AMCAT Data Analysis - An Exploratory Data Analysis

AMCAT known as Aspiring Minds Computer Adaptive Test is an AI-based computer adaptive test which evaluates job applicants on critical areas like communication skills, logical reasoning, quantitative skills, and job-specific domain skills thereby helping recruiters identify the suitability of a candidate for different job roles.

A study carried out on students with engineering disciplines by Aspiring Minds from the Aspiring Mind Employment Outcome(AMEO) showed the employment outcomes of engineering graduates along with the standardized scores from three different areas - cognitive skills, technical skills and personality skills. The study also recorded the demographic features for each candidate.

With this study, the AMCAT team were able to gather concrete data with which they hoped to understand what has become of candidates since they took part in the tests.They also have highlighted some areas or questions of interest they would like to have answers to. As a data scientist on the AMCAT team, my task is to perform an comprehensive analysis, leaving no stones unturned.

The Data

Column	Description
`'ID'`	Unique ID to identify a candidate
`'Salary'`	Annual CTC offered to the candidate (in Indian Rupees)
`'DOJ'`	Date of joining the company
`'DOL'`	Date of leaving the company
`'Designation'`	Designation offered in the job
`'JobCity'`	Location of the job (city)
`'Gender'`	Candidate's gender
`'DOB'`	Date of birth of candidate
`'10percentage'`	Overall marks obtained in grade 10 examinations
`'10board'`	The school board whose curriculum the candidate followed in grade 10
`'12graduation'`	Year of graduation - senior year high school
`'12percentage'`	Overall marks obtained in grade 12 examinations
`'12board'`	The school board whose curriculum the candidate followed in grade 12
`'CollegeID'`	Unique ID identifying the college which the candidate attended
`'CollegeTier'`	Tier of college
`'Degree'`	Degree obtained/pursued by the candidate
`'Specialization'`	Specialization pursued by the candidate
`'CollegeGPA'`	Aggregate GPA at graduation
`'CollegeCityID'`	A unique ID to identify the city in which the college is located in
`'CollegeCityTier'`	The tier of the city in which the college is located
`'CollegeState'`	Name of States
`'GraduationYear'`	Year of graduation (Bachelor's degree)
`'English'`	Scores in AMCAT English section
`'Logical'`	Scores in AMCAT Logical section
`'Quant'`	Scores in AMCAT Quantitative section
`'Domain'`	Scores in AMCAT's domain module
`'ComputerProgramming'`	Score in AMCAT's Computer programming section
`'ElectronicsAndSemicon'`	Score in AMCAT's Electronics & Semiconductor Engineering section
`'ComputerScience'`	Score in AMCAT's Computer Science section
`'MechanicalEngg'`	Score in AMCAT's Mechanical Engineering section
`'ElectricalEngg'`	Score in AMCAT's Electrical Engineering section
`'TelecomEngg'`	Score in AMCAT's Telecommunication Engineering section
`'CivilEngg'`	Score in AMCAT's Civil Engineering section
`'conscientiousness'`	Scores in one of the sections of AMCAT's personality test
`'agreeableness'`	Scores in one of the sections of AMCAT's personality test
`'extraversion'`	Scores in one of the sections of AMCAT's personality test

`'neuroticism'`	Scores in one of the sections of AMCAT's personality test
`'openess_to_experience'`	Scores in one of the sections of AMCAT's personality test

In [2]:
```
#import the pandas library and read data into a dataframe

import pandas as pd

amcat = pd.read_csv('AMCAT.csv')
amcat.head()
```

Out[2]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScience	MechanicalEng
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	...	-1	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	...	-1	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	...	-1	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	...	-1	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	...	-1	

5 rows × 39 columns

Understanding the characteristics of the amcat data

In [3]:
```
# to check the shape of the data

amcat.shape
```

Out[3]:
```
(3998, 39)
```

In [4]:
```
# to check the column characteristics

amcat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Unnamed: 0             3998 non-null   object 
 1   ID                     3998 non-null   int64  
 2   Salary                 3998 non-null   float64
 3   DOJ                    3998 non-null   object 
 4   DOL                    3998 non-null   object 
 5   Designation            3998 non-null   object 
 6   JobCity                3998 non-null   object 
 7   Gender                 3998 non-null   object 
 8   DOB                    3998 non-null   object 
 9   10percentage           3998 non-null   float64
 10  10board                3998 non-null   object 
 11  12graduation           3998 non-null   int64  
 12  12percentage           3998 non-null   float64
 13  12board                3998 non-null   object 
 14  CollegeID              3998 non-null   int64  
 15  CollegeTier            3998 non-null   int64  
 16  Degree                 3998 non-null   object 
 17  Specialization         3998 non-null   object 
 18  collegeGPA             3998 non-null   float64
 19  CollegeCityID          3998 non-null   int64  
 20  CollegeCityTier        3998 non-null   int64  
 21  CollegeState           3998 non-null   object 
 22  GraduationYear         3998 non-null   int64  
 23  English                3998 non-null   int64  
 24  Logical                3998 non-null   int64  
 25  Quant                  3998 non-null   int64  
 26  Domain                 3998 non-null   float64
 27  ComputerProgramming    3998 non-null   int64  
 28  ElectronicsAndSemicon  3998 non-null   int64  
 29  ComputerScience        3998 non-null   int64  
 30  MechanicalEngg         3998 non-null   int64  
 31  ElectricalEngg         3998 non-null   int64  
 32  TelecomEngg            3998 non-null   int64  
 33  CivilEngg              3998 non-null   int64  
 34  conscientiousness      3998 non-null   float64
 35  agreeableness          3998 non-null   float64
 36  extraversion           3998 non-null   float64
 37  nueroticism            3998 non-null   float64
 38  openess_to_experience  3998 non-null   float64
dtypes: float64(10), int64(17), object(12)
memory usage: 1.2+ MB
```

In [5]:

```
# to count the number of missing values in each column

amcat.isna().sum()
```

Out[5]:
```
Unnamed: 0              0
ID                      0
Salary                  0
DOJ                     0
DOL                     0
Designation             0
JobCity                 0
Gender                  0
DOB                     0
10percentage            0
10board                 0
12graduation            0
12percentage            0
12board                 0
CollegeID               0
CollegeTier             0
Degree                  0
Specialization          0
collegeGPA              0
CollegeCityID           0
CollegeCityTier         0
CollegeState            0
GraduationYear          0
English                 0
Logical                 0
Quant                   0
Domain                  0
ComputerProgramming     0
ElectronicsAndSemicon   0
ComputerScience         0
MechanicalEngg          0
ElectricalEngg          0
TelecomEngg             0
CivilEngg               0
conscientiousness       0
agreeableness           0
extraversion            0
nueroticism             0
openess_to_experience   0
dtype: int64
```

In [6]:
```
# to look at the data types alone

amcat.dtypes
```

Out[6]:
```
Unnamed: 0               object
ID                        int64
Salary                  float64
DOJ                      object
DOL                      object
Designation              object
JobCity                  object
Gender                   object
DOB                      object
10percentage            float64
10board                  object
12graduation              int64
12percentage            float64
12board                  object
CollegeID                 int64
CollegeTier               int64
Degree                   object
Specialization           object
collegeGPA              float64
CollegeCityID             int64
CollegeCityTier           int64
CollegeState             object
GraduationYear            int64
English                   int64
Logical                   int64
Quant                     int64
Domain                  float64
ComputerProgramming       int64
ElectronicsAndSemicon     int64
ComputerScience           int64
MechanicalEngg            int64
ElectricalEngg            int64
TelecomEngg               int64
CivilEngg                 int64
conscientiousness       float64
agreeableness           float64
extraversion            float64
nueroticism             float64
openess_to_experience   float64
dtype: object
```

In [7]:
```
# summary statistics on the numerical columns

amcat.describe()
```

Out[7]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	Colleg
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	399
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	1.925713	71.486171	5156.851426	
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	0.262270	8.167338	4802.261482	
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	1.000000	6.450000	2.000000	
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	2.000000	66.407500	494.000000	
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	2.000000	71.720000	3879.000000	
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	2.000000	76.327500	8818.000000	
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	2.000000	99.930000	18409.000000	

8 rows × 27 columns

In [8]:
```
amcat.columns
```

Out[8]:
```
Index(['Unnamed: 0', 'ID', 'Salary', 'DOJ', 'DOL', 'Designation', 'JobCity',
       'Gender', 'DOB', '10percentage', '10board', '12graduation',
       '12percentage', '12board', 'CollegeID', 'CollegeTier', 'Degree',
       'Specialization', 'collegeGPA', 'CollegeCityID', 'CollegeCityTier',
       'CollegeState', 'GraduationYear', 'English', 'Logical', 'Quant',
       'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience'],
      dtype='object')
```

In [9]:
```
# Some negative values were observed in some of the columns, let's count how many times they appear for each of

columns_to_check = ['Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience']

negative_counts = {col: (amcat[col] < 0).sum() for col in columns_to_check}

for col, count in negative_counts.items():
    print("Num of -ve values in '{}': {}".format(col, count))
```

```
Num of -ve values in 'Domain': 246
Num of -ve values in 'ComputerProgramming': 868
Num of -ve values in 'ElectronicsAndSemicon': 2854
Num of -ve values in 'ComputerScience': 3096
Num of -ve values in 'MechanicalEngg': 3763
Num of -ve values in 'ElectricalEngg': 3837
Num of -ve values in 'TelecomEngg': 3624
Num of -ve values in 'CivilEngg': 3956
Num of -ve values in 'conscientiousness': 1961
Num of -ve values in 'agreeableness': 1461
Num of -ve values in 'extraversion': 1827
Num of -ve values in 'nueroticism': 2270
Num of -ve values in 'openess_to_experience': 2026
```

Observations

- The DOJ and DOB columns needs to be converted from object to the date type
- The DOL column though it contains date values would be left in the object type since it contains 'present' string values which indicates that the candidate still works at a company.
- The 'Unnamed: 0' column appears to be irrelevant fo this exploratory data analysis, and hence would need to be removed or dropped
- College City Tier and College tier are categorical columns, the data type would therefore be converted from int to object.
- There appear to be no null values (na) in any columns; however, some columns contain -1 and other negative values, which indicates that these values are not available and will be replaced with 0 instead.

