_polytechnic"])
dataset3["republic_polytechnic"] = np.where(dataset3["republic_polytechnic"] == -1, 0, dataset3["republic_polytechnic"])
dataset3["lasalle_diploma"] = np.where(dataset3["lasalle_diploma"] == -1, 0, dataset3["lasalle_diploma"])
dataset3["lasalle_degree"] = np.where(dataset3["lasalle_degree"] == -1, 0, dataset3["lasalle_degree"])
dataset3["nafa_diploma"] = np.where(dataset3["nafa_diploma"] == -1, 0, dataset3["nafa_diploma"])
dataset3["nafa_degree"] = np.where(dataset3["nafa_degree"] == -1, 0, dataset3["nafa_degree"])
```

Graph 1: Total Number of Graduates Per Year



```
# For Total Number of Uni Graduates in 2010
for i in arrayOfUniversities:
    i_uni_total_graduates = dataset3[year_2010][i]
    total_2010_Graduates += i_uni_total_graduates
```

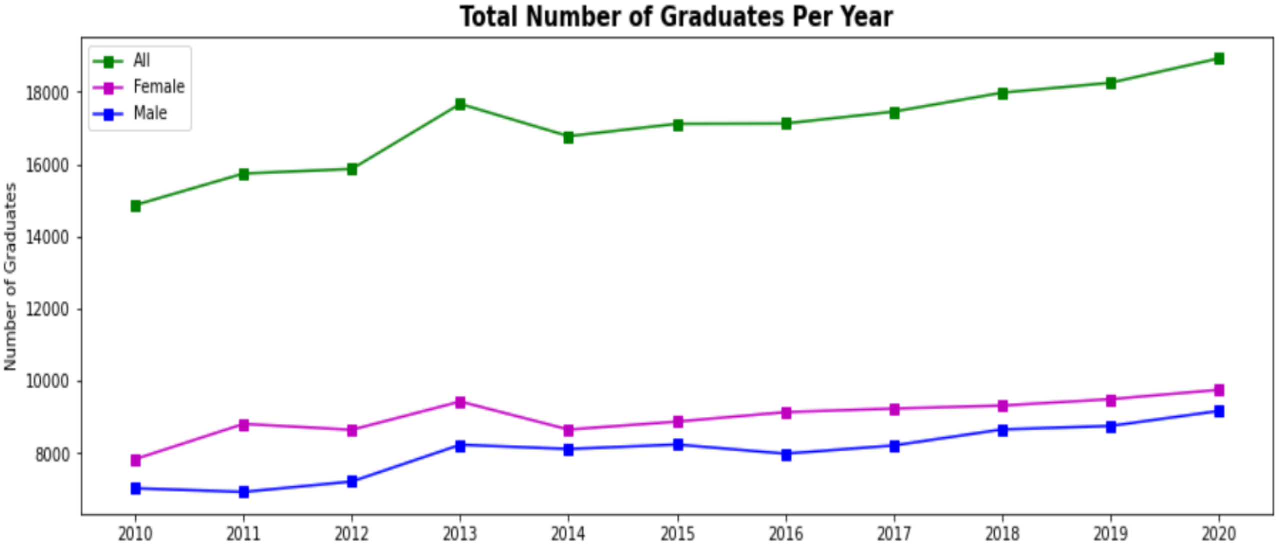
- Created “arrayOfUniversities” to store unique University names
- **For loop**:
  - Add up all number of graduates from every University to get the total for that year



```
fig, ax = plt.subplots(figsize = (15,5))

xlabels = ["2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020"]
plt.title("Total Number of Graduates Per Year", fontweight='bold', fontsize = 15)
plt.ylabel("Number of Graduates")

plt.plot(xlabels, total_Graduates_Yearly, 'gs-',
         xlabels, total_Female_Graduates_Yearly, 'ms-',
         xlabels, total_Male_Graduates_Yearly, 'bs-')
plt.legend(["All", "Female", "Male"])

