In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```python
df = pd.read_csv('data/raw.csv')
df = df.iloc[:, 2:]
df.head()
```

Out[2]:

	Age	Gender	Year	Course	Last Semester GWA (Generated Weighted Average)?	1. Have a rich vocabulary	2. Have difficulty understanding abstract ideas	3. Have a vivid imagination	4. Am not interested in abstract ideas	5. Have excellent ideas	...	1. Get stressed out easily	2. Am relaxed most of the time	3. Worry about things	4. Seldom feel blue	5. Am easily disturbed	6. Get upset easily	7. Change my mood a lot	8. Have frequent mood swings	9. Get irritated easily	10. Often feel blue
0	21	Male	Third	Bachelor of Science in Computational Data Science	1.21	4	5	3	3	4	...	3	3	2	3	3	3	3	2	2	4
1	22	Female	Third	Bachelor of Science in Computational and Data ...	1.43	4	2	4	2	3	...	3	4	3	3	2	2	2	1	2	2
2	21	Male	Third	Bachelor of Science in Computational and Data ...	1.30	5	2	5	5	3	...	5	2	5	5	3	4	5	5	5	4
3	22	Male	Third	Bachelor of Engineering Technology in Non-Dest...	1.75	3	2	3	1	4	...	4	3	5	4	4	2	1	1	4	4
4	20	Female	Second	Bachelor of Science in Business Administration	1.75	3	4	4	5	3	...	5	2	4	4	4	4	5	4	4	4

5 rows × 55 columns

Inspecting Age

In [3]:

```python
df['Age'].value_counts()
```

Out[3]:

```
21            68
20            27
22             9
23             3
19             3
20 yrs old     1
Name: Age, dtype: int64
```

In [4]:

```python
df['Age'] = df['Age'].apply(lambda x: int(x.replace('20 yrs old', '20')))
```

In [5]:

```python
df['Age'].unique()
```

Out[5]:

```
array([21, 22, 20, 23, 19], dtype=int64)
```

The data have some inappropriate data entry which is "20 years old". We transform it into 20 and all the values are converted into int

Inspecting Gender and Year

In [6]:

```python
df['Gender'].value_counts()
```

Out[6]:

```
Female               66
Male                 40
Prefer not to say     5
Name: Gender, dtype: int64
```

In [7]:

```python
df['Year'].value_counts()
```

Out[7]:

```
Third     89
Second    16
First      4
Fourth     2
Name: Year, dtype: int64
```

In [8]:

```python
df['Year'].value_counts()
```

Out[8]:

```
Third     89
Second    16
First      4
Fourth     2
Name: Year, dtype: int64
```

Seems like there is no problem with these 2 features.

Inspecting Course Feature

In [9]:

```python
df['Course'].value_counts()
```

Out[9]:

```
Bachelor of Science in Computational and Data Sciences                9
Bachelor of Science in Nursing                                        5
Bachelor of Science in Accountancy                                    4
Bachelor of Science in Computer Science                               4
Bachelor of Science in Civil Engineering                              4
                                                                     ..
Bachelor of Science in Application  Development                       1
Bachelor of Science in Computer Science major in Application          1
Bachelor of Science in College of Computing in Information Science    1
Bachelor in Multimedia Arts Major in Animation                        1
Bachelor of Science in Medical Technology                             1
Name: Course, Length: 73, dtype: int64
```

In [10]:

```python
df['Course'].nunique()
```

Out[10]:

```
73
```

In [11]:

```python
from fuzzywuzzy import process
import fuzzywuzzy

def replace_matches_in_column(df, column, string_to_match, thresh = 55):
    strings = df[column].unique()
    matches = fuzzywuzzy.process.extract(string_to_match, strings, limit=20, scorer=fuzzywuzzy.fuzz.token_sort_ratio)
    close_matches = [match[0] for match in matches if match[1] >= thresh]
    rows_with_matches = df[column].isin(close_matches)
    df.loc[rows_with_matches, column] = string_to_match
```