This means that for such columns like 'ComputerScience' and others like it, that candidates can take only and not the other since one candidate in this case cannot belong to more than one domain or field.

For the persoanlity traits assessments, it means that there was no valid score or assessment provided for that particular trait.

Data cleaning and formatting

In [10]:
```
# Converting to date time data types

amcat['DOJ'] = pd.to_datetime(amcat['DOJ'])
```

```
amcat['DOB'] = pd.to_datetime(amcat['DOB'])

amcat.dtypes
```

Out[10]:

```
Unnamed: 0                      object
ID                               int64
Salary                         float64
DOJ                     datetime64[ns]
DOL                             object
Designation                     object
JobCity                         object
Gender                          object
DOB                     datetime64[ns]
10percentage                   float64
10board                         object
12graduation                     int64
12percentage                   float64
12board                         object
CollegeID                        int64
CollegeTier                      int64
Degree                          object
Specialization                  object
collegeGPA                     float64
CollegeCityID                    int64
CollegeCityTier                  int64
CollegeState                    object
GraduationYear                   int64
English                          int64
Logical                          int64
Quant                            int64
Domain                         float64
ComputerProgramming              int64
ElectronicsAndSemicon            int64
ComputerScience                  int64
MechanicalEngg                   int64
ElectricalEngg                   int64
TelecomEngg                      int64
CivilEngg                        int64
conscientiousness              float64
agreeableness                  float64
extraversion                   float64
nueroticism                    float64
openess_to_experience          float64
dtype: object
```

In [11]:

```
# to remove the 'Unamed: 0' column

amcat.drop(columns = ['Unnamed: 0'], inplace = True)

amcat.columns
```

Out[11]:

```
Index(['ID', 'Salary', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10percentage', '10board', '12graduation', '12percentage', '12board',
       'CollegeID', 'CollegeTier', 'Degree', 'Specialization', 'collegeGPA',
       'CollegeCityID', 'CollegeCityTier', 'CollegeState', 'GraduationYear',
       'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

In [12]:

```
# converting from int to object

amcat['CollegeTier'] = amcat['CollegeTier'].astype(object)
amcat['CollegeCityTier'] = amcat['CollegeCityTier'].astype(object)

amcat.dtypes
```

Out[12]:
```
ID                             int64
Salary                       float64
DOJ                   datetime64[ns]
DOL                            object
Designation                    object
JobCity                        object
Gender                         object
DOB                   datetime64[ns]
10percentage                 float64
10board                        object
12graduation                    int64
12percentage                 float64
12board                        object
CollegeID                       int64
CollegeTier                    object
Degree                         object
Specialization                 object
collegeGPA                   float64
CollegeCityID                   int64
CollegeCityTier                object
CollegeState                   object
GraduationYear                  int64
English                         int64
Logical                         int64
Quant                           int64
Domain                       float64
ComputerProgramming             int64
ElectronicsAndSemicon           int64
ComputerScience                 int64
MechanicalEngg                  int64
ElectricalEngg                  int64
TelecomEngg                     int64
CivilEngg                       int64
conscientiousness            float64
agreeableness                float64
extraversion                 float64
nueroticism                  float64
openess_to_experience        float64
dtype: object
```

In [13]:
```
# replacing negative values with 0

# recall the list - columns_to_check

#to replace neative values with 0 in these columns

for col in columns_to_check:
    amcat.loc[amcat[col] < 0, col] = 0

# to do the count once more
negative_counts = {col: (amcat[col] < 0).sum() for col in columns_to_check}

for col, count in negative_counts.items():
    print("Num of -ve values in '{}': {}".format(col, count))
```

```
Num of -ve values in 'Domain': 0
Num of -ve values in 'ComputerProgramming': 0
Num of -ve values in 'ElectronicsAndSemicon': 0
Num of -ve values in 'ComputerScience': 0
Num of -ve values in 'MechanicalEngg': 0
Num of -ve values in 'ElectricalEngg': 0
Num of -ve values in 'TelecomEngg': 0
Num of -ve values in 'CivilEngg': 0
Num of -ve values in 'conscientiousness': 0
Num of -ve values in 'agreeableness': 0
Num of -ve values in 'extraversion': 0
Num of -ve values in 'nueroticism': 0
Num of -ve values in 'openess_to_experience': 0
```

1. Univariate analysis

Statistical Non Visual Analysis

In [14]:
```
# first divide the dataframe into two - one for numerical and the other for categorical variables

cat_df = amcat.select_dtypes(include = ['object'])
num_df = amcat.select_dtypes(include = ['int64', 'float64'])

print(cat_df)
print(num_df)
```

```
              DOL                  Designation           JobCity Gender  \
0         present      senior quality engineer         Bangalore      f
1         present            assistant manager            Indore      m
2         present             systems engineer            Chennai      f
3         present     senior software engineer            Gurgaon      m
```

```
4      3/1/15 0:00                        get           Manesar      m
...            ...                        ...              ...    ...
3993  10/1/12 0:00           software engineer        New Delhi      m
3994   7/1/13 0:00            technical writer         Hyderabad      f
3995       present  associate software engineer         Bangalore      m
3996   1/1/15 0:00           software developer  Asifabadbanglore      f
3997       present      senior systems engineer           Chennai      f

                           10board                           12board  \
0     board ofsecondary education,ap  board of intermediate education,ap
1                               cbse                              cbse
2                               cbse                              cbse
3                               cbse                              cbse
4                               cbse                              cbse
...                              ...                               ...
3993                            cbse                              cbse
3994                    state board                       state board
3995                     bse,odisha                       chse,odisha
3996                    state board                       state board
3997                            cbse                              cbse

     CollegeTier       Degree                          Specialization  \
0              2  B.Tech/B.E.                       computer engineering
1              2  B.Tech/B.E.  electronics and communication engineering
2              2  B.Tech/B.E.                     information technology
3              1  B.Tech/B.E.                       computer engineering
4              2  B.Tech/B.E.  electronics and communication engineering
...          ...          ...                                       ...
3993           2  B.Tech/B.E.                     information technology
3994           2  B.Tech/B.E.  electronics and communication engineering
3995           2  B.Tech/B.E.                       computer engineering
3996           2  B.Tech/B.E.            computer science & engineering
3997           2  B.Tech/B.E.                     information technology

     CollegeCityTier    CollegeState
0                  0  Andhra Pradesh
1                  0  Madhya Pradesh
2                  0   Uttar Pradesh
3                  1           Delhi
4                  0   Uttar Pradesh
...              ...             ...
3993               0         Haryana
3994               1       Telangana
3995               0          Orissa
3996               1       Karnataka
3997               1      Tamil Nadu

[3998 rows x 11 columns]
          ID     Salary  10percentage  12graduation  12percentage  CollegeID  \
0     203097   420000.0         84.30          2007         95.80       1141
1     579905   500000.0         85.40          2007         85.00       5807
2     810601   325000.0         85.00          2010         68.20         64
3     267447  1100000.0         85.60          2007         83.60       6920
4     343523   200000.0         78.00          2008         76.80      11368
...      ...        ...           ...           ...           ...        ...
3993   47916   280000.0         52.09          2006         55.50       6268
3994  752781   100000.0         90.00          2009         93.00       4883
3995  355888   320000.0         81.86          2008         65.50       9786
3996  947111   200000.0         78.72          2010         69.88        979
3997  324966   400000.0         70.60          2008         68.00       6609

      collegeGPA  CollegeCityID  GraduationYear  English  ...  \
0          78.00           1141            2011      515  ...
1          70.06           5807            2012      695  ...
2          70.00             64            2014      615  ...
3          74.64           6920            2011      635  ...
4          73.90          11368            2012      545  ...
...          ...            ...             ...      ...  ...
3993       61.50           6268            2010      365  ...
3994       77.30           4883            2013      415  ...
3995       70.00           9786            2012      475  ...
3996       70.42            979            2014      450  ...
3997       68.00           6609            2012      565  ...