plt.show()
```

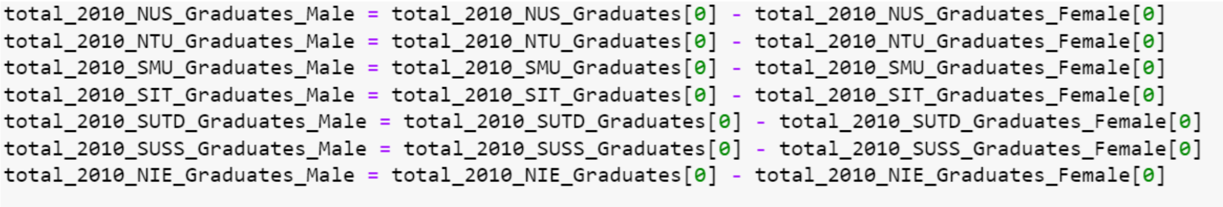
- Plotted **3 different lines** in 1 line graph to display the change in number of graduates in terms of total, female & male

Graph 1: Total Number of Graduates Per Year

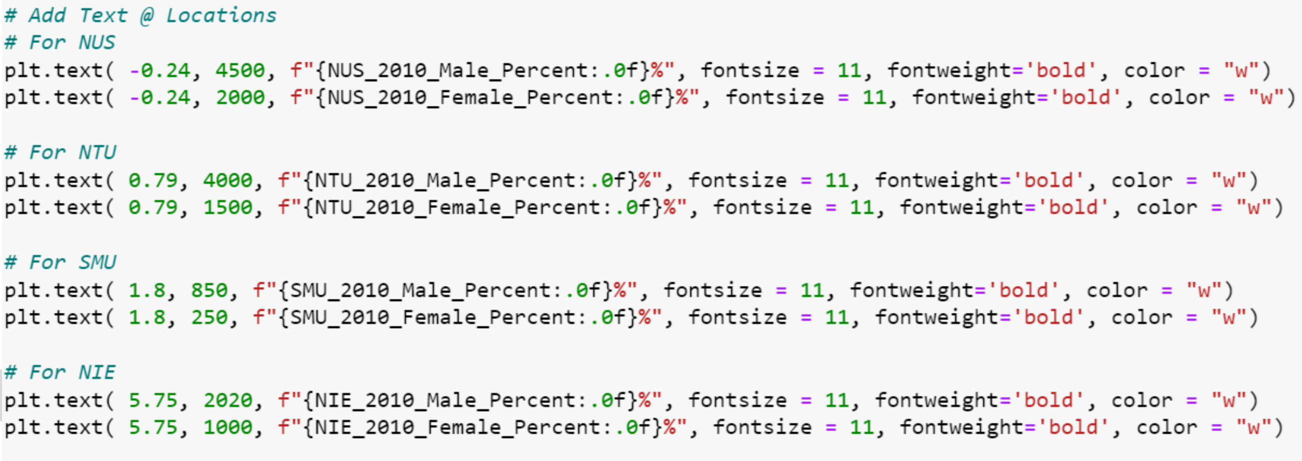


- There is a **steady increase** in total no. of graduates
    - With **one sharp increase** from 2012 to 2013
- There are **more females than male graduates** every year
    - This may be due to 2 years NS for the males

Graph 2: Number of Graduates Per University



```
total_2010_NUS_Graduates_Male = total_2010_NUS_Graduates[0] - total_2010_NUS_Graduates_Female[0]
total_2010_NTU_Graduates_Male = total_2010_NTU_Graduates[0] - total_2010_NTU_Graduates_Female[0]
total_2010_SMU_Graduates_Male = total_2010_SMU_Graduates[0] - total_2010_SMU_Graduates_Female[0]
total_2010_SIT_Graduates_Male = total_2010_SIT_Graduates[0] - total_2010_SIT_Graduates_Female[0]
total_2010_SUTD_Graduates_Male = total_2010_SUTD_Graduates[0] - total_2010_SUTD_Graduates_Female[0]
total_2010_SUSS_Graduates_Male = total_2010_SUSS_Graduates[0] - total_2010_SUSS_Graduates_Female[0]
total_2010_NIE_Graduates_Male = total_2010_NIE_Graduates[0] - total_2010_NIE_Graduates_Female[0]
```

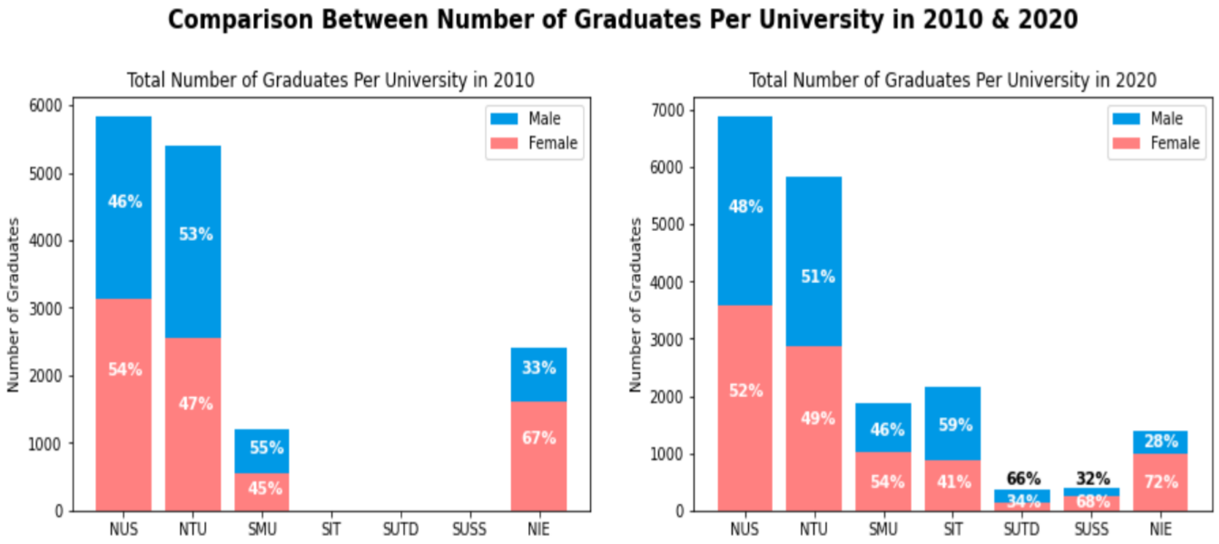