```
D:\anaconda\lib\site-packages\fuzzywuzzy\fuzz.py:11: UserWarning: Using slow pure-python SequenceMatcher. Install python-Levenshtein to remove this warning
  warnings.warn('Using slow pure-python SequenceMatcher. Install python-Levenshtein to remove this warning')
```

In [12]:

```python
for x in df['Course'].unique():
    replace_matches_in_column(df, 'Course', x, 90)

print('replace possible duplicate entries!')
```

```
replace possible duplicate entries!
```

In [13]:

```python
df['Course'].value_counts()
```

Out[13]:

```
Bachelor of Science in Computation and Data Sciences                                   15
Bachelor of Science in Accountancy                                                      8
Bachelor of Science in Nursing                                                          6
Bachelor Of Science in Computer Science Major in Application Development                6
Bachelor of Science Major in Psychology                                                 5
Bachelor of Science in Civil Engineering                                                4
Bachelor of Science in Information Technology                                           4
Bachelor of Science in Computer Science                                                 4
Bachelor of science Business Administration                                             4
Bachelor of Science in Office Administration                                            4
BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING                                           2
Bachelor of Science in Business Administration Major in Human Resource Development Management      2
Bachelor of Science in Tourism Management                                               2
Bachelor of Science in Business Administration major in Marketing Management            2
Diploma in civil engineering technology                                                 2
Bachelor of Science in Hospitality Management                                           2
Bachelor of Science in Application  Development                                         2
Bachelor of Secondary Education Major in Filipino                                       2
Bachelor of Science in Radiologic Technology                                            2
Bachelor of Science in Environmental Science                                            2
Bachelor of Engineering Technology Major in Electronics Technology                      2
Bachelor of Science in Biology                                                          2
Bachelor of Science in Architecture                                                     2
Bachelor of Science in Management Accounting                                            2
Bachelor in Graphics Technology Major in Architectural Technology                       1
BACHELOR OF BUSINESS ADMININISTRATION MAJOR IN HUMAN RESOURCE DEVELOPMENT               1
Bachelor of Secondary Education major in Math                                           1
Bachelor of Science in  Industrial Engineering                                          1
Bachelor of Science of Office Management                                                1
Bachelor of Science in Education major in English program                               1
Doctor of Optometry                                                                     1
DIPLOMA IN INDUSTRIAL FACILITIES TECHNOLOGYMAJOR IN SERVICE MECHANICS                   1
Diploma in Civil Engineering Technology (Ladderized)                                    1
Bachelor of Science in Instrumentatiin and Control Engineering                          1
Bachelor of Science in Mechanical Engineering                                           1
Bachelor of Early Childhood Education                                                   1
Bachelor of Science in Railway Engineering                                              1
Bachelor in Secondary  Education Major in Mathematics                                   1
Bachelor of Secondary Education Major in Social Studies                                 1
Bachelor in Multimedia Arts Major in Animation                                          1
Bachelor of Science in College of Computing in Information Science                      1
Bachelor of Physical Education                                                          1
Bachelor of Science in Aviation Electronics Technology                                  1
Bachelor of Engineering Technology in Non-Destructive Testing Engineering                1
 Bachelor of Science in Food Technology                                                 1
BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGYMAJOR IN INFORMATION AND NETWORK SECURITY  1
Bachelor of Science in Medical Technology                                               1
Name: Course, dtype: int64
```

In [14]:

```python
df['Course'].nunique()
```

Out[14]:

```
47
```

There are multiple duplicated entry. The solution was using fuzzy wuzzy to transform all the related features into 1 value only.

Inspecting GWA

In [15]:

```python
df['Last Semester GWA (Generated Weighted Average)?'].value_counts()
```

Out[15]:

```
1.7500    16
1.5000    12
1.4200     4
1.2100     3
1.3300     3
1.5300     3
1.1400     3
1.7200     3
1.7100     3
1.1800     3
1.7900     2
1.4900     2
1.7000     2
1.2900     2
1.4500     2
1.4300     2
1.3100     2
1.8000     2
1.3000     2
1.5700     2
1.9800     1
1.0600     1
1.8903     1
1.2500     1
1.9600     1
1.7400     1
1.4400     1
1.2000     1
1.1100     1
2.7000     1
1.3400     1
1.6500     1
1.2200     1
1.5800     1
1.2969     1
1.8900     1
2.3000     1
2.0000     1
1.6300     1
1.0000     1
1.6800     1
1.3600     1
1.5200     1
1.9100     1
2.2100     1
1.9300     1
1.5940     1
1.3200     1
1.4600     1
1.9000     1
2.2500     1
2.1000     1
1.1300     1
1.8800     1
2.0500     1
1.0900     1
1.6700     1
3.7600     1
Name: Last Semester GWA (Generated Weighted Average)?, dtype: int64
```

No problem with the GWA

Processing OCEAN

In [16]:

```python
df.iloc[:,5:15]
```

Out[16]:

	1. Have a rich vocabulary	2. Have difficulty understanding abstract ideas	3. Have a vivid imagination	4. Am not interested in abstract ideas	5. Have excellent ideas	6. Do not have a good imagination	7. Am quick to understand things	8. Use difficult words	9. Spend time reflecting on things	10. Am full of ideas
0	4	5	3	3	4	3	4	5	5	3
1	4	2	4	2	3	4	4	4	5	3
2	5	2	5	5	3	4	5	3	5	4
3	3	2	3	1	4	3	4	2	5	3
4	3	4	4	5	3	2	3	2	2	3
...	...	...	...	...	...	...	...	...	...	...
106	3	3	3	2	3	3	3	3	4	3
107	4	3	3	2	3	2	4	3	4	3
108	2	2	4	3	3	2	4	1	3	3
109	3	2	4	1	4	3	4	3	5	4
110	5	4	5	2	5	2	5	4	4	4

111 rows × 10 columns

In [17]:

```python
df['Openness to Experience'] = 8 + df.iloc[:,5] - df.iloc[:,6] + df.iloc[:,7] - df.iloc[:,8] + df.iloc[:,9] - df.iloc[:,10] + df.iloc[:,11] + df.iloc[:,12] + df.iloc[:,13] + df.iloc[:,14]
df['Conscientiousness'] = 14 +  df.iloc[:,15] - df.iloc[:,16] + df.iloc[:,17] - df.iloc[:,18] + df.iloc[:,19] - df.iloc[:,20] + df.iloc[:,21] - df.iloc[:,22] + df.iloc[:,23] + df.iloc[:,24]
df['Extroversion'] = 20 +  df.iloc[:,25] - df.iloc[:,26] + df.iloc[:,27] - df.iloc[:,28] + df.iloc[:,29] - df.iloc[:,30] + df.iloc[:,31] - df.iloc[:,32] + df.iloc[:,33] - df.iloc[:,34]
df['Agreeableness'] = 14 - df.iloc[:,35] + df.iloc[:,36] - df.iloc[:,37] + df.iloc[:,38] - df.iloc[:,39] + df.iloc[:,40] - df.iloc[:,41] + df.iloc[:,42] + df.iloc[:,43] + df.iloc[:,44]
df['Neuroticism'] = 38 - df.iloc[:,45] + df.iloc[:,46] - df.iloc[:,47] + df.iloc[:,48] - df.iloc[:,49] - df.iloc[:,50] - df.iloc[:,51] - df.iloc[:,52] - df.iloc[:,53] - df.iloc[:,54]
```

We transform the results of the questions into the Big Five Personality (OCEAN). This is calculated using the formula from: https://openpsychometrics.org/printable/big-five-personality-test.pdf

Finalizing all the Features

In [18]:

```python
df.columns
```

Out[18]:

```
Index(['Age', 'Gender', 'Year', 'Course',
       'Last Semester GWA (Generated Weighted Average)?',
       '1. Have a rich vocabulary',
       '2. Have difficulty understanding abstract ideas',
       '3. Have a vivid imagination', '4. Am not interested in abstract ideas',
       '5. Have excellent ideas', '6. Do not have a good imagination',
       '7. Am quick to understand things', '8. Use difficult words',
       '9. Spend time reflecting on things', '10. Am full of ideas',
       '1. Am always prepared.', '2. Leave my belongings around',
       '3. Pay attention to details', '4. Make a mess of things',
       '5. Get chores done right away',
       '6. Often forget to put things back in their proper place',
       '7. Like order', '8. Shirk my duties', '9. Follow a schedule',
       '10. Am exacting in my work', '1. Am the life of the party',
       '2. Don't talk a lot', '3. Feel comfortable around people',
       '4. Keep in the background', '5. Start conversations',
       '6. Have little to say',
       '7. Talk to a lot of different people at parties',
       '8. Don't like to draw attention to myself',
       '9. Don't mind being the center of attention',
       '10. Am quiet around strangers', '1. Feel little concern for others',
       '2. Am interested in people', '3. Insult people',
       '4. Sympathize with others' feelings',
       '5. Am not interested in other people's problems',
       '6. Have a soft heart', '7. Am not really interested in others',
       '8. Take time out for others', '9. Feel others' emotions',
       '10. Make people feel at ease', '1. Get stressed out easily',
       '2. Am relaxed most of the time', '3. Worry about things',
       '4. Seldom feel blue', '5. Am easily disturbed', '6. Get upset easily',
       '7. Change my mood a lot', '8. Have frequent mood swings',
       '9. Get irritated easily', '10. Often feel blue',
       'Openness to Experience', 'Conscientiousness', 'Extroversion',
       'Agreeableness', 'Neuroticism'],
      dtype='object')
```

In [19]:

```python
df.columns = df.columns.str.replace(' ', '_').str.lower()
df.rename(columns = {'last_semester_gwa_(generated_weighted_average)?':'gwa'}, inplace=True)
df.columns
```

Out[19]:

```
Index(['age', 'gender', 'year', 'course', 'gwa', '1._have_a_rich_vocabulary',
       '2._have_difficulty_understanding_abstract_ideas',
       '3._have_a_vivid_imagination', '4._am_not_interested_in_abstract_ideas',
       '5._have_excellent_ideas', '6._do_not_have_a_good_imagination',
       '7._am_quick_to_understand_things', '8._use_difficult_words',
       '9._spend_time_reflecting_on_things', '10._am_full_of_ideas',
       '1._am_always_prepared.', '2._leave_my_belongings_around',
       '3._pay_attention_to_details', '4._make_a_mess_of_things',
       '5._get_chores_done_right_away',
       '6._often_forget_to_put_things_back_in_their_proper_place',
       '7._like_order', '8._shirk_my_duties', '9._follow_a_schedule',
       '10._am_exacting_in_my_work', '1._am_the_life_of_the_party',
       '2._don't_talk_a_lot', '3._feel_comfortable_around_people',
       '4._keep_in_the_background', '5._start_conversations',
       '6._have_little_to_say',
       '7._talk_to_a_lot_of_different_people_at_parties',
       '8._don't_like_to_draw_attention_to_myself',
       '9._don't_mind_being_the_center_of_attention',
       '10._am_quiet_around_strangers', '1._feel_little_concern_for_others',
       '2._am_interested_in_people', '3._insult_people',
       '4._sympathize_with_others'_feelings',
       '5._am_not_interested_in_other_people's_problems',
       '6._have_a_soft_heart', '7._am_not_really_interested_in_others',
       '8._take_time_out_for_others', '9._feel_others'_emotions',
       '10._make_people_feel_at_ease', '1._get_stressed_out_easily',
       '2._am_relaxed_most_of_the_time', '3._worry_about_things',
       '4._seldom_feel_blue', '5._am_easily_disturbed', '6._get_upset_easily',
       '7._change_my_mood_a_lot', '8._have_frequent_mood_swings',
       '9._get_irritated_easily', '10._often_feel_blue',
       'openness_to_experience', 'conscientiousness', 'extroversion',
       'agreeableness', 'neuroticism'],
      dtype='object')
```

In [20]:

```python
cols = ['age', 'gender', 'year', 'course', 'gwa', 'openness_to_experience',
        'conscientiousness', 'extroversion','agreeableness', 'neuroticism']

clean_df = df[cols]
clean_df.head()
```

Out[20]:

	age	gender	year	course	gwa	openness_to_experience	conscientiousness	extroversion	agreeableness	neuroticism
0	21	Male	Third	Bachelor of Science in Computation and Data Sc...	1.21	25	28	22	25	22
1	22	Female	Third	Bachelor of Science in Computation and Data Sc...	1.43	27	36	7	20	28
2	21	Male	Third	Bachelor of Science in Computation and Data Sc...	1.30	27	32	35	30	9
3	22	Male	Third	Bachelor of Engineering Technology in Non-Dest...	1.75	26	27	18	28	20
4	20	Female	Second	Bachelor of science Business Administration	1.75	17	17	16	24	10

In [21]:

```python
clean_df.shape
```

Out[21]:

```
(111, 10)
```

The final dataset contains 111 rows and 10 features. All respective data entries are fixed and ready for further processing.

Checking Outliers

In [22]:

```python
def check_outliers(df, column):
    per25, per75 = np.percentile(df[column], [25, 75])
    iqr = per75 - per25
    lb = per25 - (1.5 * iqr)
    up = per75 + (1.5 * iqr)
    return df[(df[column] < lb) | (df[column] > up)].index, up

def treat_outliers(df, column, state):
    indexes, upbound = check_outliers(df, column)
    if state == 'delete':
        df.drop(indexes, axis = 0, inplace=True)
        print('Outlier Deleted!')
    elif state == 'cap':
        for x in indexes:
            df.loc[x, column] = upbound
        print('Ouliter Cappped!')
```

In [23]:

```python
check_outliers(clean_df, 'gwa')
```

Out[23]:

```
(Int64Index([57, 110], dtype='int64'), 2.3725)
```

In [24]:

```python
treat_outliers(clean_df, 'gwa', 'cap')
```

```
Ouliter Cappped!
D:\anaconda\lib\site-packages\pandas\core\indexing.py:1817: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(loc, value, pi)
```

In [25]:

```python
check_outliers(clean_df, 'gwa')
```

Out[25]:

```
(Int64Index([], dtype='int64'), 2.3725)
```

In [26]:

```python
clean_df.shape
```

Out[26]:

```
(111, 10)
```

Based from our visualizations, there are 2 outliers in gwa feature. We checked it using 1.5 iqr. Since the data is limited, we capped the outliers into the upper bound instead of droppping them.

Scaling

In [27]:

```python
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
```

In [28]:

```python
encoder = LabelEncoder()
scaler = MinMaxScaler()
for x in ['gender', 'year', 'course']:
    clean_df[x] = encoder.fit_transform(clean_df[x])

cols = ['openness_to_experience','conscientiousness','extroversion','agreeableness','neuroticism']
clean_df[cols] = scaler.fit_transform(clean_df[cols])

clean_df.head()
```

```
C:\Users\rurue\AppData\Local\Temp/ipykernel_1788/3430215838.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  clean_df[x] = encoder.fit_transform(clean_df[x])
D:\anaconda\lib\site-packages\pandas\core\frame.py:3678: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self[col] = igetitem(value, i)
```

Out[28]:

	age	gender	year	course	gwa	openness_to_experience	conscientiousness	extroversion	agreeableness	neuroticism
0	21	1	3	23	1.21	0.500000	0.62500	0.567568	0.423077	0.709677
1	22	0	3	23	1.43	0.583333	0.87500	0.162162	0.230769	0.903226
2	21	1	3	23	1.30	0.583333	0.75000	0.918919	0.615385	0.290323
3	22	1	3	10	1.75	0.541667	0.59375	0.459459	0.538462	0.645161
4	20	0	2	42	1.75	0.166667	0.28125	0.405405	0.384615	0.322581

The MinMax Scaler is calculated using the formula x-min/max-min. These will give us values within 0-1. This is required for reducing computational power and is required for distance based algorithms.

Saving clean dataset

In [29]:

```python
clean_df.to_csv('data/clean.csv', index=False)
```