      ComputerScience  MechanicalEngg  ElectricalEngg  TelecomEngg  CivilEngg  \
0                   0               0               0            0          0
1                   0               0               0            0          0
2                   0               0               0            0          0
3                   0               0               0            0          0
4                   0               0               0            0          0
...               ...             ...             ...          ...        ...
3993                0               0               0            0          0
3994                0               0               0            0          0
3995                0               0               0            0          0
3996              438               0               0            0          0
3997                0               0               0            0          0

      conscientiousness  agreeableness  extraversion  nueroticism  \
0                0.9737         0.8128        0.5269      1.35490
```

```
1               0.0000         0.3789        1.2396      0.00000
2               0.2718         1.7109        0.1637      0.00000
3               0.0464         0.3448        0.0000      0.00000
4               0.0000         0.0000        0.0000      0.09163
...                ...            ...           ...          ...
3993            0.0000         0.3448        0.2366      0.64980
3994            0.0000         0.8784        0.9322      0.77980
3995            0.0000         0.0000        0.0000      0.00000
3996            0.0000         0.0459        0.0000      0.00000
3997            0.0000         0.0000        0.0000      1.32553

      openess_to_experience
0                    0.0000
1                    0.8637
2                    0.6721
3                    0.0000
4                    0.0000
...                     ...
3993                 0.0000
3994                 0.0000
3995                 0.0000
3996                 0.0000
3997                 0.0000

[3998 rows x 25 columns]
```

In [15]:
```
# second, create two functions to carry out summary statistics on this two sub data frames

# for cateorical variables
def cat_univariate_analysis(categorical_data):
    for col_name in categorical_data:
        print('Column name: ', col_name)
        print(categorical_data[col_name].agg(['count', 'nunique', 'unique']))
        print('Value counts:\n',  categorical_data[col_name].value_counts())
        print()

# for numerical variables
def num_univariate_analysis (numerical_data):
    for col_name in numerical_data:
        print('Column name: ', col_name)
        print(numerical_data[col_name].agg(['min', 'max', 'mean', 'median', 'std']))
```

In [16]:
```
# carrying out univariate analysis for the categorical data
cat_univariate_analysis (cat_df)
```

```
Column name:  DOL
count                                                  3998
nunique                                                  67
unique     [present, 3/1/15 0:00, 5/1/15 0:00, 7/1/15 0:0...
Name: DOL, dtype: object
Value counts:
 present         1875
4/1/15 0:00      573
3/1/15 0:00      124
5/1/15 0:00      112
1/1/15 0:00       99
                ...
3/1/05 0:00        1
10/1/15 0:00       1
2/1/10 0:00        1
2/1/11 0:00        1
10/1/10 0:00       1
Name: DOL, Length: 67, dtype: int64

Column name:  Designation
count                                                  3998
nunique                                                 419
unique     [senior quality engineer, assistant manager, s...
Name: Designation, dtype: object
Value counts:
 software engineer                    539
software developer                   265
system engineer                      205
programmer analyst                   139
systems engineer                     118
                                    ...
cad drafter                            1
noc engineer                           1
human resources intern                 1
senior quality assurance engineer      1
jr. software developer                 1
Name: Designation, Length: 419, dtype: int64

Column name:  JobCity
count                                                  3998
nunique                                                 339
unique     [Bangalore, Indore, Chennai, Gurgaon, Manesar,...
Name: JobCity, dtype: object
```

```
Value counts:
 Bangalore          627
-1                  461
Noida               368
Hyderabad           335
Pune                290
                   ...
Tirunelvelli          1
Ernakulam             1
Nanded                1
Dharmapuri            1
Asifabadbanglore      1
Name: JobCity, Length: 339, dtype: int64

Column name:  Gender
count        3998
nunique         2
unique     [f, m]
Name: Gender, dtype: object
Value counts:
 m    3041
f     957
Name: Gender, dtype: int64

Column name:  10board
count                                                  3998
nunique                                                 275
unique     [board ofsecondary education,ap, cbse, state b...
Name: 10board, dtype: object
Value counts:
 cbse                          1395
state board                   1164
0                              350
icse                           281
ssc                            122
                              ...
hse,orissa                       1
national public school           1
nagpur board                     1
jharkhand academic council       1
bse,odisha                       1
Name: 10board, Length: 275, dtype: int64

Column name:  12board
count                                                  3998
nunique                                                 340
unique     [board of intermediate education,ap, cbse, sta...
Name: 12board, dtype: object
Value counts:
 cbse                                1400
state board                         1254
0                                    359
icse                                 129
up board                              87
                                    ...
jawahar higher secondary school        1
nagpur board                           1
bsemp                                  1
board of higher secondary orissa       1
boardofintermediate                    1
Name: 12board, Length: 340, dtype: int64

Column name:  CollegeTier
count        3998
nunique         2
unique     [2, 1]
Name: CollegeTier, dtype: object
Value counts:
 2    3701
1     297
Name: CollegeTier, dtype: int64

Column name:  Degree
count                                                  3998
nunique                                                   4
unique     [B.Tech/B.E., MCA, M.Tech./M.E., M.Sc. (Tech.)]
Name: Degree, dtype: object
Value counts:
 B.Tech/B.E.      3700
MCA               243
M.Tech./M.E.       53
M.Sc. (Tech.)       2
Name: Degree, dtype: int64

Column name:  Specialization
count                                                  3998
nunique                                                  46
unique     [computer engineering, electronics and communi...
Name: Specialization, dtype: object
```

```
Value counts:
 electronics and communication engineering      880
computer science & engineering                 744
information technology                         660
computer engineering                           600
computer application                           244
mechanical engineering                         201
electronics and electrical engineering         196
electronics & telecommunications               121
electrical engineering                          82
electronics & instrumentation eng               32
civil engineering                               29
electronics and instrumentation engineering     27
information science engineering                 27
instrumentation and control engineering         20
electronics engineering                         19
biotechnology                                   15
other                                           13
industrial & production engineering             10
applied electronics and instrumentation          9
chemical engineering                             9
computer science and technology                  6
telecommunication engineering                    6
mechanical and automation                        5
automobile/automotive engineering                5
instrumentation engineering                      4
mechatronics                                     4
aeronautical engineering                         3
electronics and computer engineering             3
electrical and power engineering                 2
biomedical engineering                           2
information & communication technology           2
industrial engineering                           2
computer science                                 2
metallurgical engineering                        2
power systems and automation                     1
control and instrumentation engineering          1
mechanical & production engineering              1
embedded systems technology                      1
polymer technology                               1
computer and communication engineering           1
information science                              1
internal combustion engine                       1
computer networking                              1
ceramic engineering                              1
electronics                                      1
industrial & management engineering              1
Name: Specialization, dtype: int64

Column name:  CollegeCityTier
count        3998
nunique         2
unique     [0, 1]
Name: CollegeCityTier, dtype: object
Value counts:
 0    2797
1    1201
Name: CollegeCityTier, dtype: int64

Column name:  CollegeState
count                                                  3998
nunique                                                  26
unique     [Andhra Pradesh, Madhya Pradesh, Uttar Pradesh...
Name: CollegeState, dtype: object
Value counts:
 Uttar Pradesh        915
Karnataka            370
Tamil Nadu           367
Telangana            319
Maharashtra          262
Andhra Pradesh       225
West Bengal          196
Punjab               193
Madhya Pradesh       189
Haryana              180
Rajasthan            174
Orissa               172
Delhi                162
Uttarakhand          113
Kerala                33
Jharkhand             28
Chhattisgarh          27
Gujarat               24
Himachal Pradesh      16
Bihar                 10
Jammu and Kashmir      7
Assam                  5
Union Territory        5
Sikkim                 3
```

```
Meghalaya              2
Goa                    1
Name: CollegeState, dtype: int64
```

In [17]:
```
num_univariate_analysis(num_df)
```

```
Column name:  ID
min       1.124400e+04
max       1.298275e+06
mean      6.637945e+05
median    6.396000e+05
std       3.632182e+05
Name: ID, dtype: float64
Column name:  Salary
min       3.500000e+04
max       4.000000e+06
mean      3.076998e+05
median    3.000000e+05
std       2.127375e+05
Name: Salary, dtype: float64
Column name:  10percentage
min       43.000000
max       97.760000
mean      77.925443
median    79.150000
std        9.850162
Name: 10percentage, dtype: float64
Column name:  12graduation
min       1995.000000
max       2013.000000
mean      2008.087544
median    2008.000000
std          1.653599
Name: 12graduation, dtype: float64
Column name:  12percentage
min       40.000000
max       98.700000
mean      74.466366
median    74.400000
std       10.999933
Name: 12percentage, dtype: float64
Column name:  CollegeID
min           2.000000
max       18409.000000
mean       5156.851426
median     3879.000000
std        4802.261482
Name: CollegeID, dtype: float64
Column name:  collegeGPA
min        6.450000
max       99.930000
mean      71.486171
median    71.720000
std        8.167338
Name: collegeGPA, dtype: float64
Column name:  CollegeCityID
min           2.000000
max       18409.000000
mean       5156.851426
median     3879.000000
std        4802.261482
Name: CollegeCityID, dtype: float64
Column name:  GraduationYear
min          0.000000
max       2017.000000
mean      2012.105803
median    2013.000000
std         31.857271
Name: GraduationYear, dtype: float64
Column name:  English
min       180.000000
max       875.000000
mean      501.649075
median    500.000000
std       104.940021
Name: English, dtype: float64
Column name:  Logical
min       195.000000
max       795.000000
mean      501.598799
median    505.000000
std        86.783297
Name: Logical, dtype: float64
Column name:  Quant
min       120.000000
max       900.000000
mean      513.378189
median    515.000000
```

```
std       122.302332
Name: Quant, dtype: float64
Column name:  Domain
min       0.000000
max       0.999910
mean      0.572020
median    0.622643
std       0.302460
Name: Domain, dtype: float64
Column name:  ComputerProgramming
min         0.000000
max       840.000000
mean      353.319910
median    415.000000
std       204.981129
Name: ComputerProgramming, dtype: float64
Column name:  ElectronicsAndSemicon
min         0.000000
max       612.000000
mean       96.042271
median      0.000000
std       157.806602
Name: ElectronicsAndSemicon, dtype: float64
Column name:  ComputerScience
min         0.000000
max       715.000000
mean       91.516758
median      0.000000
std       174.867677
Name: ComputerScience, dtype: float64
Column name:  MechanicalEngg
min         0.000000
max       623.000000
mean       23.915958
median      0.000000
std        97.893295
Name: MechanicalEngg, dtype: float64
Column name:  ElectricalEngg
min         0.000000
max       676.000000
mean       17.438469
median      0.000000
std        87.394072
Name: ElectricalEngg, dtype: float64
Column name:  TelecomEngg
min         0.000000
max       548.000000
mean       32.757629
median      0.000000
std       104.568796
Name: TelecomEngg, dtype: float64
Column name:  CivilEngg
min         0.000000
max       516.000000
mean        3.673337
median      0.000000
std        36.559052
Name: CivilEngg, dtype: float64
Column name:  conscientiousness
min       0.000000
max       1.995300
mean      0.391062
median    0.046400
std       0.519548
Name: conscientiousness, dtype: float64
Column name:  agreeableness
min       0.000000
max       1.904800
mean      0.442923
median    0.212400
std       0.500033
Name: agreeableness, dtype: float64
Column name:  extraversion
min       0.000000
max       2.535400
mean      0.377168
median    0.091400
std       0.503572
Name: extraversion, dtype: float64
Column name:  nueroticism
min       0.000000
max       3.352500
mean      0.327201
median    0.000000
std       0.547939
Name: nueroticism, dtype: float64
Column name:  openess_to_experience
min       0.000000
max       1.822400
```