```
# Add Text @ Locations
# For NUS
plt.text( -0.24, 4500, f"{NUS_2010_Male_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")
plt.text( -0.24, 2000, f"{NUS_2010_Female_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")

# For NTU
plt.text( 0.79, 4000, f"{NTU_2010_Male_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")
plt.text( 0.79, 1500, f"{NTU_2010_Female_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")

# For SMU
plt.text( 1.8, 850, f"{SMU_2010_Male_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")
plt.text( 1.8, 250, f"{SMU_2010_Female_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")

# For NIE
plt.text( 5.75, 2020, f"{NIE_2010_Male_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")
plt.text( 5.75, 1000, f"{NIE_2010_Female_Percent:.0f}%", fontsize = 11, fontweight='bold', color = "w")
```

- To retrieve number of male graduates:
  - Performed **"-" operation** to subtract individual elements from arrays containing female graduates & both gender graduates statistics

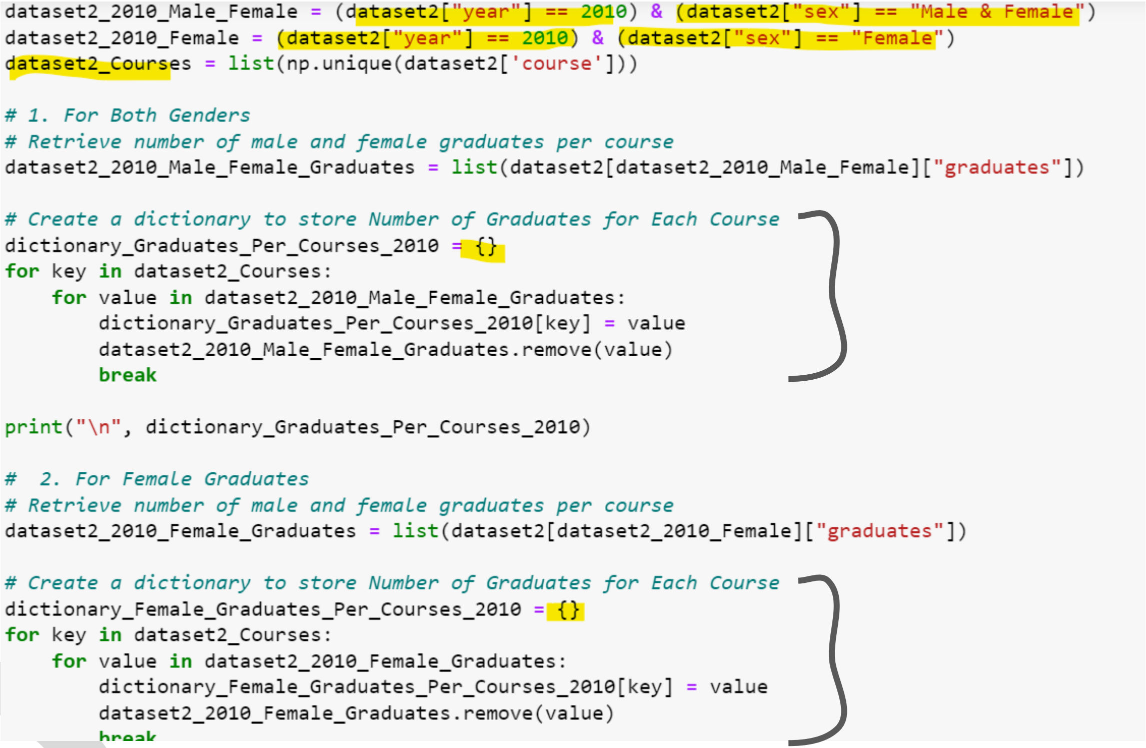
- **plt.text()**:
  - Add text about percentages as it is visually hard to see difference when the bar length is similar

Graph 2: Number of Graduates Per University



- **NUS** has the most number of Graduates
  - Possibly due to its long history
- Generally there is a **even proportion** of genders or a **higher proportion of females** per University

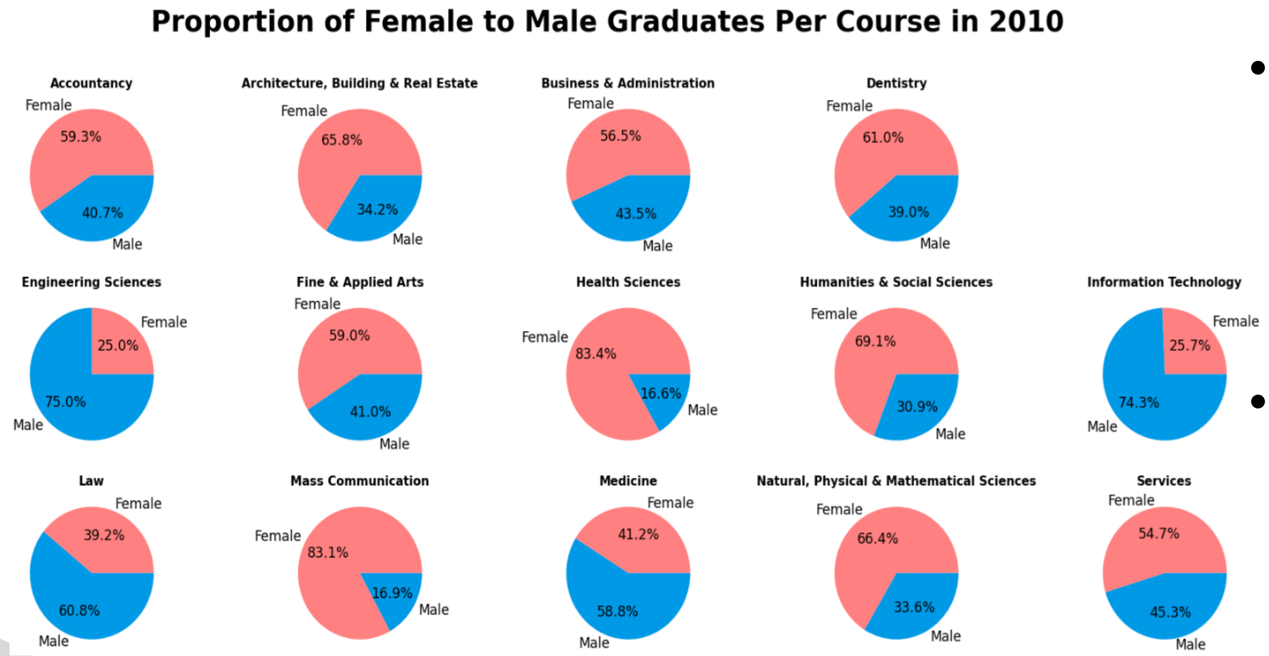
Graph 3 & 4: Proportion of Female To Male Graduates Per Course in 2010 & 2020