```
mean      0.302385
median    0.000000
std       0.423412
Name: openess_to_experience, dtype: float64
```

Visual Analysis

In [18]:
```
# importing libraries

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

In [19]:
```
cat_df.columns
```

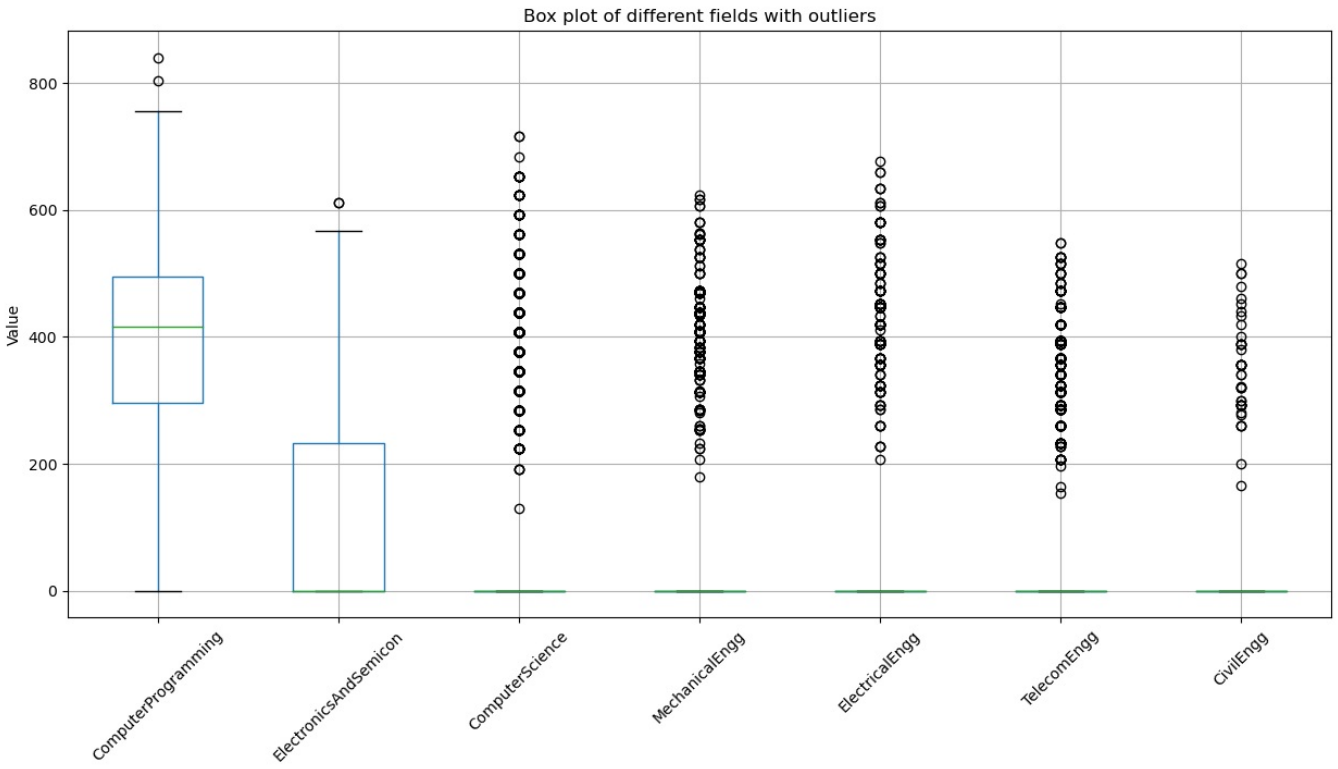
Out[19]:
```
Index(['DOL', 'Designation', 'JobCity', 'Gender', '10board', '12board',
       'CollegeTier', 'Degree', 'Specialization', 'CollegeCityTier',
       'CollegeState'],
      dtype='object')
```

In [20]:
```
num_df.columns
```

Out[20]:
```
Index(['ID', 'Salary', '10percentage', '12graduation', '12percentage',
       'CollegeID', 'collegeGPA', 'CollegeCityID', 'GraduationYear', 'English',
       'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

To understand the frequency distribution of the numerical Columns

In [21]:
```
# a plot of the various fied/ subject areas where the AMCAT tests were offered


field_columns = ['ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg']

plt.figure(figsize=(15, 7))
amcat[field_columns].boxplot()
plt.title('Box plot of different fields with outliers')
plt.ylabel('Value')
plt.xticks(rotation=45)
plt.show()
```

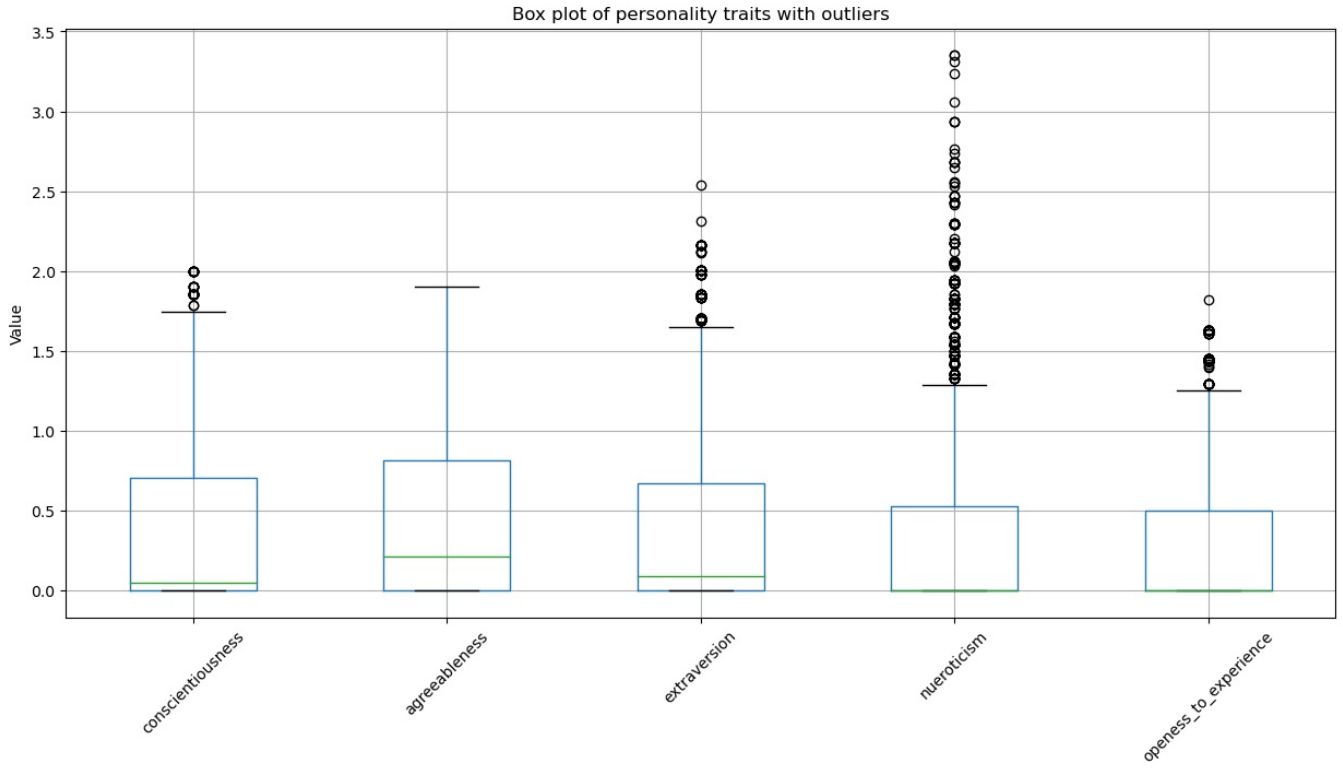


Observations:

- Computer programming has the largest spread of scores compared to the others, also boasting of the highest median value.
- The fields of Computer Science, Mechanical, Electrical, Telecom and Civil engineering have the least variability and contain considerably large number of outlier cases

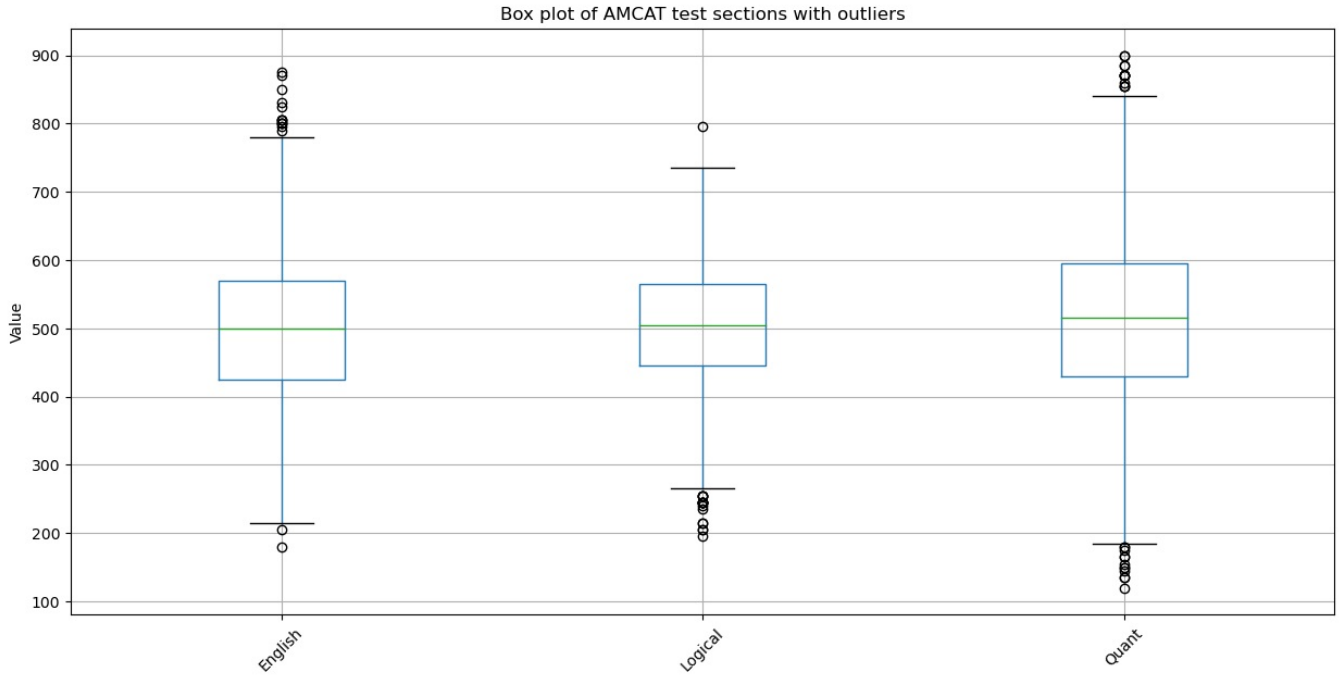
In [22]:
```
personality_traits = ['conscientiousness',
```

```
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience']

plt.figure(figsize=(15, 7))
amcat[personality_traits].boxplot()
plt.title('Box plot of personality traits with outliers')
plt.ylabel('Value')
plt.xticks(rotation=45)
plt.show()
```



This shows the distribution of personality traits scores amongst the candidates.

In [23]:
```
amcat_tests = ['English', 'Logical', 'Quant']

plt.figure(figsize=(15, 7))
amcat[amcat_tests].boxplot()
plt.title('Box plot of AMCAT test sections with outliers')
plt.ylabel('Value')
plt.xticks(rotation=45)
plt.show()
```

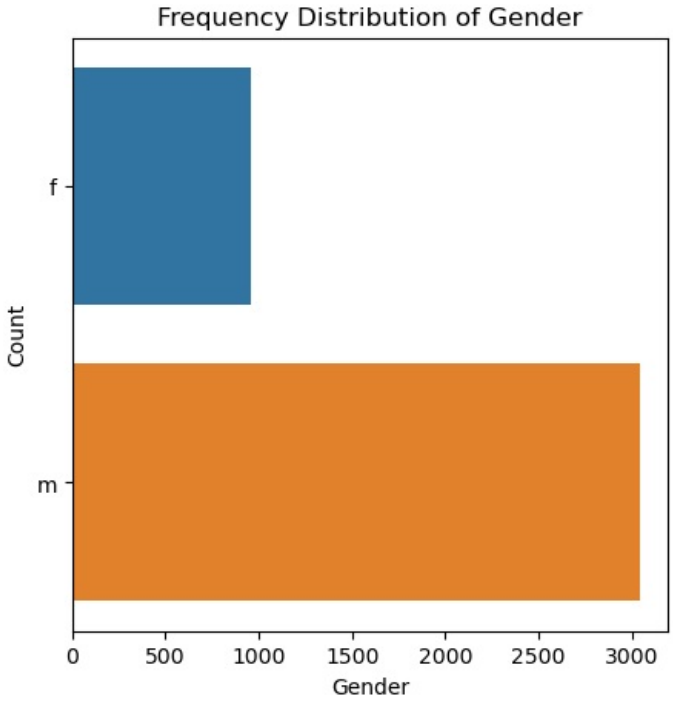


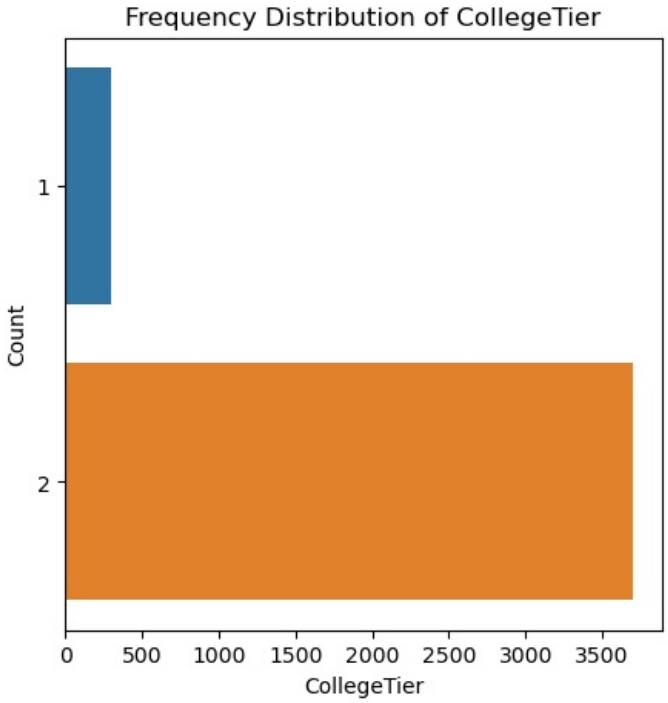
Again, this shows the distribution of scores for the various AMCAT section. The maximum score for each area is 900.

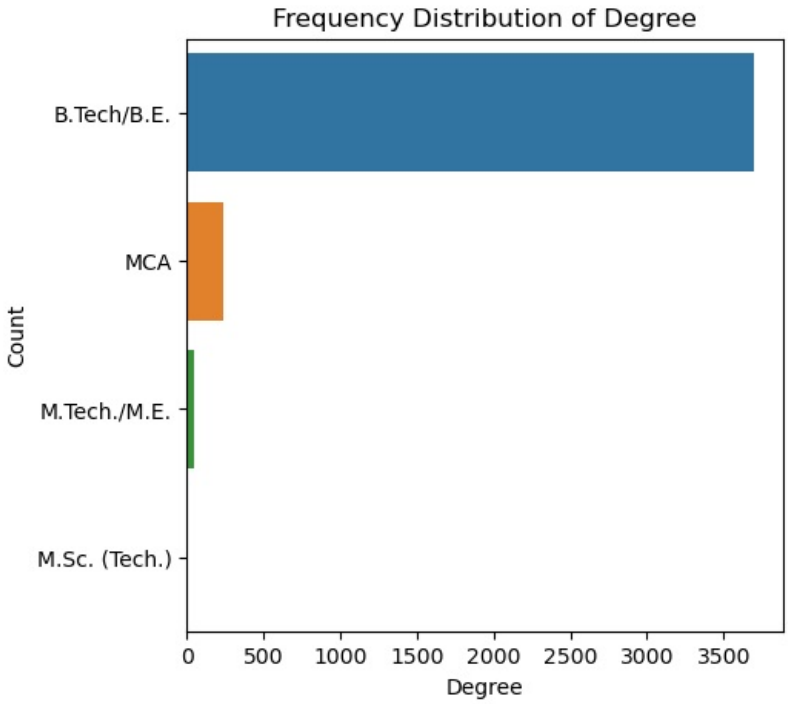
- The Quant section has the largest spread, showing a higher median value compared to others.

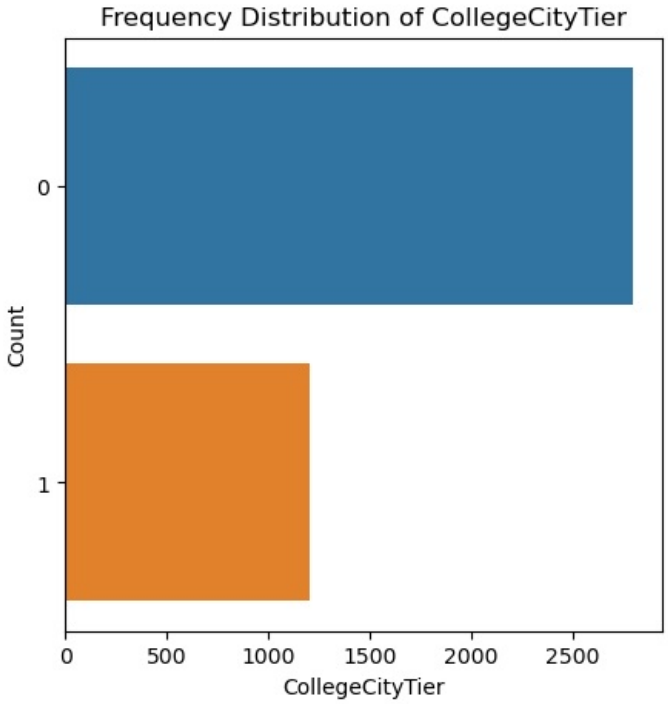
To understand the frequency distribution of specific categorical Columns

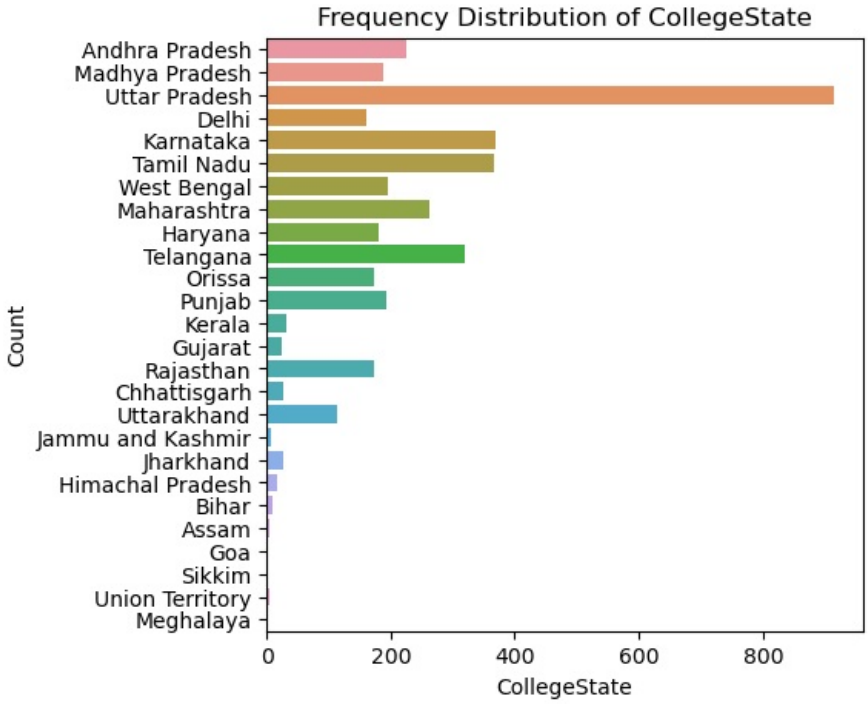
In [24]:
```
# To specify the columns I want to plot
columns_to_plot = ['Gender', 'CollegeTier', 'Degree', 'CollegeCityTier', 'CollegeState']