```
dataset2_2010_Male_Female = (dataset2["year"] == 2010) & (dataset2["sex"] == "Male & Female")
dataset2_2010_Female = (dataset2["year"] == 2010) & (dataset2["sex"] == "Female")
dataset2_Courses = list(np.unique(dataset2['course']))

# 1. For Both Genders
# Retrieve number of male and female graduates per course
dataset2_2010_Male_Female_Graduates = list(dataset2[dataset2_2010_Male_Female]["graduates"])

# Create a dictionary to store Number of Graduates for Each Course
dictionary_Graduates_Per_Courses_2010 = {}
for key in dataset2_Courses:
    for value in dataset2_2010_Male_Female_Graduates:
        dictionary_Graduates_Per_Courses_2010[key] = value
        dataset2_2010_Male_Female_Graduates.remove(value)
        break

print("\n", dictionary_Graduates_Per_Courses_2010)

#  2. For Female Graduates
# Retrieve number of male and female graduates per course
dataset2_2010_Female_Graduates = list(dataset2[dataset2_2010_Female]["graduates"])

# Create a dictionary to store Number of Graduates for Each Course
dictionary_Female_Graduates_Per_Courses_2010 = {}
for key in dataset2_Courses:
    for value in dataset2_2010_Female_Graduates:
        dictionary_Female_Graduates_Per_Courses_2010[key] = value
        dataset2_2010_Female_Graduates.remove(value)
        break
```

- **Boolean Indexing** to select rows only have “2010” or “2020” & “Male & Female” or “Female”
- **Np.unique()**: get unique courses
- Make **dictionary** to loop through the unique courses to get the number of graduates per unique courses

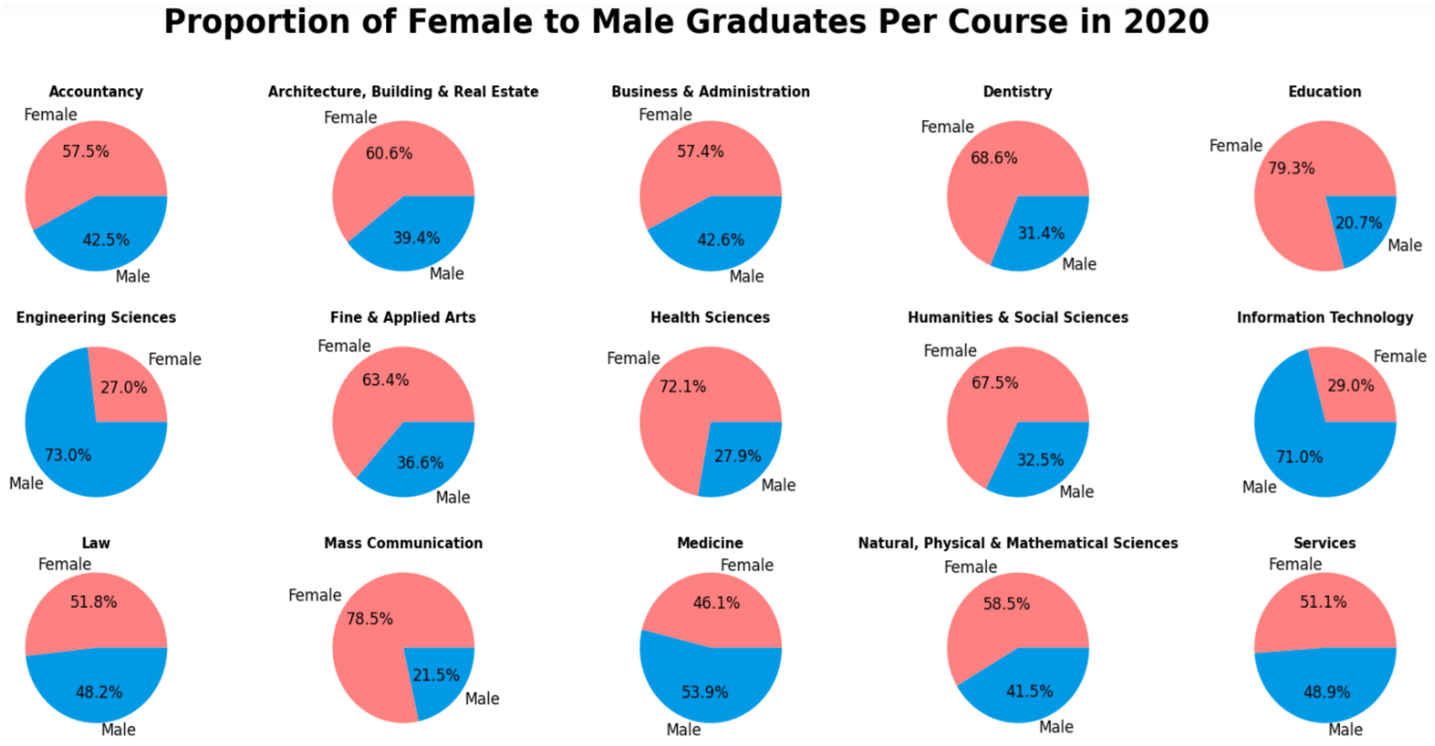
Graph 3: Proportion of Female To Male Graduates Per Course in 2010



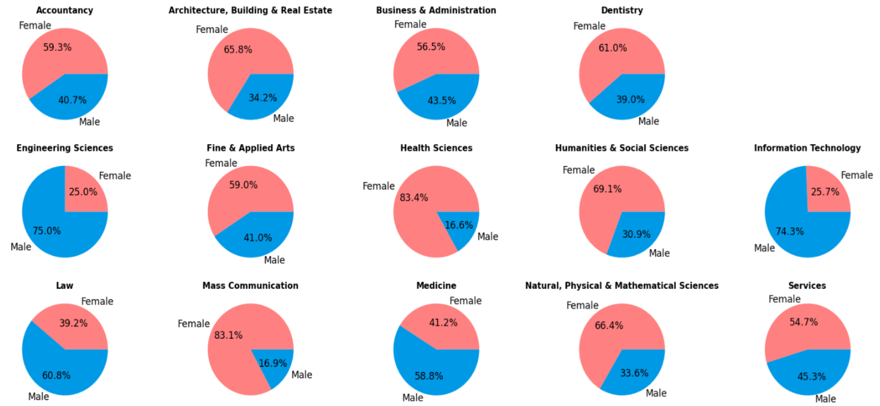
- Generally across the courses, there are **higher proportions of females** to males
  - Especially:
  - Health Sciences
  - Mass Communication
  - Humanities & Social Sciences

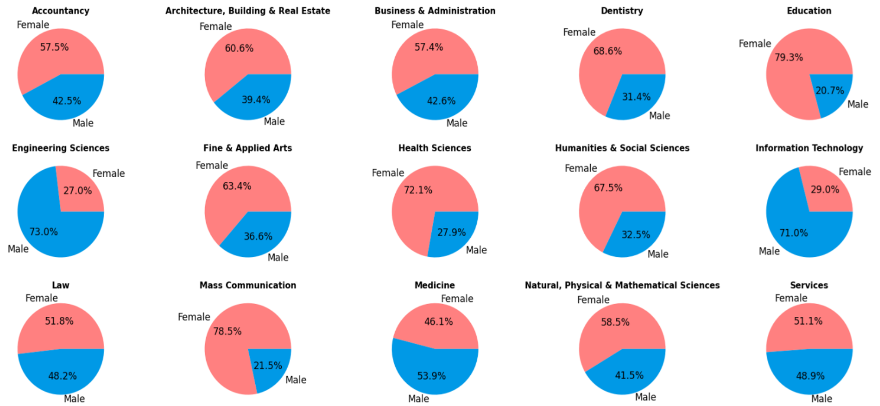
- Courses with **higher proportions of males**:
  - Engineering Sciences
  - Law
  - Medicine
  - Information Technology

Graph 4: Proportion of Female To Male Graduates Per Course in 2020



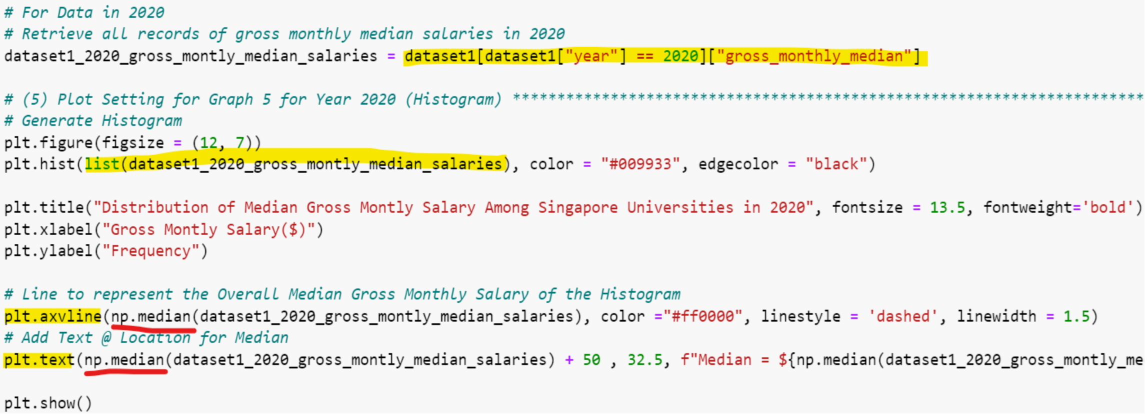
Comparison Between Pie Charts


Proportion of Female to Male Graduates Per Course in 2010


Proportion of Female to Male Graduates Per Course in 2020

- Generally, the **proportions of females to males remained higher** in the same courses in 2010 to 2020
- Exception in 2020:
  - For **Law** Course, the female proportion increased
  - For **Health** Sciences, the male proportion increased from 16.6% to 27.9%

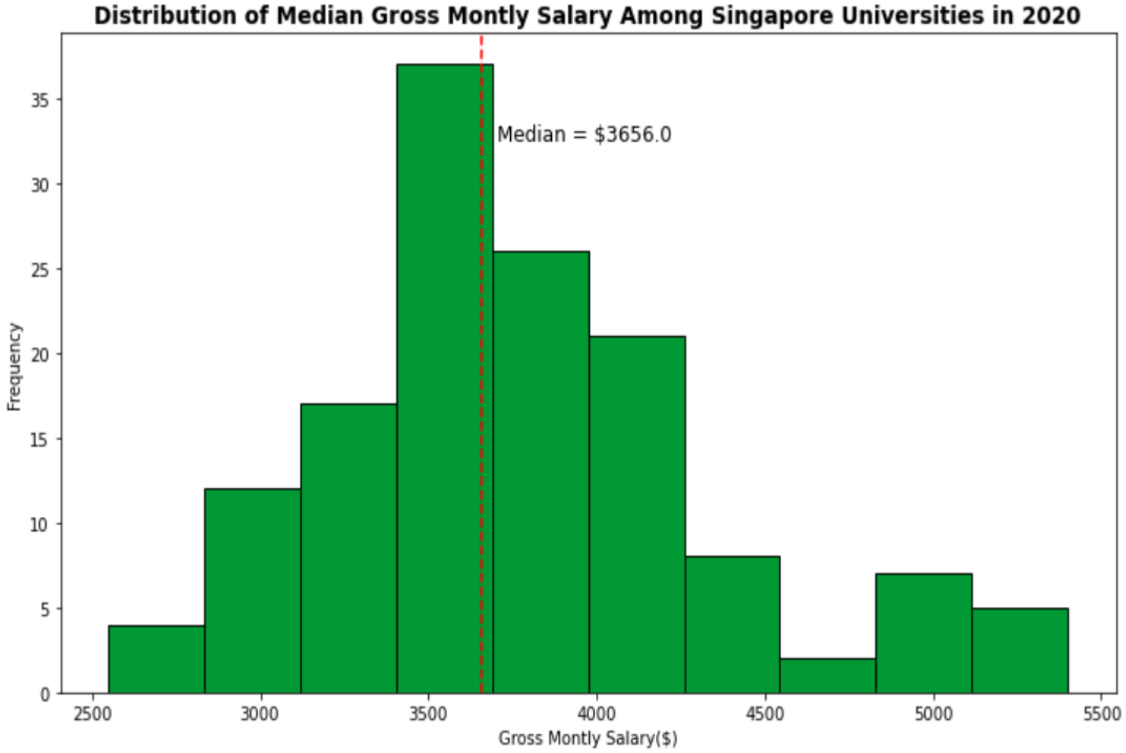
Graph 5: Distribution of Median Gross Monthly Salary Among Singapore Universities in 2020



```
# For Data in 2020
# Retrieve all records of gross monthly median salaries in 2020
dataset1_2020_gross_montly_median_salaries = dataset1[dataset1["year"] == 2020]["gross_monthly_median"]

# (5) Plot Setting for Graph 5 for Year 2020 (Histogram) ************************************************************************
# Generate Histogram
plt.figure(figsize = (12, 7))
plt.hist(list(dataset1_2020_gross_montly_median_salaries), color = "#009933", edgecolor = "black")

plt.title("Distribution of Median Gross Montly Salary Among Singapore Universities in 2020", fontsize = 13.5, fontweight='bold')
plt.xlabel("Gross Montly Salary($)")
plt.ylabel("Frequency")

# Line to represent the Overall Median Gross Monthly Salary of the Histogram
plt.axvline(np.median(dataset1_2020_gross_montly_median_salaries), color ="#ff0000", linestyle = 'dashed', linewidth = 1.5)
# Add Text @ Location for Median
plt.text(np.median(dataset1_2020_gross_montly_median_salaries) + 50 , 32.5, f"Median = ${np.median(dataset1_2020_gross_montly_me

plt.show()
```

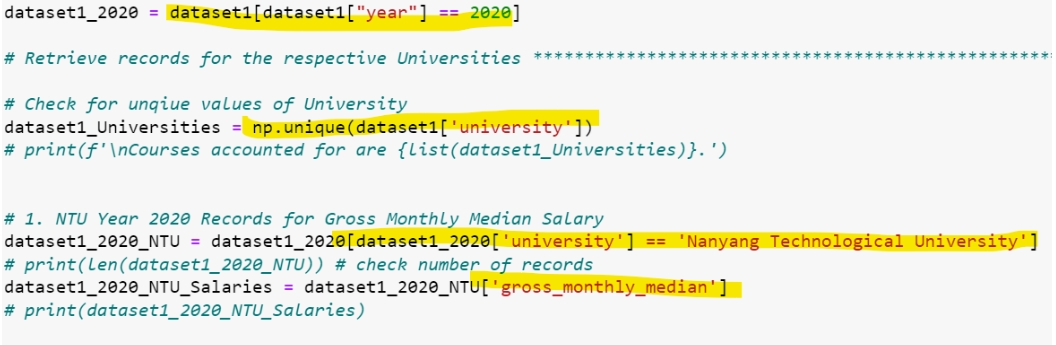
- **Boolean Indexing** to extract statistics all records for gross monthly median in 2020
- **np.median()**:
  - Get <u>median</u> of histogram
- **plt.axvline()**:
  - Generate a <u>dashed line at median</u>
- **plt.text()**:
  - To <u>add text</u> beside dashed line

Graph 5: Distribution of Median Gross Monthly Salary Among Singapore Universities in 2020



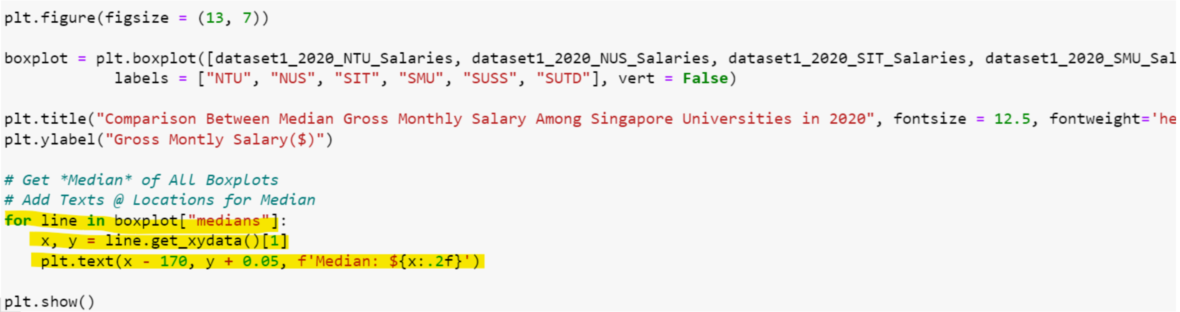
- Distribution is **widespread**
- Distribution is **skewed to one direction**
- Median = $3656

Graph 6: Comparison Between Median Gross Monthly Salary Among Singapore Universities in 2020



```
dataset1_2020 = dataset1[dataset1["year"] == 2020]

# Retrieve records for the respective Universities **************************************************

# Check for unqiue values of University
dataset1_Universities = np.unique(dataset1['university'])
# print(f'\nCourses accounted for are {list(dataset1_Universities)}.')


# 1. NTU Year 2020 Records for Gross Monthly Median Salary
dataset1_2020_NTU = dataset1_2020[dataset1_2020['university'] == 'Nanyang Technological University']
# print(len(dataset1_2020_NTU)) # check number of records
dataset1_2020_NTU_Salaries = dataset1_2020_NTU['gross_monthly_median']
# print(dataset1_2020_NTU_Salaries)
```

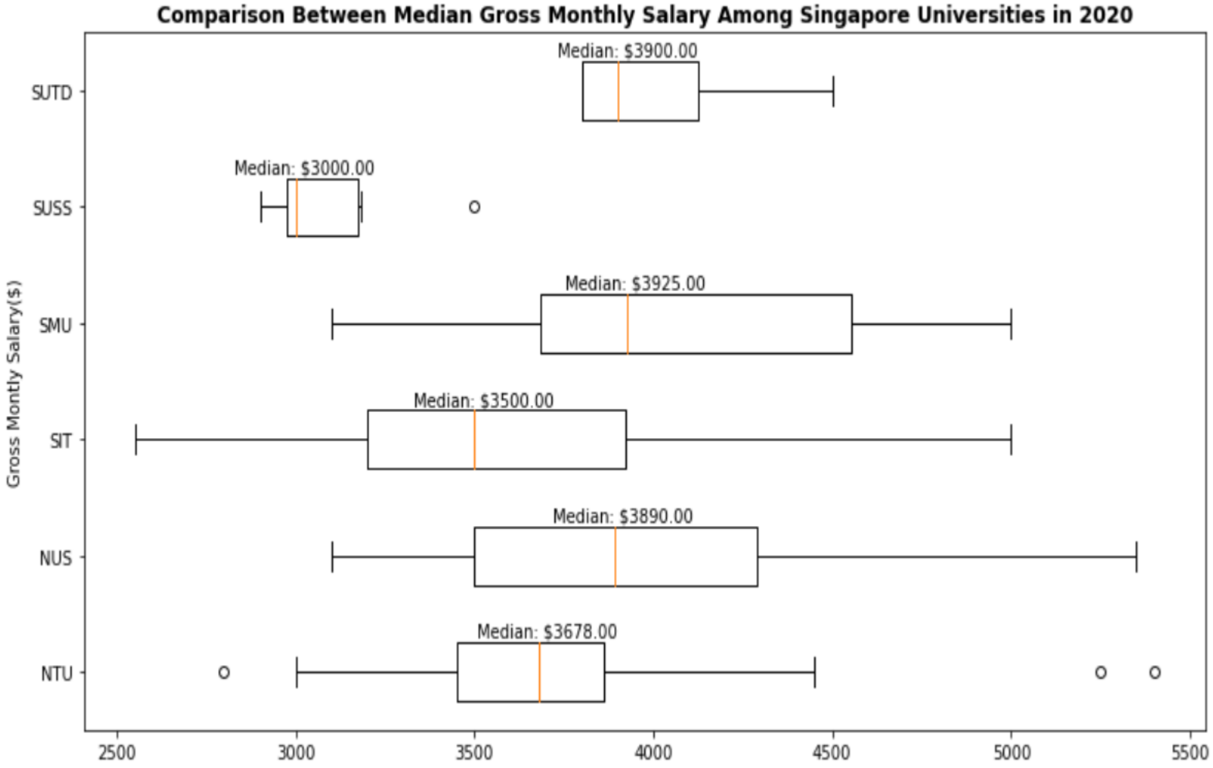
- **Boolean Indexing** with unique university values to extract specific rows with the correct university



```
plt.figure(figsize = (13, 7))

boxplot = plt.boxplot([dataset1_2020_NTU_Salaries, dataset1_2020_NUS_Salaries, dataset1_2020_SIT_Salaries, dataset1_2020_SMU_Sal
          labels = ["NTU", "NUS", "SIT", "SMU", "SUSS", "SUTD"], vert = False)

plt.title("Comparison Between Median Gross Monthly Salary Among Singapore Universities in 2020", fontsize = 12.5, fontweight='he
plt.ylabel("Gross Montly Salary($)")

# Get *Median* of All Boxplots
# Add Texts @ Locations for Median
for line in boxplot["medians"]:
    x, y = line.get_xydata()[1]
    plt.text(x - 170, y + 0.05, f'Median: ${x:.2f}')

plt.show()
```

- **For loop in boxplot['median]:**
    - Retrieve median values from all boxplots & add text to location

Graph 6: Comparison Between Median Gross Monthly Salary Among Singapore Universities in 2020



- **SMU** has the <u>greatest</u> Median Gross Monthly Salary at $3925
- **SUSS** has the <u>least</u> Median Gross Monthly Salary at $30000
- **SIT** has the <u>widest range</u> of Gross Monthly Salary from $2500 to $5000
- **SUTD** has a <u>tight range</u> of Gross Monthly Salary from $3800 to $4500

Conclusion

- Generally, there is **more population of female** than male in Universities
- There is evidence of **advertising of courses towards both genders** in Universities being effective as seen in the increase in
  - Proportion of females in the Law Course from 2010 to 2020
  - Proportion of males in Health Sciences Course from 2010 to 2020
- Universities that have a **longer history** have **higher number of graduates** and higher median **gross monthly income**

Recommendations

- Advise Universities to advertise courses such that **both genders have equal opportunities & exposure to all courses**
- Advise Universities like SUSS, to r**efine their courses** to be more competitive & to **expose their students to the industry**, allowing SUSS to **gain exposure**. This would increase SUSS's students opportunities, giving them more options in their pay, resulting in an increase in gross monthly salary

REFERENCES

https://data.gov.sg/dataset/graduate-employment-survey-ntu-nus-sit-smu-suss-sutd

https://data.gov.sg/dataset/universities-intake-enrolment-and-graduates-by-course

https://data.gov.sg/dataset/intake-enrolment-and-graduates-by-institutions

https://www.freepik.com/free-vector/office-secretary-with-notebook-pencil-her-hand_10704426.htm#query=office%20lady&position=19&from_view=search&track=sph