# Plot each column's frequency distribution
for column in columns_to_plot:
    plt.figure(figsize=(5, 5))
    sns.countplot(data=cat_df, y=column)
    plt.title(f'Frequency Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=360)
    plt.show()
```






Frequency Distribution of Degree
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)
Count
0
500
1000
1500
2000
2500
3000
3500
Degree


Frequency Distribution of CollegeCityTier
0
1
Count
0
500
1000
1500
2000
2500
CollegeCityTier



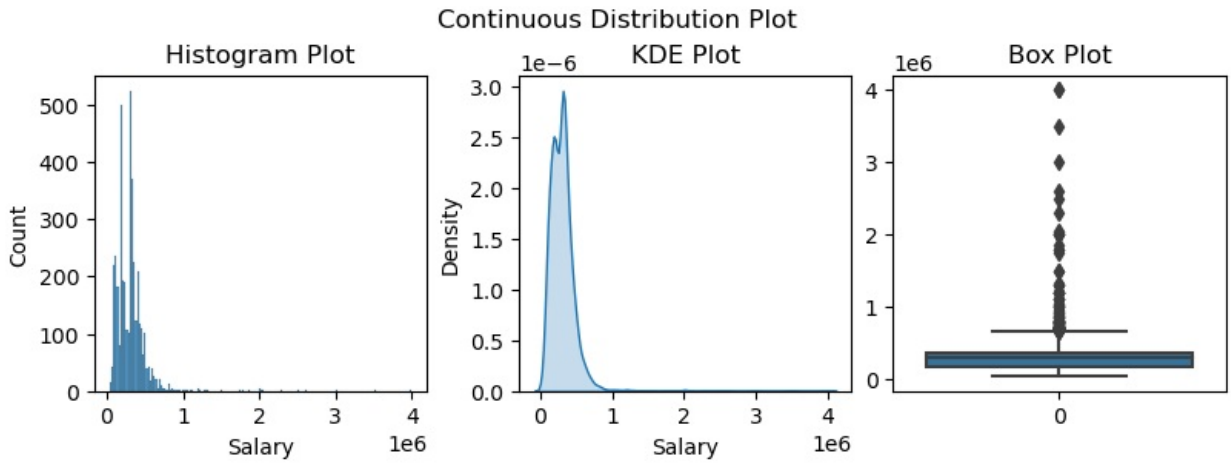
The following are observed:

- There were more male candidates who took the test from the study in 2015
- The number of candidates from College Tier 2 exceeds that of Tier 1
- Very few of the candididates have an Msc. (Tech) degree, most of them rather, have a B.Tech\B.E degree
- The College City Tier refers to the tier of the city in which the college is located. I found that most candidates are from the College city Tier tagged O while a few of them are from the tier 1.
- Most candidates attended the Colleges in Uttar Pradesh state. Following that are Karnataka and Tamil Nadu states as the next state where most candidates attended college.

In [25]:
```
# defining two functions to carry out visual analysis on the categorical and numerical columns

def cat_viz_analysis(cat_data):

    fig, ax = plt.subplots(figsize=(5, 3), constrained_layout=True)
    fig.suptitle("Discrete Distribution Plot")

    ax.set_title("Count Plot")
    sns.countplot(x=cat_data, ax=ax)

    plt.xticks(rotation=45)
    plt.show()



def num_viz_analysis(num_data):

    fig, axes = plt.subplots(1, 3, figsize=(8, 3), constrained_layout=True)
    fig.suptitle("Continuous Distribution Plot")

    axes[0].set_title("Histogram Plot")
    sns.histplot(num_data, ax=axes[0])

    axes[1].set_title("KDE Plot")
    sns.kdeplot(num_data, fill=True, ax=axes[1])

    axes[2].set_title("Box Plot")
    sns.boxplot(num_data, ax=axes[2])
```

```
    plt.show()
```

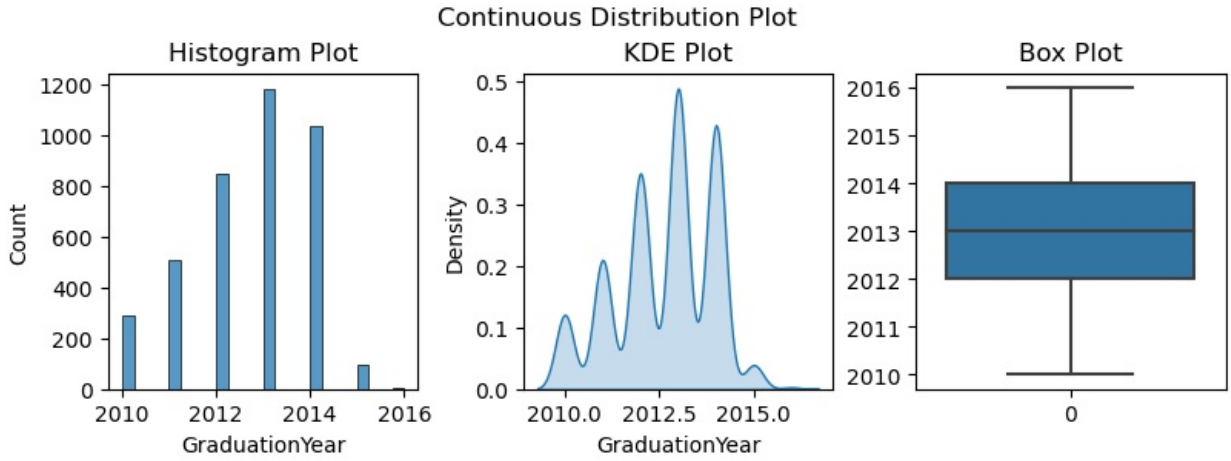
In [26]:
```
num_viz_analysis(amcat['Salary'])
```



This shows the distribution of salaray amongst the candidates.

2. Bivariate analysis

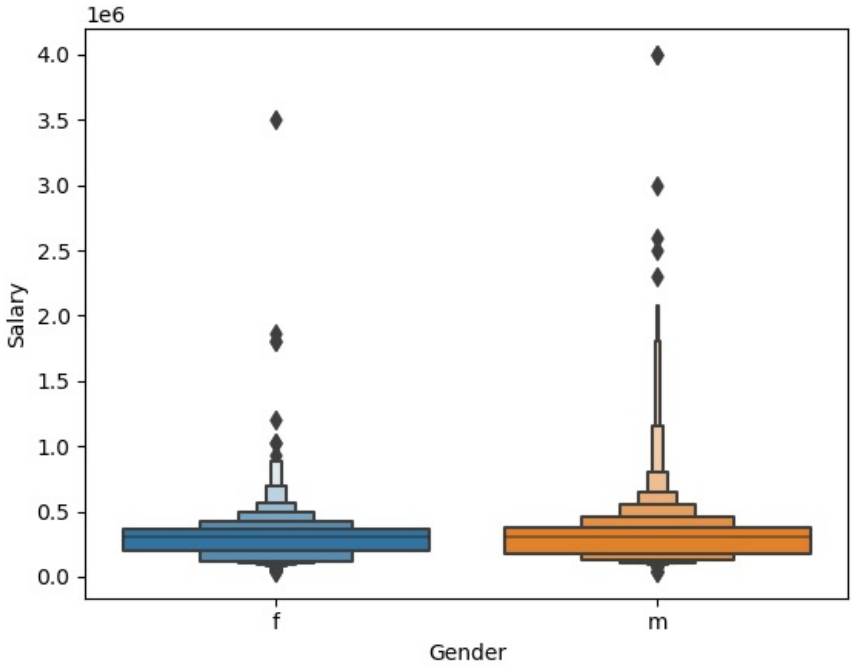
In [27]:
```
# Remove outliers for Graduation Year and perform EDA

percentile25 = amcat['GraduationYear'].quantile(0.25)
percentile75 = amcat['GraduationYear'].quantile(0.75)

#defining the thresholds
iqr=percentile75-percentile25
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr


amcat = amcat[amcat['GraduationYear'] < upper_limit]
amcat = amcat[amcat['GraduationYear'] > lower_limit]

num_viz_analysis(amcat['GraduationYear'])
```



Salary vs Gender

In [28]:
```
sns.boxenplot(data=amcat, x="Gender", y="Salary")
```

Out[28]:
```
<Axes: xlabel='Gender', ylabel='Salary'>
```

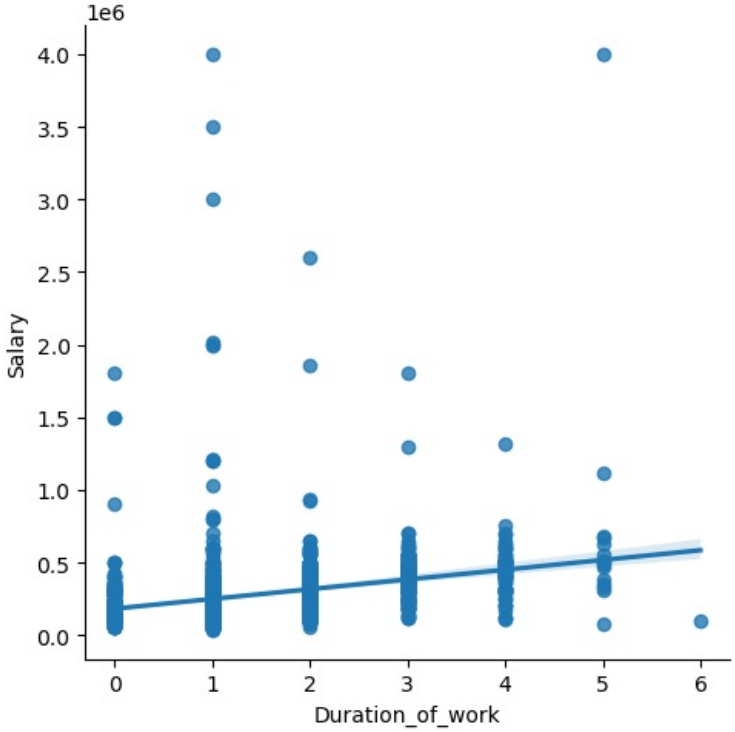


From the plot, it is seen that the distribution of salary for males is higher, which also means they earn more. But as we saw in earlier analaysis, that the number of males to females in this study is not equal. This also implies that there were more males than females in the 2015 study.

Salary vs Experience

In [29]:
```
# Does the level of experience affect the salary earned?

# convert the DOL to date time first

amcat['DOL'] = pd.to_datetime(amcat['DOL'], errors='coerce')

from datetime import datetime

today_date = datetime.today().strftime('%Y-%m-%d')
amcat['DOL'] = amcat['DOL'].replace('present', today_date)

# next, subtract the year of leabing joining a company from the year of leaving the company

amcat['Duration_of_work'] = amcat['DOL'].dt.year - amcat['DOJ'].dt.year

sns.lmplot(x="Duration_of_work", y="Salary", data=amcat)
```

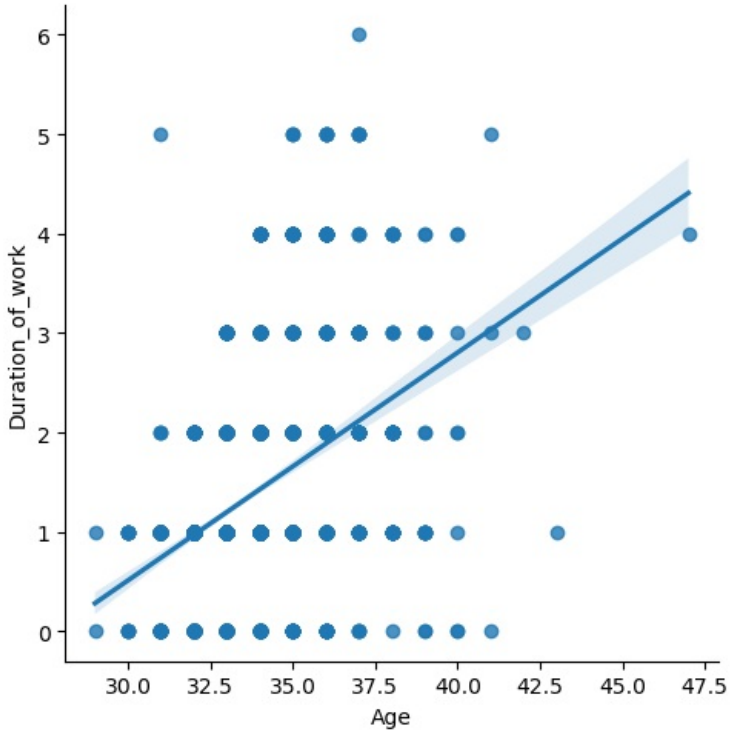
Out[29]:
```
<seaborn.axisgrid.FacetGrid at 0x24884665210>
```



Age vs Experience

In [30]:
```
amcat['Age'] = 2024 - amcat['DOB'].dt.year

sns.lmplot(x="Age", y="Duration_of_work", data=amcat)
```

Out[30]:
```
<seaborn.axisgrid.FacetGrid at 0x248858e7b50>
```



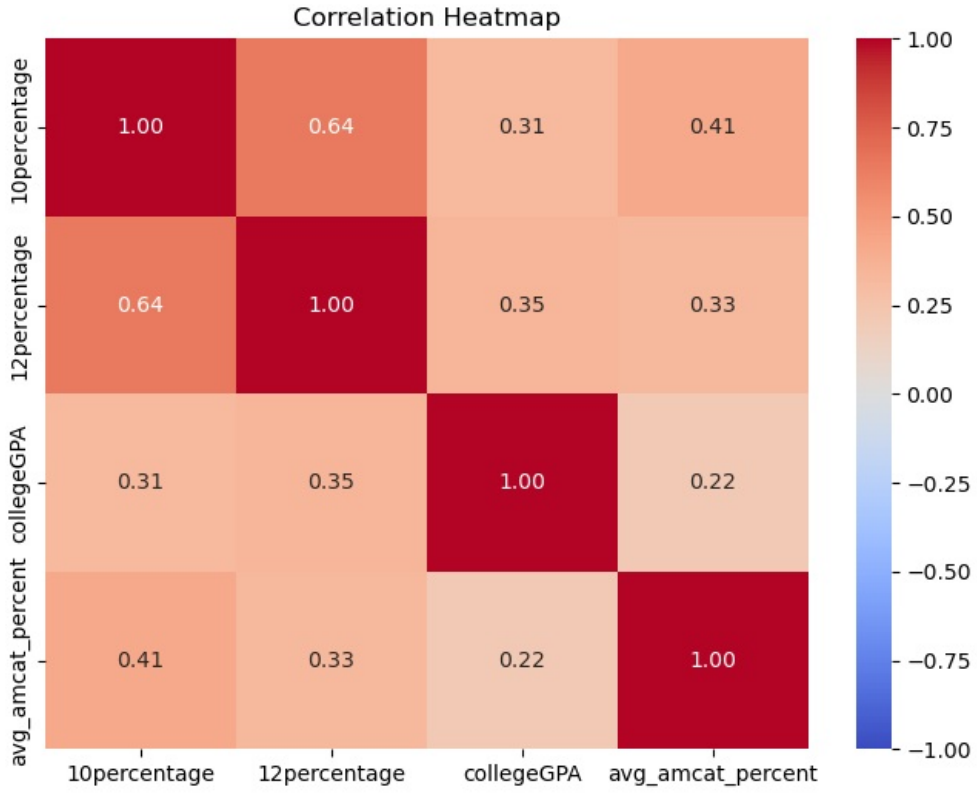
3. Research questions

Is there a correlation between college GPA and AMCAT scores?

In [31]:
```
# defining the tests available on the amcat platform

amcat_test_scores = amcat[['English','Logical','Quant','ComputerProgramming','ElectronicsAndSemicon',
                    'ComputerScience','MechanicalEngg','ElectricalEngg','TelecomEngg','CivilEngg']]

amcat_test_scores['avg'] = (amcat_test_scores['English'] + amcat_test_scores['Logical'] + amcat_test_scores['Qu
amcat_test_scores['avg_amcat_percent'] = (amcat_test_scores['avg']/900)*100
amcat_test_scores['10percentage'] = amcat['10percentage']
amcat_test_scores['12percentage'] = amcat['12percentage']
amcat_test_scores['collegeGPA'] = amcat['collegeGPA']

columns_to_plot = amcat_test_scores[['10percentage', '12percentage', 'collegeGPA', 'avg_amcat_percent']]


correlation_matrix = columns_to_plot.corr()


plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1,fmt=".2f")
plt.title("Correlation Heatmap")
```

```
plt.show()
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_14792\1914028812.py:6: 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#ret
urning-a-view-versus-a-copy
  amcat_test_scores['avg'] = (amcat_test_scores['English'] + amcat_test_scores['Logical'] + amcat_test_scores['
Quant'])/3
C:\Users\dell\AppData\Local\Temp\ipykernel_14792\1914028812.py:7: 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#ret
urning-a-view-versus-a-copy
  amcat_test_scores['avg_amcat_percent'] = (amcat_test_scores['avg']/900)*100
C:\Users\dell\AppData\Local\Temp\ipykernel_14792\1914028812.py:8: 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#ret
urning-a-view-versus-a-copy
  amcat_test_scores['10percentage'] = amcat['10percentage']
C:\Users\dell\AppData\Local\Temp\ipykernel_14792\1914028812.py:9: 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#ret
urning-a-view-versus-a-copy
  amcat_test_scores['12percentage'] = amcat['12percentage']
C:\Users\dell\AppData\Local\Temp\ipykernel_14792\1914028812.py:10: 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#ret
urning-a-view-versus-a-copy
  amcat_test_scores['collegeGPA'] = amcat['collegeGPA']
```



The correlation between collegeGPA and AMCAT scores (avg_amcat_percent) is 0.22 which is relatively small and shows very little association between the college GPA of the candidate and his/her AMCAT scores.

Salary vs Specialization - What specialization earns more salary?

In [32]:
```
# first, I would clean up the Specialization column a bit

amcat['Specialization'].unique()

# defining a function to do this easily and save time - I utilized multiple if - elif statements inside the fun

def clean_specialization(row):
    if 'computer' in row:
        return 'CSE'
```

```
    elif 'communication' in row:
        return 'ECE'
    elif 'information' in row:
        return 'IT'
    elif 'electrical' in row:
        return 'EEE'
    elif 'electro' in row:
        return 'EEE'
    elif 'telecomm' in row:
        return 'EEE'
    elif 'power' in row:
        return 'EEE'
    elif 'embedded' in row:
        return 'EEE'
    elif 'combus' in row:
        return 'Mechanical Engg'
    elif 'polymer' in row:
        return 'Chemical Engg'
    elif 'chem' in row:
        return 'Chemical Engg'
    elif 'civil' in row:
        return 'Civil Engg'
    elif 'metallurgical' in row:
        return 'Metallurgical Engg'
    elif 'instrument' in row:
        return 'Instrumentation Engg'
    elif 'mech' in row:
        return 'Mechanical Engg'
    elif 'industrial' in row:
        return 'Production Engg'
    elif 'bio' in row:
        return 'Biomedical Engineering'
    elif 'auto' in row:
        return 'Automobiles'
    elif 'aero' in row:
        return 'Aeronautical Engg'
    elif 'ceramic' in row:
        return 'Civil Engg'
    return row


amcat['Specialization'] = amcat['Specialization'].apply(clean_specialization)
```

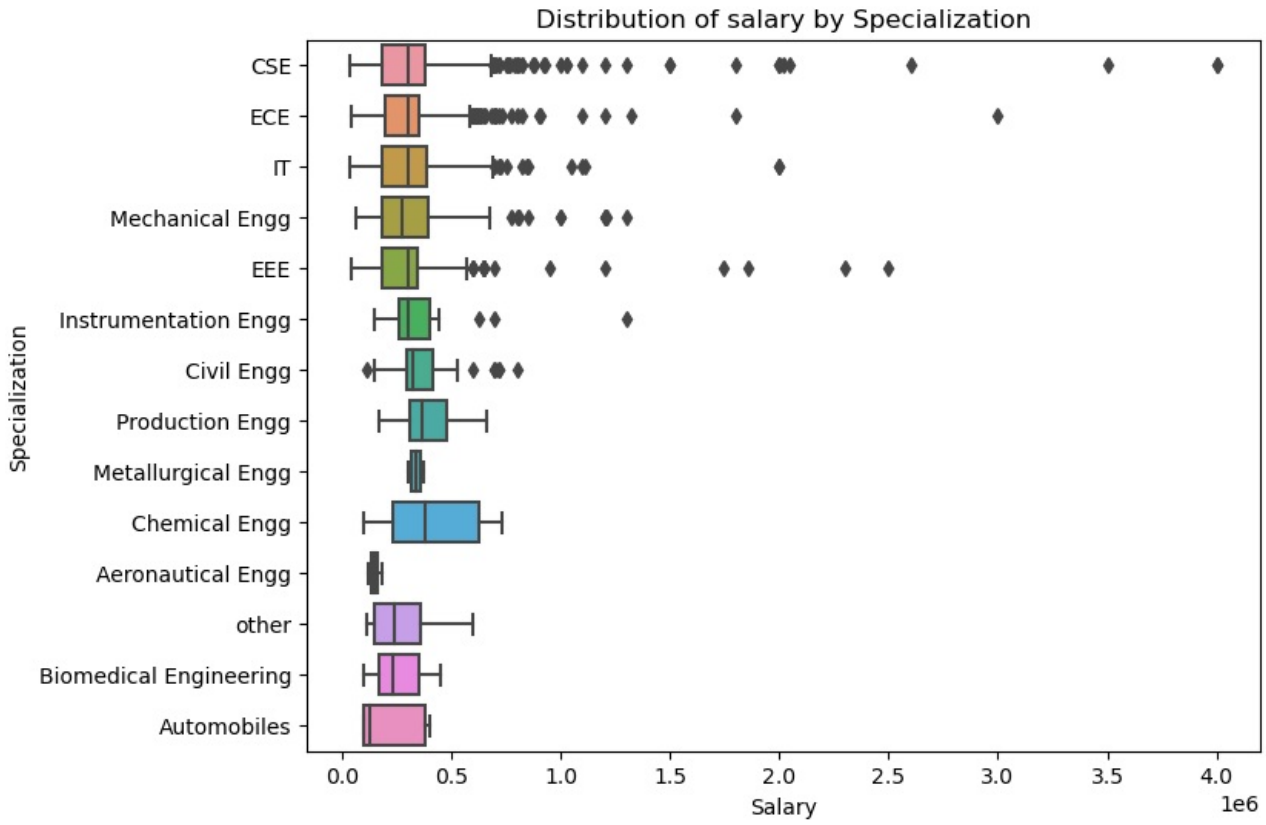
In [33]:
```
# to check if it worked
amcat['Specialization'].unique()
```

Out[33]:
```
array(['CSE', 'ECE', 'IT', 'Mechanical Engg', 'EEE',
       'Instrumentation Engg', 'Civil Engg', 'Production Engg',
       'Metallurgical Engg', 'Chemical Engg', 'Aeronautical Engg',
       'other', 'Biomedical Engineering', 'Automobiles'], dtype=object)
```

It worked!!

In [34]:
```
# to do the plot

plt.figure(figsize=(8, 6))
sns.boxplot(y='Specialization',x='Salary',data=amcat)
plt.title("Distribution of salary by Specialization")
```

Out[34]:
```
Text(0.5, 1.0, 'Distribution of salary by Specialization')
```



I made the following observations:

- Chemical engineering has a wider spread of salary range
- Computer Science (CSE) has the most outlier cases - larger salary cases comapred to other fields.
- Aeronautical engineering and Metallurgical Engineering had the least spread of salary ranges
- Production and Chemical Engineers have the highest median salary amongst the different spcializations

Is there a relationship between gender and specialization?

In [35]:
```
from scipy.stats import chi2_contingency

# Creating a contingency table
contingency_table = pd.crosstab(amcat['Specialization'], amcat['Gender'])

# Perform chi-square test for independence
chi2, p, dof, expected = chi2_contingency(contingency_table)
print("Chi-square statistic:", chi2)
print("p-value:", p)
```

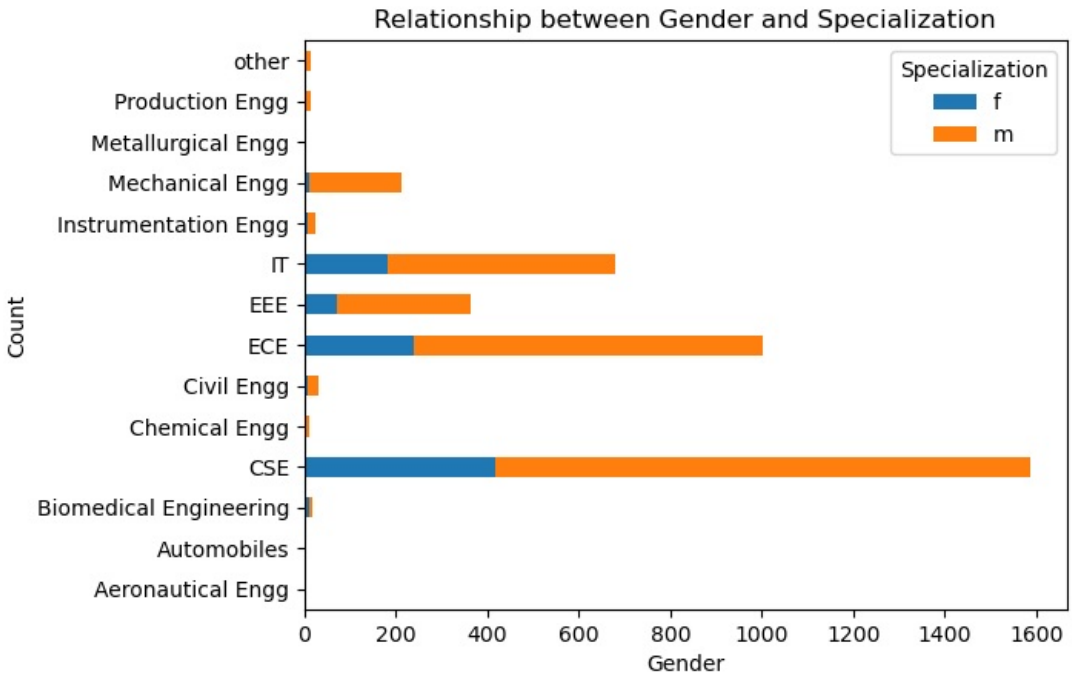
```
Chi-square statistic: 76.89814311812007
p-value: 4.2113434650609814e-11
```

Interpretation

- Since the p-value (4.21e-11) is much smaller than the typical significance level of 0.05, I reject the null hypothesis - H0: Gender and specialization are independent.
- Therefore, I conclude that there is a significant relationship between gender and specialization in the data provided

In [36]:
```
# plotting the relationship

plt.figure(figsize=(8, 6))
contingency_table.plot(kind='barh', stacked=True)
plt.title('Relationship between Gender and Specialization')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Specialization')
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



In [ ]:

In [ ]:

In [ ]: