Data description:

-The dataset was released by Aspiring Minds from the Aspiring Mind Employment Outcome 2015 (AMEO). The study is primarily limited only to students with engineering disciplines. The dataset contains the employment outcomes of engineering graduates as dependent variables (Salary, Job Titles, and Job Locations) along with the standardized scores from three different areas – cognitive skills, technical skills and personality skills. The dataset also contains demographic features. The dataset contains around 40 independent variables and 4000 data points. The independent variables are both continuous and categorical in nature. The dataset contains a unique identifier for each candidate. Below mentioned table contains the details for the original dataset.

In [ ]:
```
! pip install pandas
! pip install numpy
! pip install matplotlib.pyplot
! pip install seaborn
```

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [8]:
```
df = pd.read_csv('C:/Users/yuvas/OneDrive/Desktop/innomatics/data.xlsx - Sheet1.
pd.set_option('display.max_columns', None)
df1= df.copy()
df1.head()
```

Out[8]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	D
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19 (
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4 (
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3 (
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5 (
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27 (

In [9]:
```
df1.shape
```

Out[9]: (3998, 39)

In [10]:
```
df1.columns
```

Out[10]:
```
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')
```

Data Cleaning

1.Remove unnecessary columns

In [11]:
```
df1.drop(columns= ['Unnamed: 0', 'ID', 'CollegeID', 'CollegeCityID'], axis = 1,
df1.shape
```

Out[11]: (3998, 35)

In [13]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 35 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Salary                 3998 non-null   float64
 1   DOJ                    3998 non-null   object 
 2   DOL                    3998 non-null   object 
 3   Designation            3998 non-null   object 
 4   JobCity                3998 non-null   object 
 5   Gender                 3998 non-null   object 
 6   DOB                    3998 non-null   object 
 7   10percentage           3998 non-null   float64
 8   10board                3998 non-null   object 
 9   12graduation           3998 non-null   int64  
 10  12percentage           3998 non-null   float64
 11  12board                3998 non-null   object 
 12  CollegeTier            3998 non-null   int64  
 13  Degree                 3998 non-null   object 
 14  Specialization         3998 non-null   object 
 15  collegeGPA             3998 non-null   float64
 16  CollegeCityTier        3998 non-null   int64  
 17  CollegeState           3998 non-null   object 
 18  GraduationYear         3998 non-null   int64  
 19  English                3998 non-null   int64  
 20  Logical                3998 non-null   int64  
 21  Quant                  3998 non-null   int64  
 22  Domain                 3998 non-null   float64
 23  ComputerProgramming    3998 non-null   int64  
 24  ElectronicsAndSemicon  3998 non-null   int64  
 25  ComputerScience        3998 non-null   int64  
 26  MechanicalEngg         3998 non-null   int64  
 27  ElectricalEngg         3998 non-null   int64  
 28  TelecomEngg            3998 non-null   int64  
 29  CivilEngg              3998 non-null   int64  
 30  conscientiousness      3998 non-null   float64
 31  agreeableness          3998 non-null   float64
 32  extraversion           3998 non-null   float64
 33  nueroticism            3998 non-null   float64
 34  openess_to_experience  3998 non-null   float64
dtypes: float64(10), int64(14), object(11)
memory usage: 1.1+ MB
```

In [ ]:

2.Searching for nulls and duplicates in the dataset

In [17]:

```
df1.isnull().sum()
```

Out[17]:

```
Salary                    0
DOJ                       0
DOL                       0
Designation               0
JobCity                   0
Gender                    0
DOB                       0
10percentage              0
10board                   0
12graduation              0
12percentage              0
12board                   0
CollegeTier               0
Degree                    0
Specialization            0
collegeGPA                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 [18]:

```
df1.duplicated().sum()
```

Out[18]:

```
0
```

Hence no nulls and duplicates found in the dataset.

In [ ]:

3.Formatting to Correct DATE data types

```
-ASSUMPTION: DOL column has "present" for some people as on
2015(because the data is upto 2015 only). So, for our analysis
purpose we assumed that for such people their DOL is the end of
the year which is 31/12/2015. So, we need to replace "present"
with the assumed date.
-After that we need to convert DOL,DOJ,DOB columns to datetime
data type.
```

In [20]:

```
df1['DOL'] =df1['DOL'].replace('present', '12/31/15')
df1['DOL'].head()
```

```
Out[20]: 0        12/31/15
         1        12/31/15
         2        12/31/15
         3        12/31/15
         4     3/1/15 0:00
         Name: DOL, dtype: object
```

```
In [22]: df1[['DOJ', 'DOL', 'DOB']]= df1[['DOJ', 'DOL', 'DOB']].apply(pd.to_datetime, for
         df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 35 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Salary                 3998 non-null   float64
 1   DOJ                    3998 non-null   datetime64[ns]
 2   DOL                    3998 non-null   datetime64[ns]
 3   Designation            3998 non-null   object
 4   JobCity                3998 non-null   object
 5   Gender                 3998 non-null   object
 6   DOB                    3998 non-null   datetime64[ns]
 7   10percentage           3998 non-null   float64
 8   10board                3998 non-null   object
 9   12graduation           3998 non-null   int64
 10  12percentage           3998 non-null   float64
 11  12board                3998 non-null   object
 12  CollegeTier            3998 non-null   int64
 13  Degree                 3998 non-null   object
 14  Specialization         3998 non-null   object
 15  collegeGPA             3998 non-null   float64
 16  CollegeCityTier        3998 non-null   int64
 17  CollegeState           3998 non-null   object
 18  GraduationYear         3998 non-null   int64
 19  English                3998 non-null   int64
 20  Logical                3998 non-null   int64
 21  Quant                  3998 non-null   int64
 22  Domain                 3998 non-null   float64
 23  ComputerProgramming    3998 non-null   int64
 24  ElectronicsAndSemicon  3998 non-null   int64
 25  ComputerScience        3998 non-null   int64
 26  MechanicalEngg         3998 non-null   int64
 27  ElectricalEngg         3998 non-null   int64
 28  TelecomEngg            3998 non-null   int64
 29  CivilEngg              3998 non-null   int64
 30  conscientiousness      3998 non-null   float64
 31  agreeableness          3998 non-null   float64
 32  extraversion           3998 non-null   float64
 33  nueroticism            3998 non-null   float64
 34  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](3), float64(10), int64(14), object(8)
memory usage: 1.1+ MB
```

```
In [23]: df1[['DOJ', 'DOL', 'DOB']].sample(5)
```

Out[23]:

	DOJ	DOL	DOB
1332	2014-06-01	2015-12-31	1992-12-16
1803	2014-08-01	2015-12-31	1993-05-20
935	2012-09-01	2013-08-01	1990-10-24
2583	2013-01-01	2013-12-01	1990-10-24
3482	2011-01-01	2015-12-31	1988-08-05

Now you can see the corrected dates in proper formats and datatypes.

In [ ]:

4.Checking if DOL (Date of leaving) earlier than DOJ (Date of joining).

In [28]:
```
error_dates = df1[(df1['DOL'] < df1['DOJ'])]
print('Number of records where DOL is earlier than DOJ: ', error_dates.shape[0])
```

```
Number of records where DOL is earlier than DOJ:  40
```

Treatment: Need to drop these 40 records to avoid bias during our analysis on the dataset.

In [31]:
```
df1.drop(error_dates.index, inplace= True)
df1.shape
```

Out[31]:
```
(3958, 35)
```

There are total 3998 rows and after removing the 40 records, now it becomes 3958 records/rows (remaining).

In [ ]:

NOW THE DATASET IS READY FOR EDA

1.Univariate Analysis - Statistical Non Visual Analysis

In [36]:
```
categorical_df = df1.select_dtypes(include =['object'])
numerical_df = df1.select_dtypes(include=['int64', 'float64'])
```

In [37]:
```
def categorical_univariate_analysis(categorical_data):
    for col_name in categorical_data:
        print("*"*25, col_name, "*"*25)
        print(categorical_data[col_name].agg(['count', 'nunique', 'unique']))
        print('value Counts: \n', categorical_data[col_name].value_counts())
        print()
```

In [38]:
```
categorical_univariate_analysis(categorical_df)
```

```
************************* Designation *************************
count                                                   3958
nunique                                                  417
unique     [senior quality engineer, assistant manager, s...
Name: Designation, dtype: object
value Counts:
 Designation
software engineer                  537
software developer                 263
system engineer                    203
programmer analyst                 139
systems engineer                   117
                                  ...
human resources intern               1
senior quality assurance engineer    1
clerical assistant                   1
delivery software engineer           1
jr. software developer               1
Name: count, Length: 417, dtype: int64

************************* JobCity *************************
count                                                   3958
nunique                                                  338
unique     [Bangalore, Indore, Chennai, Gurgaon, Manesar,...
Name: JobCity, dtype: object
value Counts:
 JobCity
Bangalore           626
-1                  449
Noida               362
Hyderabad           331
Pune                287
                   ...
Asansol               1
Tirunelvelli          1
Ernakulam             1
Nanded                1
Asifabadbanglore      1
Name: count, Length: 338, dtype: int64

************************* Gender *************************
count        3958
nunique         2
unique     [f, m]
Name: Gender, dtype: object
value Counts:
 Gender
m    3009
f     949
Name: count, dtype: int64

************************* 10board *************************
count                                                   3958
nunique                                                  275
unique     [board ofsecondary education,ap, cbse, state b...
Name: 10board, dtype: object
value Counts:
 10board
cbse                          1379
state board                   1146
```

```
0                              350
icse                           278
ssc                            122
                              ...
hse,orissa                       1
national public school           1
nagpur board                     1
jharkhand academic council       1
bse,odisha                       1
Name: count, Length: 275, dtype: int64

************************* 12board *************************
count                                                    3958
nunique                                                   340
unique     [board of intermediate education,ap, cbse, sta...
Name: 12board, dtype: object
value Counts:
 12board
cbse                                    1383
state board                             1235
0                                        359
icse                                     128
up board                                  87
                                        ...
jawahar higher secondary school            1
nagpur board                               1
bsemp                                      1
board of higher secondary orissa           1
boardofintermediate                        1
Name: count, Length: 340, dtype: int64

************************* Degree *************************
count                                                  3958
nunique                                                   4
unique     [B.Tech/B.E., MCA, M.Tech./M.E., M.Sc. (Tech.)]
Name: Degree, dtype: object
value Counts:
 Degree
B.Tech/B.E.      3663
MCA               240
M.Tech./M.E.       53
M.Sc. (Tech.)       2
Name: count, dtype: int64

************************* Specialization *************************
count                                                    3958
nunique                                                    46
unique     [computer engineering, electronics and communi...
Name: Specialization, dtype: object
value Counts:
 Specialization
electronics and communication engineering      867
computer science & engineering                 735
information technology                         657
computer engineering                           596
computer application                           241
mechanical engineering                         201
electronics and electrical engineering         192
electronics & telecommunications               120
electrical engineering                          80
```

```
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                             8
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: count, dtype: int64

************************* CollegeState *************************
count                                                   3958
nunique                                                   26
unique     [Andhra Pradesh, Madhya Pradesh, Uttar Pradesh...
Name: CollegeState, dtype: object
value Counts:
 CollegeState
Uttar Pradesh       905
Karnataka           369
Tamil Nadu          363
Telangana           314
Maharashtra         260
Andhra Pradesh      223
West Bengal         195
Madhya Pradesh      189
Punjab              189
Haryana             177
Orissa              172
Rajasthan           170
Delhi               161
Uttarakhand         112
```

```
Kerala                 33
Jharkhand              27
Chhattisgarh           27
Gujarat                24
Himachal Pradesh       16
Bihar                  10
Jammu and Kashmir       7
Assam                   5
Union Territory         5
Sikkim                  3
Goa                     1
Meghalaya               1
Name: count, dtype: int64
```

Observations:

```
- There are 417 different designations, among them Software
Engineer is the most occupied designation/position.
- Out of 338 different job cities, Banglore provided the most
number of jobs.
- Males occupied the jobs 3 times more compared to females.
- Most of the people had cbse as their 10 and 12 board.
- People from B.Tech/ B.E with specialization in electronics and
communication engineering received more jobs.
- Uttar Pradesh college students grabs more job opportunities
compared to students from other state colleges.
```

In [39]:
```python
numerical_df.columns
```

Out[39]:
```
Index(['Salary', '10percentage', '12graduation', '12percentage', 'CollegeTier',
       'collegeGPA', 'CollegeCityTier', 'GraduationYear', 'English', 'Logical',
       'Quant', 'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience'],
      dtype='object')
```

In [40]:
```python
numerical_df.shape
```

Out[40]:
```
(3958, 24)
```

In [41]:
```python
numerical_df.nunique()
```

```
Out[41]: Salary                    176
         10percentage              844
         12graduation               16
         12percentage              797
         CollegeTier                 2
         collegeGPA               1275
         CollegeCityTier             2
         GraduationYear             11
         English                   111
         Logical                   107
         Quant                     138
         Domain                    243
         ComputerProgramming        79
         ElectronicsAndSemicon      29
         ComputerScience            20
         MechanicalEngg             42
         ElectricalEngg             31
         TelecomEngg                26
         CivilEngg                  23
         conscientiousness         141
         agreeableness             148
         extraversion              153
         nueroticism               217
         openess_to_experience     141
         dtype: int64
```

```
In [42]: discrete_num_col = ['12graduation', 'CollegeTier', 'CollegeCityTier', 'Graduatic
         numerical_df.drop(columns =discrete_num_col, axis=1, inplace=True)
```

```
In [43]: numerical_df.columns
```

```
Out[43]: Index(['Salary', '10percentage', '12percentage', 'collegeGPA', 'English',
                'Logical', 'Quant', 'Domain', 'ComputerProgramming',
                'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
                'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
                'agreeableness', 'extraversion', 'nueroticism',
                'openess_to_experience'],
               dtype='object')
```

```
In [44]: numerical_df.shape
```

```
Out[44]: (3958, 20)
```

```
In [45]: discrete_num_df = df1[discrete_num_col]
         discrete_num_df.columns
```

```
Out[45]: Index(['12graduation', 'CollegeTier', 'CollegeCityTier', 'GraduationYear'], dty
         pe='object')
```

```
In [46]: def discrete_univariate_analysis(discrete_data):
             for col_name in discrete_data:
                 print("*"*25, col_name, "*"*25)
                 print(discrete_data[col_name].agg(['count', 'nunique', 'unique']))
                 print('value Counts: \n', discrete_data[col_name].value_counts())
                 print()
```

```
In [47]: discrete_univariate_analysis(discrete_num_df)
```

```
************************* 12graduation *************************
count                                                   3958
nunique                                                   16
unique     [2007, 2010, 2008, 2009, 2006, 2011, 2005, 199...
Name: 12graduation, dtype: object
value Counts:
 12graduation
2009    1037
2008     925
2010     730
2007     526
2006     406
2005     160
2004      73
2011      46
2003      25
2002      14
2012      10
2001       2
1995       1
1998       1
2013       1
1999       1
Name: count, dtype: int64

************************* CollegeTier *************************
count        3958
nunique         2
unique     [2, 1]
Name: CollegeTier, dtype: object
value Counts:
 CollegeTier
2    3662
1     296
Name: count, dtype: int64

************************* CollegeCityTier *************************
count        3958
nunique         2
unique     [0, 1]
Name: CollegeCityTier, dtype: object
value Counts:
 CollegeCityTier
0    2769
1    1189
Name: count, dtype: int64

************************* GraduationYear *************************
count                                                   3958
nunique                                                   11
unique     [2011, 2012, 2014, 2016, 2013, 2010, 2015, 200...
Name: GraduationYear, dtype: object
value Counts:
 GraduationYear
2013    1165
2014    1018
2012     842
2011     507
2010     291
2015      94
```

```
2009      24
2017       8
2016       7
0          1
2007       1
Name: count, dtype: int64
```

Observations:

```
- People graduated in 2013 grab more job opportunities.
- People from CollegeCityTier 0 and CollegeTier 2 received more
jobs.
```

In [48]:
```
def numerical_univariate_analysis(numeric_data):
    for col_name in numeric_data:
        print('*'*15, col_name, '*'*15)
        print(numeric_data[col_name].agg(['min','max','mean','median','std']))
        print()
```

In [49]:
```
numerical_univariate_analysis(numerical_df)
```

```
*************** Salary ***************
min       3.500000e+04
max       4.000000e+06
mean      3.084901e+05
median    3.000000e+05
std       2.122649e+05
Name: Salary, dtype: float64

*************** 10percentage ***************
min       43.000000
max       97.760000
mean      77.948633
median    79.200000
std        9.845458
Name: 10percentage, dtype: float64

*************** 12percentage ***************
min       40.000000
max       98.700000
mean      74.475950
median    74.400000
std       11.004427
Name: 12percentage, dtype: float64

*************** collegeGPA ***************
min        6.450000
max       99.930000
mean      71.491127
median    71.710000
std        8.184509
Name: collegeGPA, dtype: float64

*************** English ***************
min       180.000000
max       875.000000
mean      501.592471
median    500.000000
std       104.935965
Name: English, dtype: float64

*************** Logical ***************
min       195.000000
max       795.000000
mean      501.607377
median    505.000000
std        86.769459
Name: Logical, dtype: float64

*************** Quant ***************
min       120.000000
max       900.000000
mean      513.359272
median    515.000000
std       122.227017
Name: Quant, dtype: float64

*************** Domain ***************
min      -1.000000
max       0.999910
mean      0.512228
```

```
median    0.622643
std       0.467977
Name: Domain, dtype: float64

*************** ComputerProgramming ***************
min        -1.000000
max       840.000000
mean      352.919404
median    415.000000
std       205.805997
Name: ComputerProgramming, dtype: float64

*************** ElectronicsAndSemicon ***************
min        -1.000000
max       612.000000
mean       95.219303
median     -1.000000
std       158.331531
Name: ElectronicsAndSemicon, dtype: float64

*************** ComputerScience ***************
min        -1.000000
max       715.000000
mean       90.651592
median     -1.000000
std       175.333064
Name: ComputerScience, dtype: float64

*************** MechanicalEngg ***************
min        -1.000000
max       623.000000
mean       23.217029
median     -1.000000
std        98.588251
Name: MechanicalEngg, dtype: float64

*************** ElectricalEngg ***************
min        -1.000000
max       676.000000
mean       16.086155
median     -1.000000
std        86.623399
Name: ElectricalEngg, dtype: float64

*************** TelecomEngg ***************
min        -1.000000
max       548.000000
mean       32.004295
median     -1.000000
std       105.080590
Name: TelecomEngg, dtype: float64

*************** CivilEngg ***************
min        -1.000000
max       516.000000
mean        2.721071
median     -1.000000
std        36.841443
Name: CivilEngg, dtype: float64
```

```
*************** conscientiousness ***************
min      -4.126700
max       1.995300
mean     -0.040113
median    0.046400
std       1.027492
Name: conscientiousness, dtype: float64

*************** agreeableness ***************
min      -5.781600
max       1.904800
mean      0.144828
median    0.212400
std       0.942313
Name: agreeableness, dtype: float64

*************** extraversion ***************
min      -4.600900
max       2.535400
mean     -0.001104
median    0.091400
std       0.952860
Name: extraversion, dtype: float64

*************** nueroticism ***************
min      -2.643000
max       3.352500
mean     -0.169251
median   -0.234400
std       1.006326
Name: nueroticism, dtype: float64

*************** openess_to_experience ***************
min      -7.375700
max       1.822400
mean     -0.140694
median   -0.094300
std       1.007413
Name: openess_to_experience, dtype: float64
```

2. Univariate - Visual Analysis

In [50]:
```
df1.shape
```

Out[50]:
```
(3958, 35)
```

In [52]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3958 entries, 0 to 3997
Data columns (total 35 columns):
 #   Column                 Non-Null Count  Dtype         
---  ------                 --------------  -----         
 0   Salary                 3958 non-null   float64       
 1   DOJ                    3958 non-null   datetime64[ns]
 2   DOL                    3958 non-null   datetime64[ns]
 3   Designation            3958 non-null   object        
 4   JobCity                3958 non-null   object        
 5   Gender                 3958 non-null   object        
 6   DOB                    3958 non-null   datetime64[ns]
 7   10percentage           3958 non-null   float64       
 8   10board                3958 non-null   object        
 9   12graduation           3958 non-null   int64         
 10  12percentage           3958 non-null   float64       
 11  12board                3958 non-null   object        
 12  CollegeTier            3958 non-null   int64         
 13  Degree                 3958 non-null   object        
 14  Specialization         3958 non-null   object        
 15  collegeGPA             3958 non-null   float64       
 16  CollegeCityTier        3958 non-null   int64         
 17  CollegeState           3958 non-null   object        
 18  GraduationYear         3958 non-null   int64         
 19  English                3958 non-null   int64         
 20  Logical                3958 non-null   int64         
 21  Quant                  3958 non-null   int64         
 22  Domain                 3958 non-null   float64       
 23  ComputerProgramming    3958 non-null   int64         
 24  ElectronicsAndSemicon  3958 non-null   int64         
 25  ComputerScience        3958 non-null   int64         
 26  MechanicalEngg         3958 non-null   int64         
 27  ElectricalEngg         3958 non-null   int64         
 28  TelecomEngg            3958 non-null   int64         
 29  CivilEngg              3958 non-null   int64         
 30  conscientiousness      3958 non-null   float64       
 31  agreeableness          3958 non-null   float64       
 32  extraversion           3958 non-null   float64       
 33  nueroticism            3958 non-null   float64       
 34  openess_to_experience  3958 non-null   float64       
dtypes: datetime64[ns](3), float64(10), int64(14), object(8)
memory usage: 1.1+ MB
```

2.1) Salary

In [61]:
```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['Salary'], bins=20, kde=True)
plt.xlabel('Salary')
plt.ylabel('Density')
plt.title('Salary Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='Salary', data=df1)
plt.xlabel('Salary')
plt.title('Salary Outliers')
plt.show()
```

Observations:

Histogram and PDF

```
- Most of the peoples salaries falls under 0.5 million(5 lakhs).
- As salaries increase beyond this range, the frequency of
individuals decreases significantly.
- The distribution is right-skewed, indicating that there are
relatively fewer high earners compared to the middle-income
group.
```

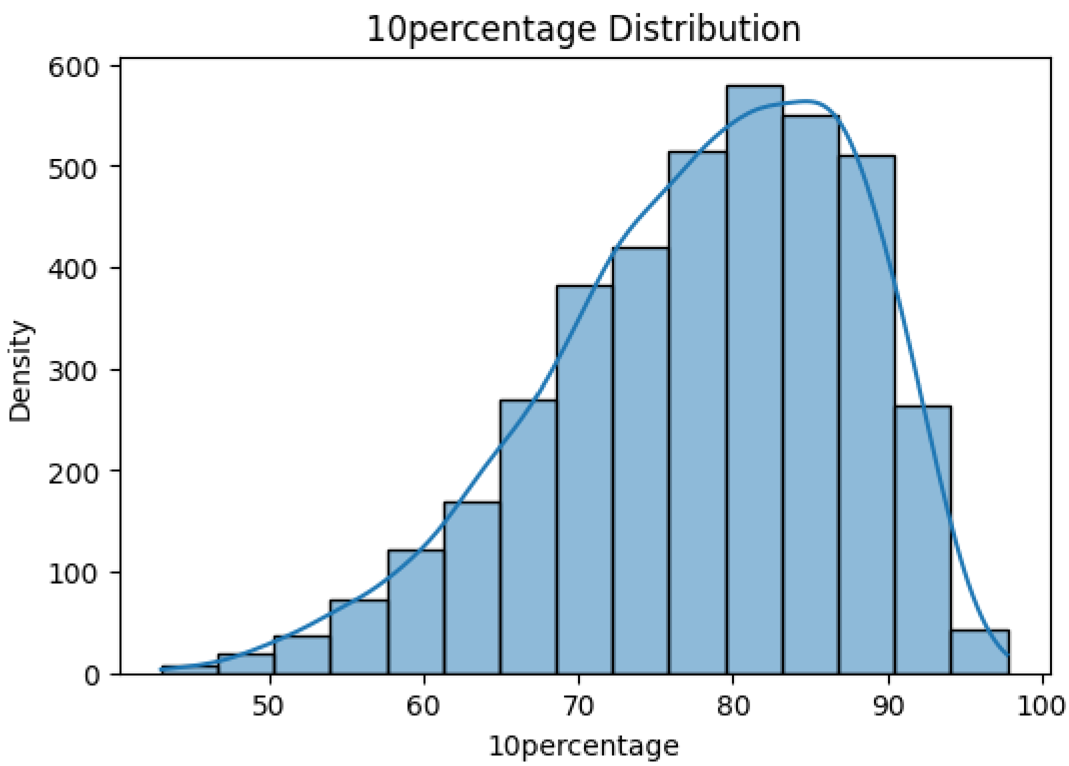
Box plot

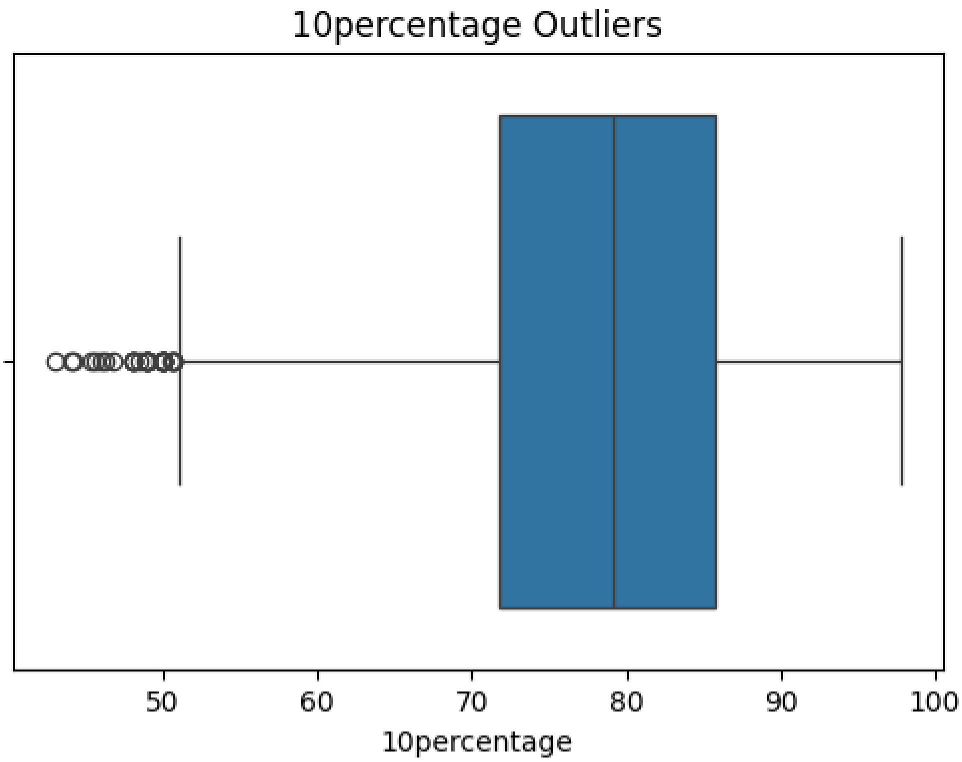
    - The presence of outliers suggests that there are a few high
    earners.

2.2) 10percentage

In [65]:
```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['10percentage'], bins=15, kde=True)
plt.xlabel('10percentage')
plt.ylabel('Density')
plt.title('10percentage Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='10percentage', data=df1)
plt.xlabel('10percentage')
plt.title('10percentage Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 50% during their 10th class
and majority have scored in between 75%-90%.
- 80% is where the peak touched.
```

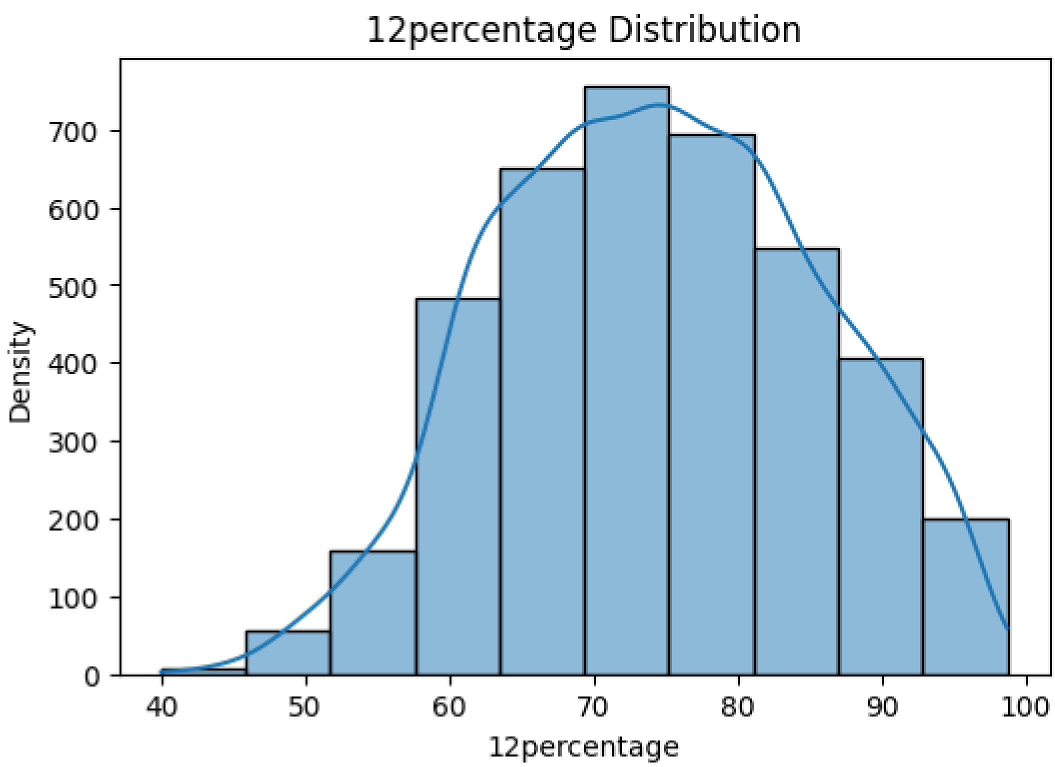
Box plot

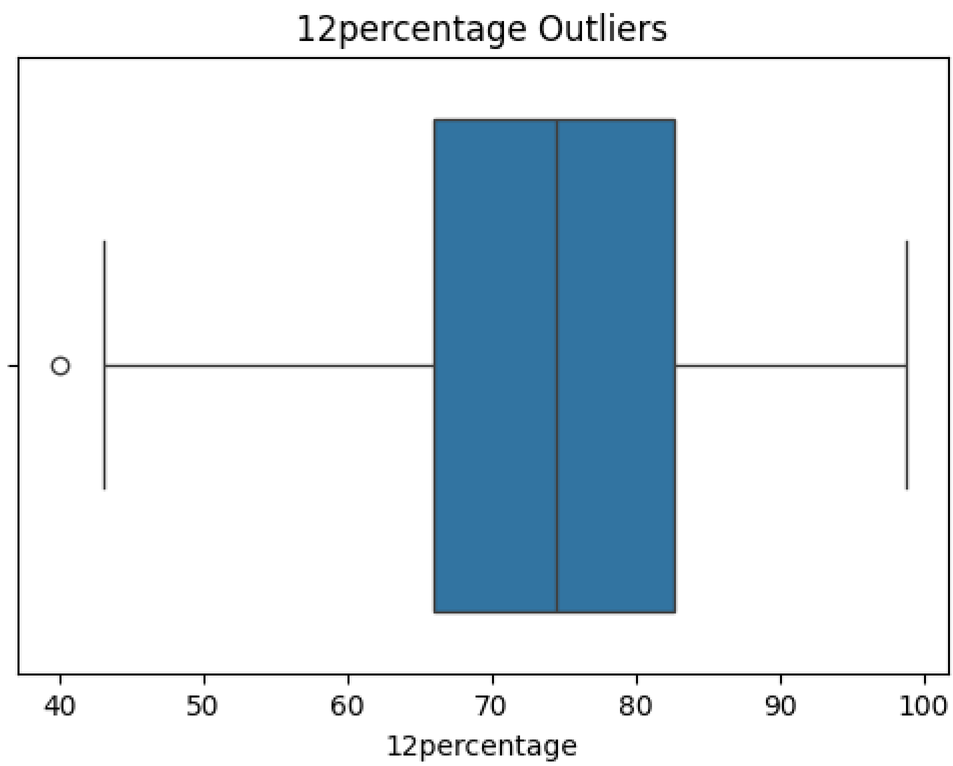
```
- The presence of outliers suggests that there are a few people
scored very poor.
```

2.3) 12percentage

In [69]:
```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['12percentage'], bins=10, kde=True)
plt.xlabel('12percentage')
plt.ylabel('Density')
plt.title('12percentage Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='12percentage', data=df1)
plt.xlabel('12percentage')
plt.title('12percentage Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 50% during their 12th class
and majority have scored in between 64%-81%.
- 75% is where the peak touched.
```

Box plot

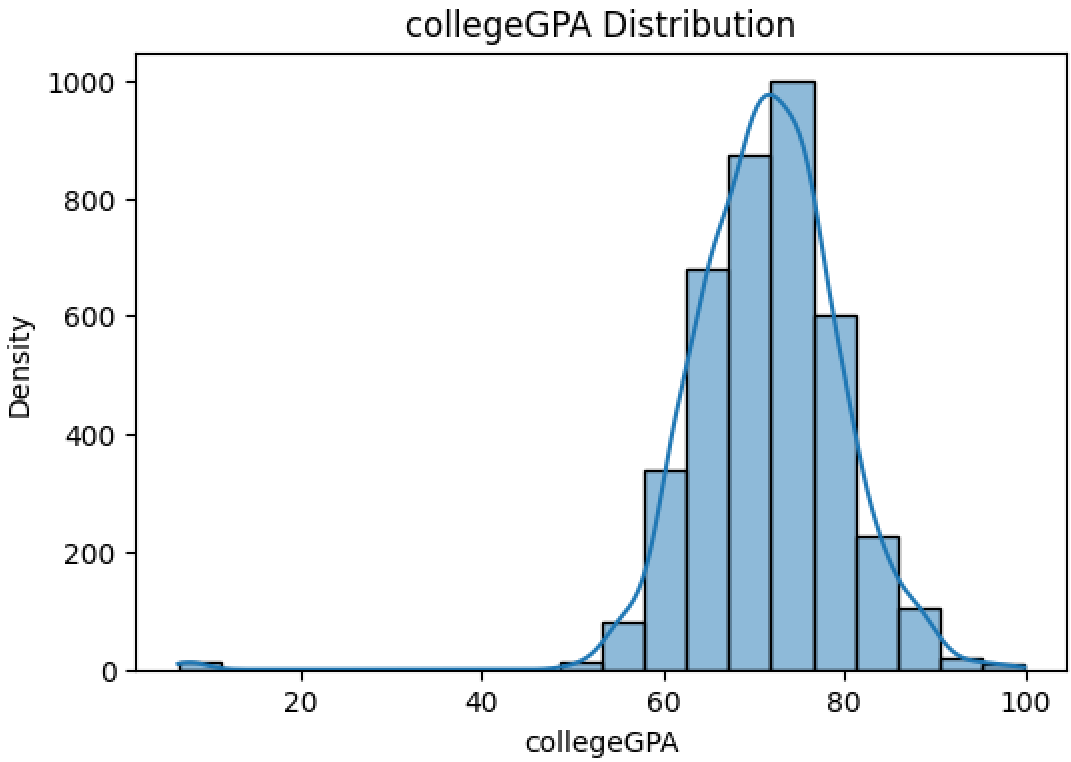
```
- There exists only one extreme score.
```

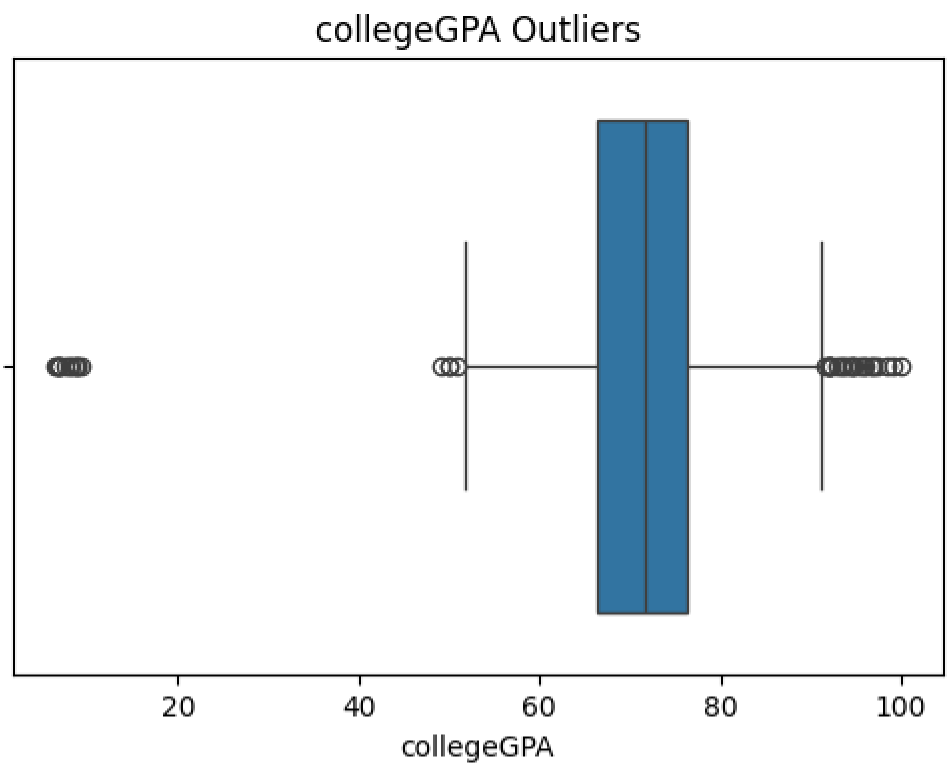
2.4) collegeGPA

In [73]:

```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['collegeGPA'], bins=20, kde=True)
plt.xlabel('collegeGPA')
plt.ylabel('Density')
plt.title('collegeGPA Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='collegeGPA', data=df1)
plt.xlabel('collegeGPA')
plt.title('collegeGPA Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 58% during their college and
majority have scored in between 62%-81%.
- 72% is where the peak touched.
```

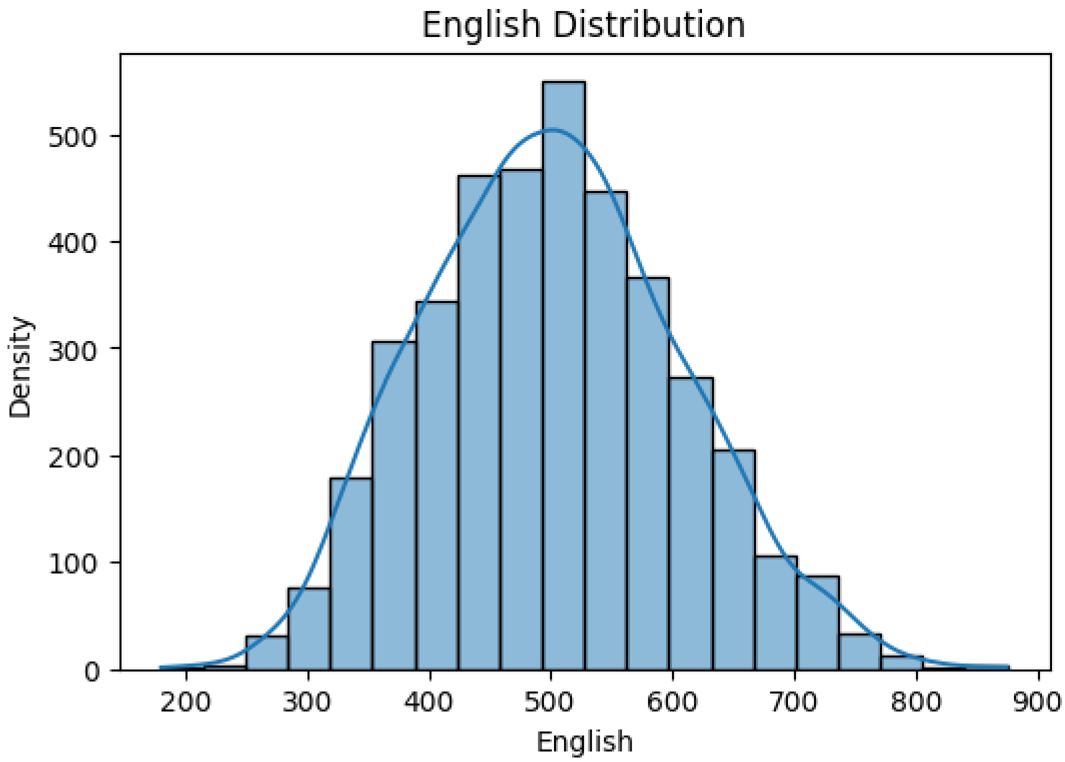
Box plot

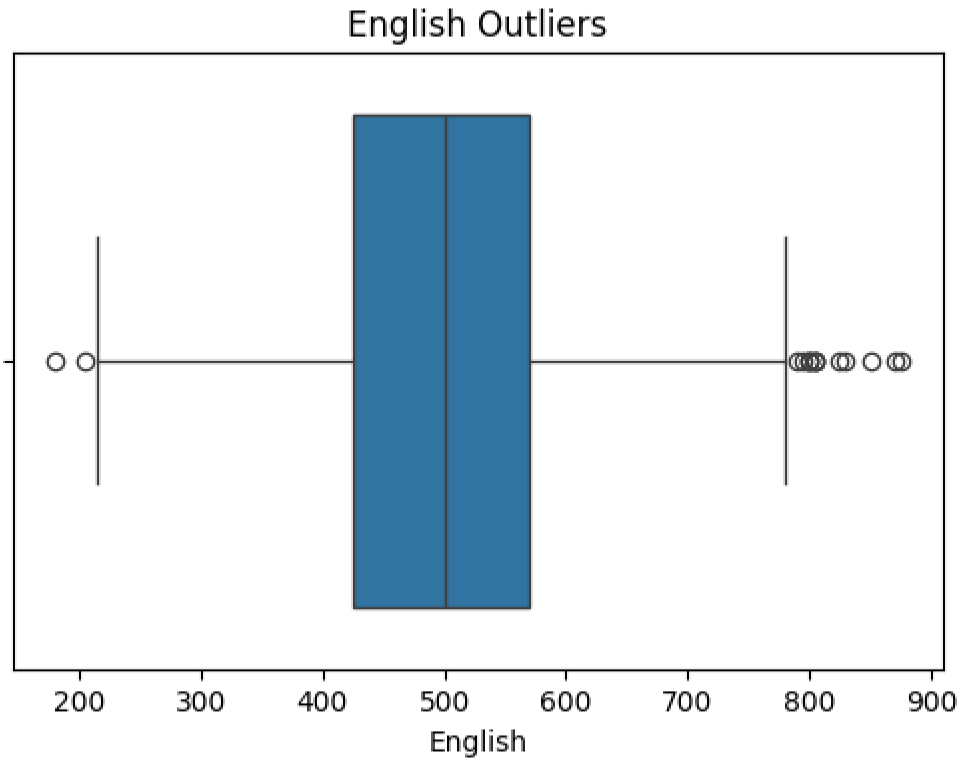
```
- There exists both low and high extreme values in the dataset.
```

2.5) English

In [74]:
```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['English'], bins=20, kde=True)
plt.xlabel('English')
plt.ylabel('Density')
plt.title('English Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='English', data=df1)
plt.xlabel('English')
plt.title('English Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 320 during their English
section and majority have scored in between 400-600.
- 500 is where the peak touched.
- Few people scored more than 730.
```

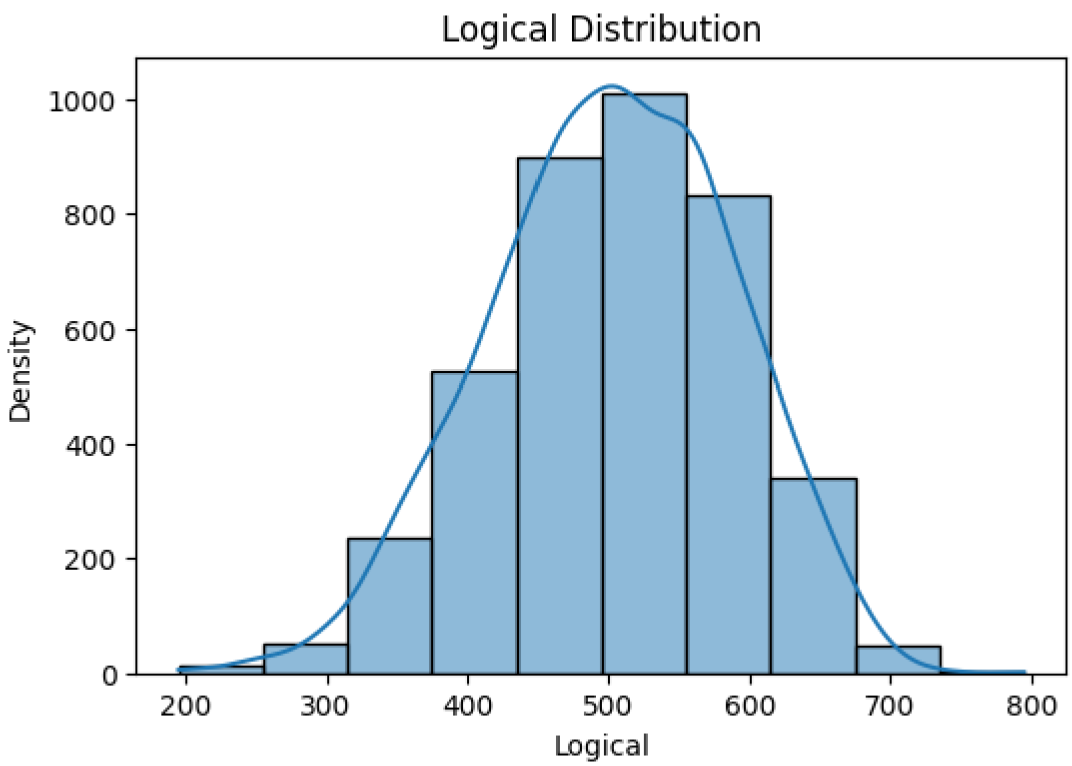
Box plot

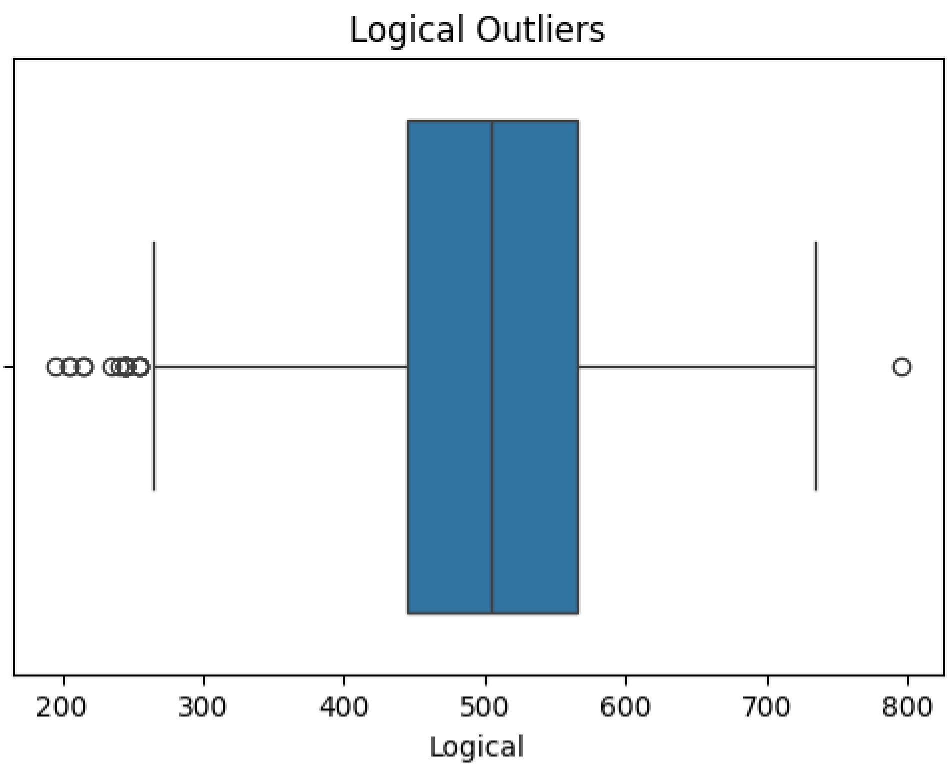
    - There exists both low and high extreme values in the dataset.

2.6) Logical

In [76]:
```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['Logical'], bins=10, kde=True)
plt.xlabel('Logical')
plt.ylabel('Density')
plt.title('Logical Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='Logical', data=df1)
plt.xlabel('Logical')
plt.title('Logical Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 320 during their Logical
section and majority have scored in between 440-620.
- 500 is where the peak touched.
- Few people scored more than 680.
```

Box plot

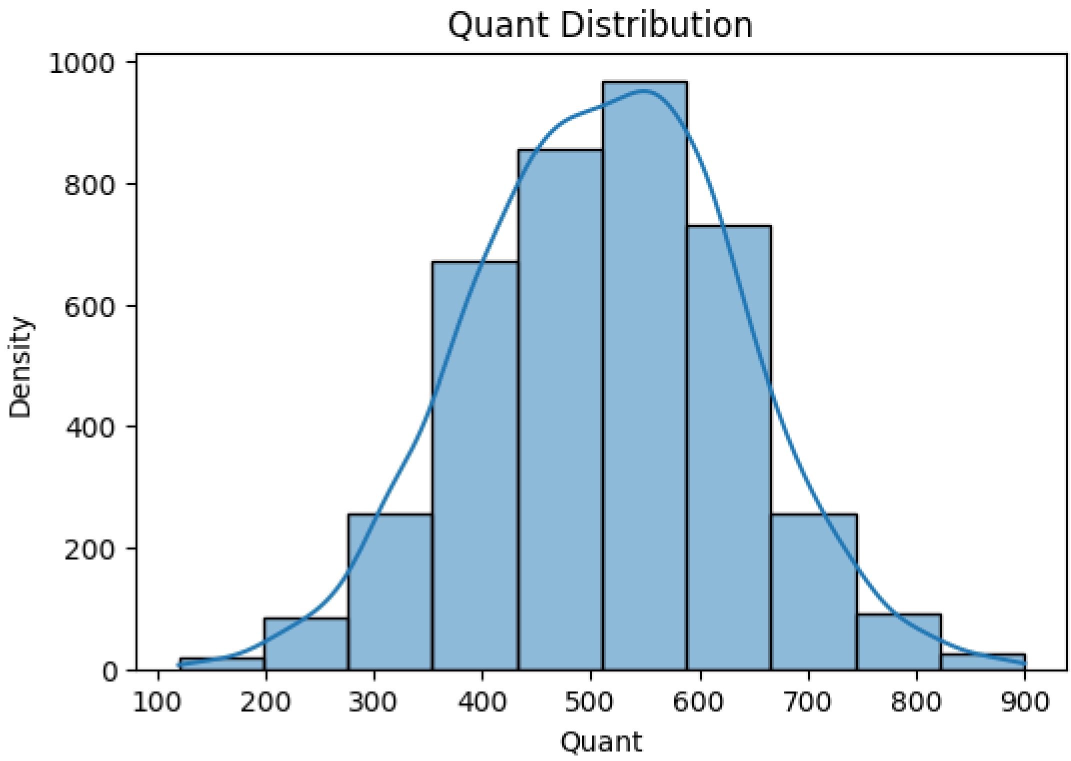
```
- There exists more low and only one high extreme values in the
dataset.
```

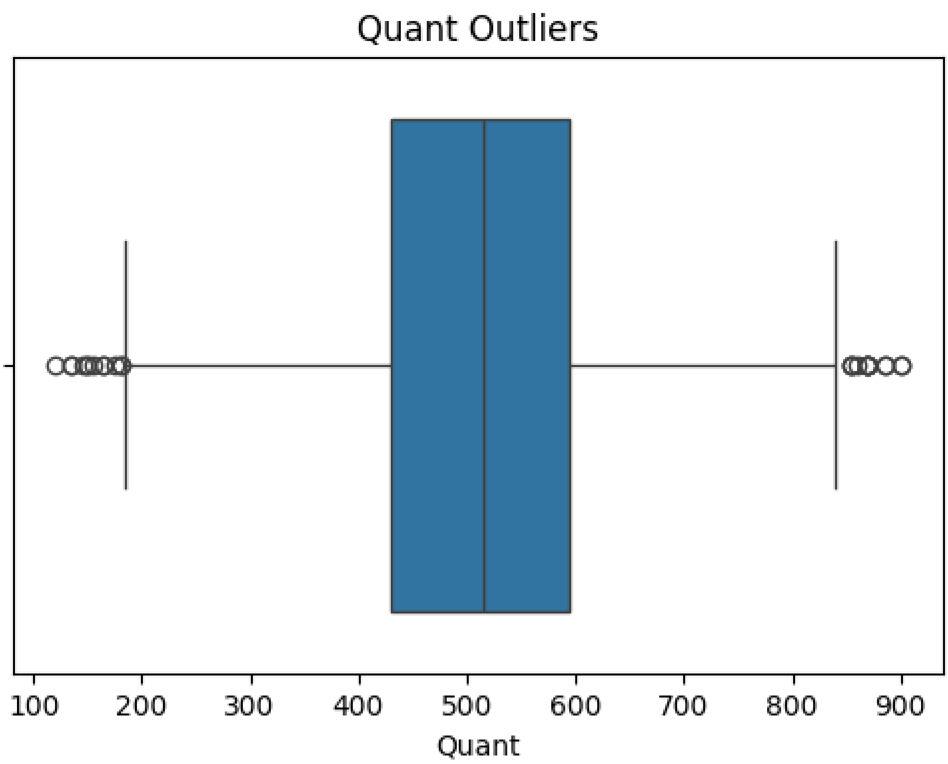
2.7) Quant

In [78]:

```
# Histogram and PDF
plt.figure(figsize=(6,4))
sns.histplot(df1['Quant'], bins=10, kde=True)
plt.xlabel('Quant')
plt.ylabel('Density')
plt.title('Quant Distribution')
plt.show()

#Boxplot
plt.figure(figsize=(6,4))
sns.boxplot(x='Quant', data=df1)
plt.xlabel('Quant')
plt.title('Quant Outliers')
plt.show()
```





Observations:

Histogram and PDF

```
- Very few people scored less than 275 during their Quantitative
section and majority have scored in between 350-750.
- 550 is where the peak touched.
- Few people scored more than 750.
```

Box plot

```
- There exists low and  high extreme values in the dataset.
```

3. Bivariate Analysis

3.1) Numerical vs Numerical Data

In [89]:
```
numerical_df.columns
```

Out[89]:
```
Index(['Salary', '10percentage', '12percentage', 'collegeGPA', 'English',
       'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg',
       'ElectricalEngg', 'TelecomEngg', 'CivilEngg', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience'],
      dtype='object')
```

In [90]:
```
numerical_df.corr()
```

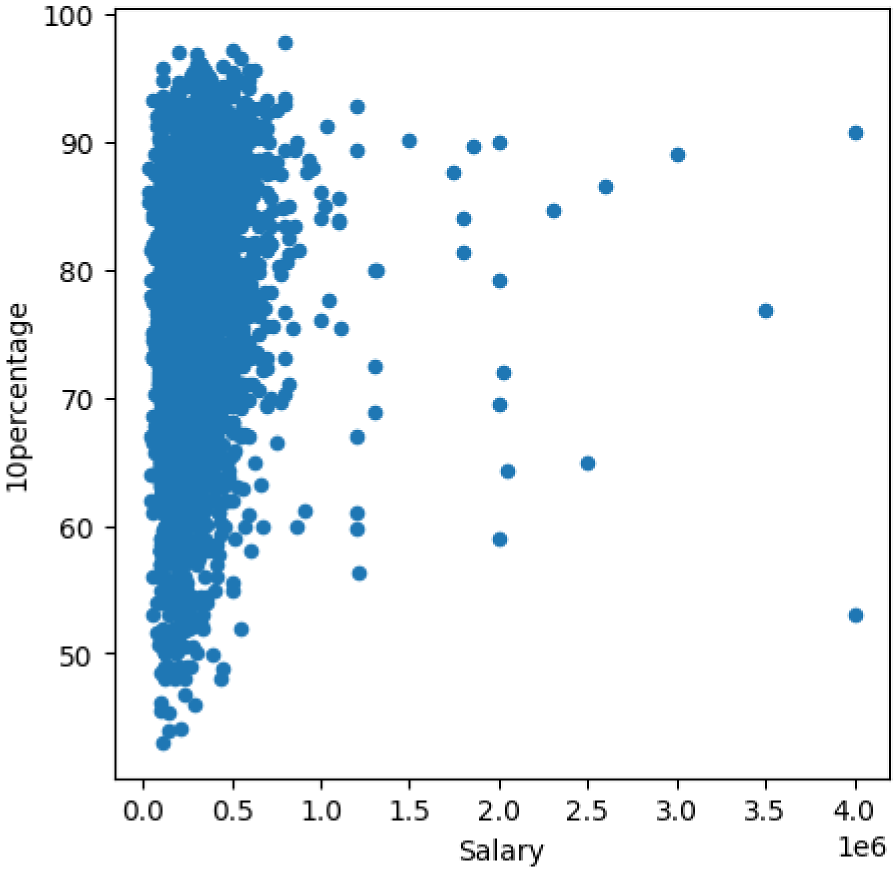
Out[90]:

	Salary	10percentage	12percentage	collegeGPA	English
Salary	1.000000	0.176111	0.168387	0.131170	0.181473
10percentage	0.176111	1.000000	0.642804	0.312301	0.351221
12percentage	0.168387	0.642804	1.000000	0.345299	0.213469
collegeGPA	0.131170	0.312301	0.345299	1.000000	0.108084
English	0.181473	0.351221	0.213469	0.108084	1.000000
Logical	0.181925	0.314110	0.243251	0.196904	0.443568
Quant	0.233882	0.315396	0.312657	0.218343	0.375012
Domain	0.104620	0.078206	0.074581	0.106584	0.092401
ComputerProgramming	0.116969	0.052754	0.079941	0.136796	0.125609
ElectronicsAndSemicon	0.000542	0.084924	0.118661	0.030494	0.016030
ComputerScience	-0.102292	-0.014481	-0.041577	0.008572	0.062696
MechanicalEngg	0.017700	0.050078	0.037606	-0.032017	-0.002358
ElectricalEngg	-0.046517	0.072771	0.063380	0.050072	0.033735
TelecomEngg	-0.023638	0.048320	0.042561	-0.005114	-0.007648
CivilEngg	0.037538	0.029931	0.005850	-0.019068	-0.007709
conscientiousness	-0.064481	0.067123	0.057569	0.069386	0.033413
agreeableness	0.057049	0.136331	0.103919	0.068916	0.193818
extraversion	-0.010532	-0.004990	-0.008968	-0.032335	0.016578
nueroticism	-0.054375	-0.131989	-0.094070	-0.074362	-0.157609
openess_to_experience	-0.012081	0.037397	0.004325	0.028021	0.068409

In [91]:

```
numerical_df.plot(kind='scatter', x='Salary', y='10percentage', figsize=(5, 5))
```

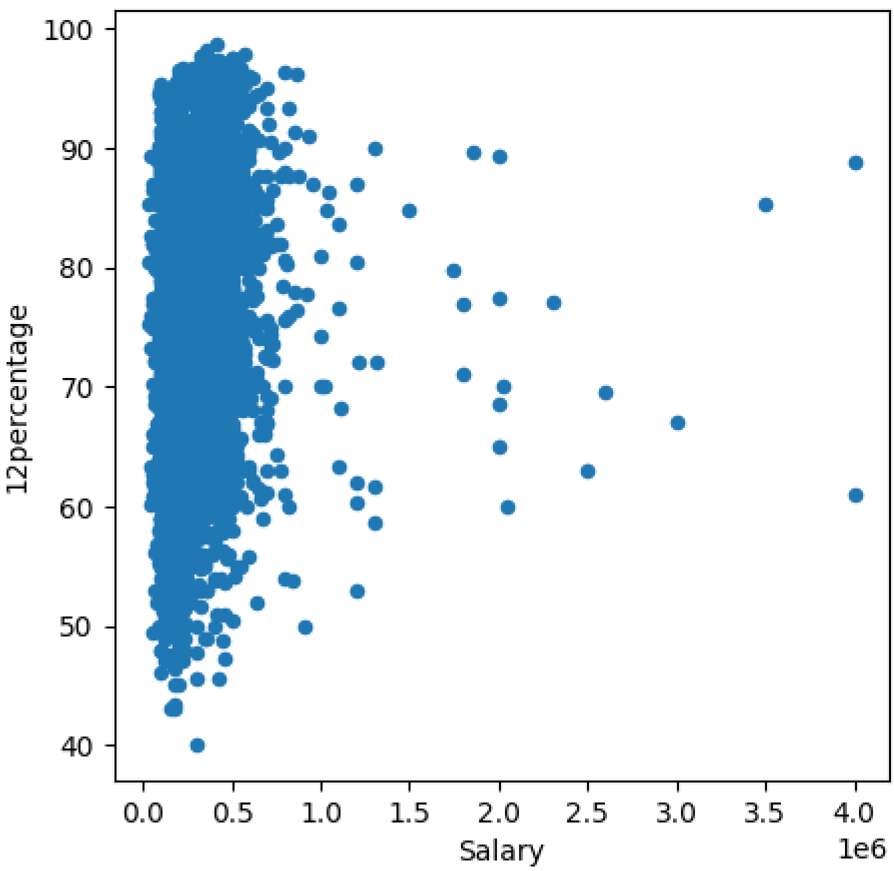
Out[91]: <Axes: xlabel='Salary', ylabel='10percentage'>



No correlation exists.

In [92]:
```
numerical_df.plot(kind='scatter', x='Salary', y='12percentage', figsize=(5, 5))
```

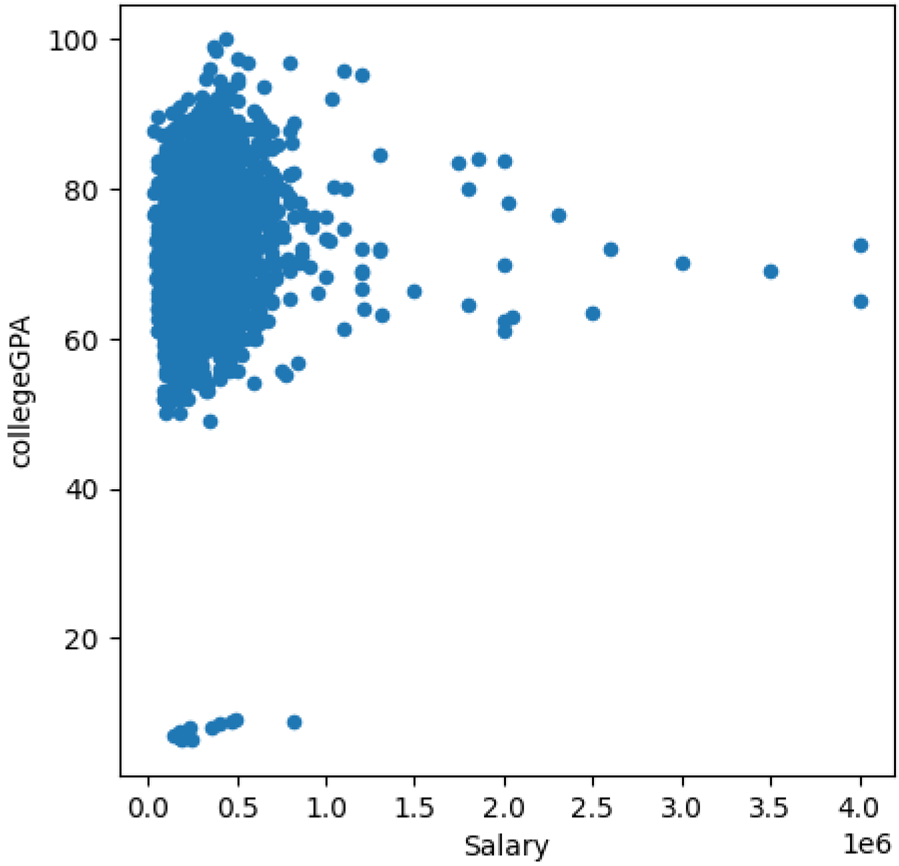
Out[92]: <Axes: xlabel='Salary', ylabel='12percentage'>



No correlation exists.

In [93]:
```
numerical_df.plot(kind='scatter', x='Salary', y='collegeGPA', figsize=(5, 5))
```

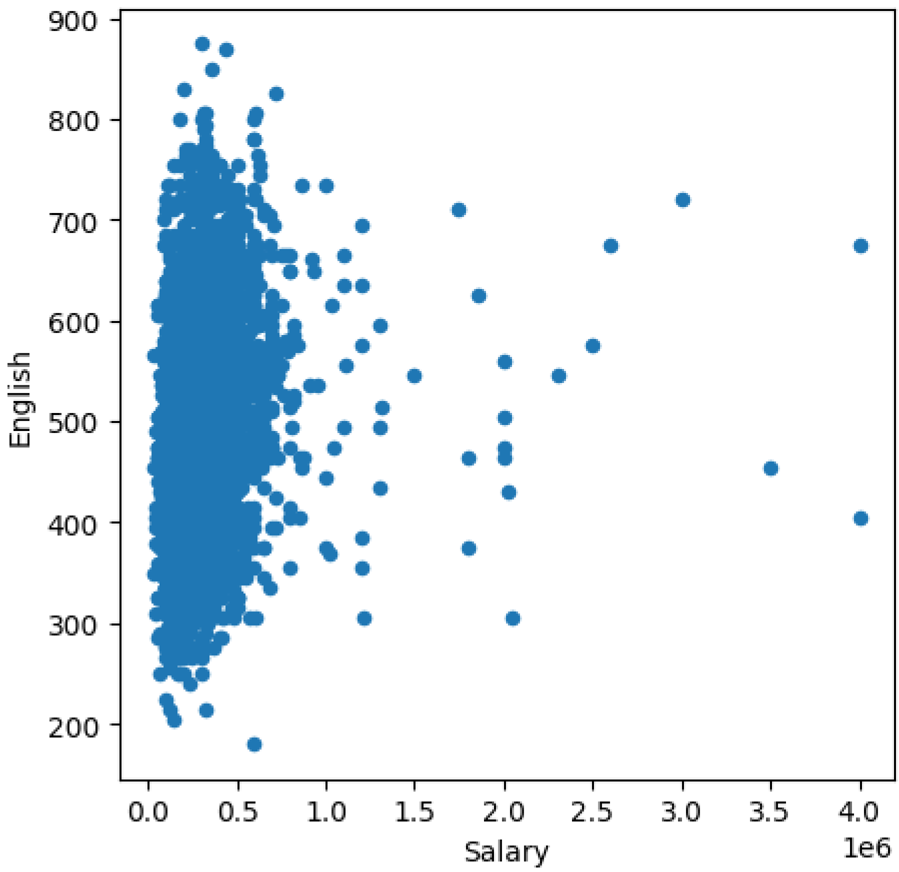
Out[93]: <Axes: xlabel='Salary', ylabel='collegeGPA'>



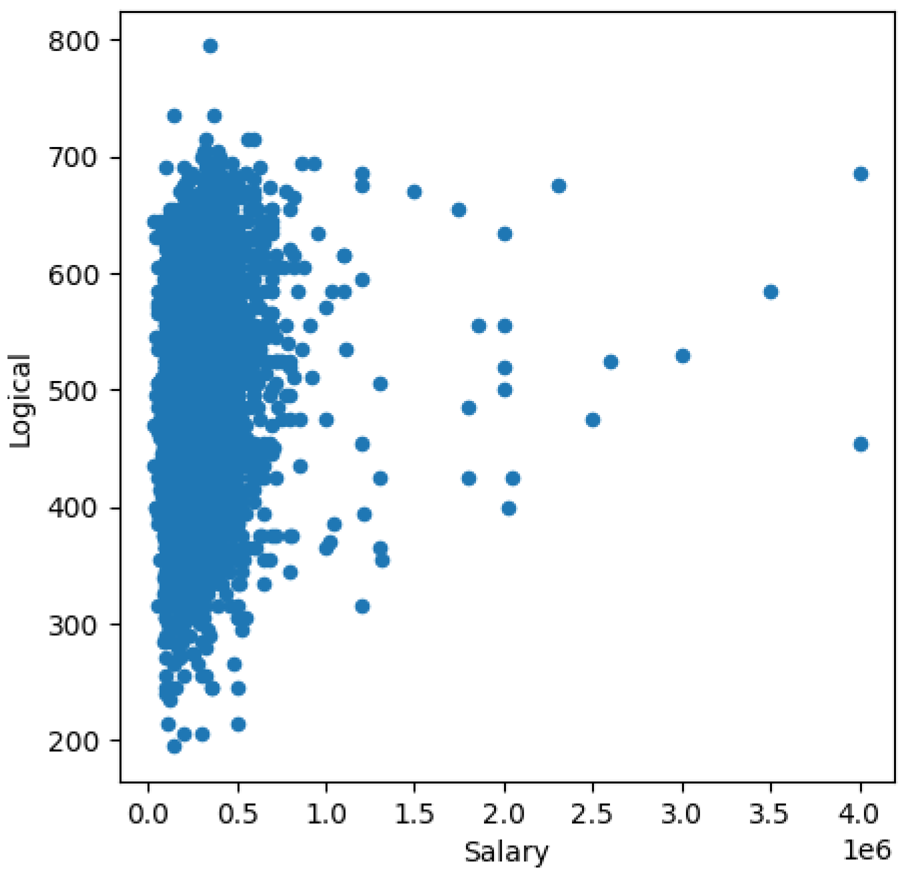
No correlation exists.

In [94]:
```
numerical_df.plot(kind='scatter', x='Salary', y='English', figsize=(5, 5))
numerical_df.plot(kind='scatter', x='Salary', y='Logical', figsize=(5, 5))
numerical_df.plot(kind='scatter', x='Salary', y='Quant', figsize=(5, 5))
```

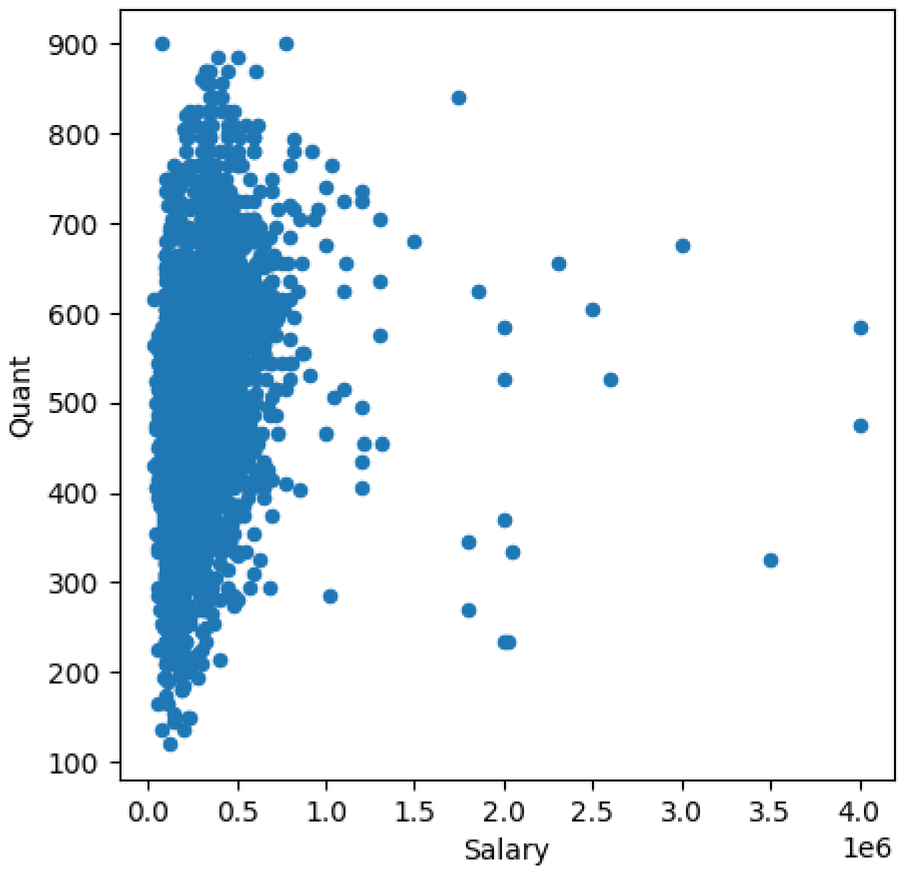
Out[94]: <Axes: xlabel='Salary', ylabel='Quant'>



English
Salary
1e6



Logical
Salary
1e6



No correlation exists.

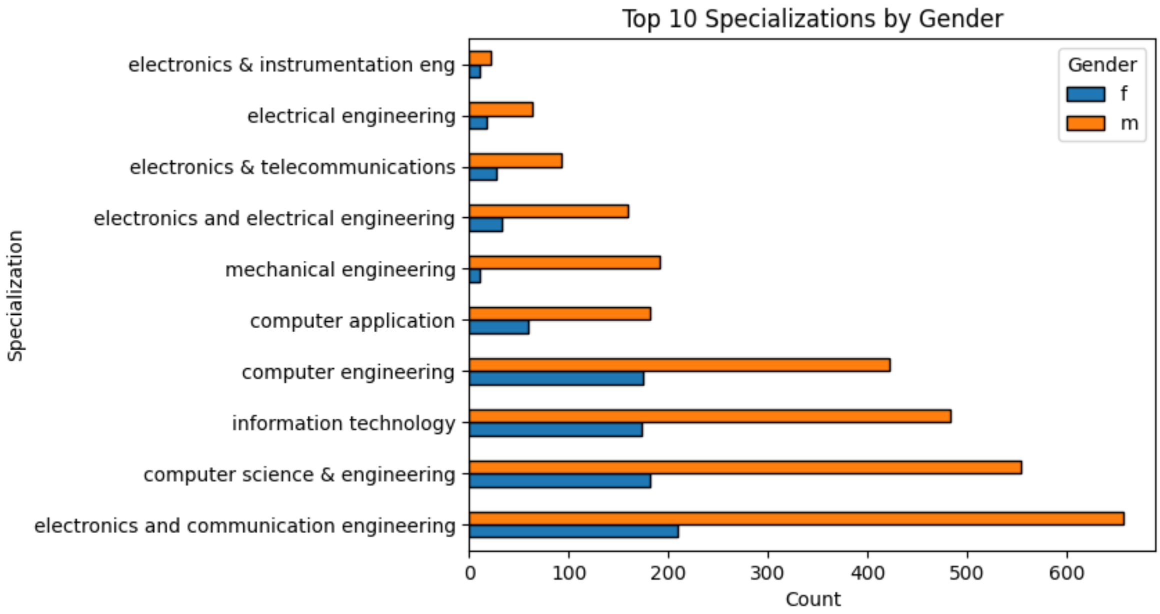
3.2) categorical vs categorical data

In [95]:
```
categorical_df.columns
```

Out[95]:
```
Index(['Designation', 'JobCity', 'Gender', '10board', '12board', 'Degree',
       'Specialization', 'CollegeState'],
      dtype='object')
```

In [101…
```
plt.figure(figsize=(6, 4))
top_10_specializations = categorical_df['Specialization'].value_counts().nlarges
crosstab_df = pd.crosstab(categorical_df['Gender'], categorical_df['Specializati
crosstab_df.loc[:, top_10_specializations].T.plot(kind='barh', ec='k')
plt.xlabel('Count')
plt.ylabel('Specialization')
plt.title('Top 10 Specializations by Gender')
plt.legend(title='Gender')
plt.show()
```

```
<Figure size 600x400 with 0 Axes>
```

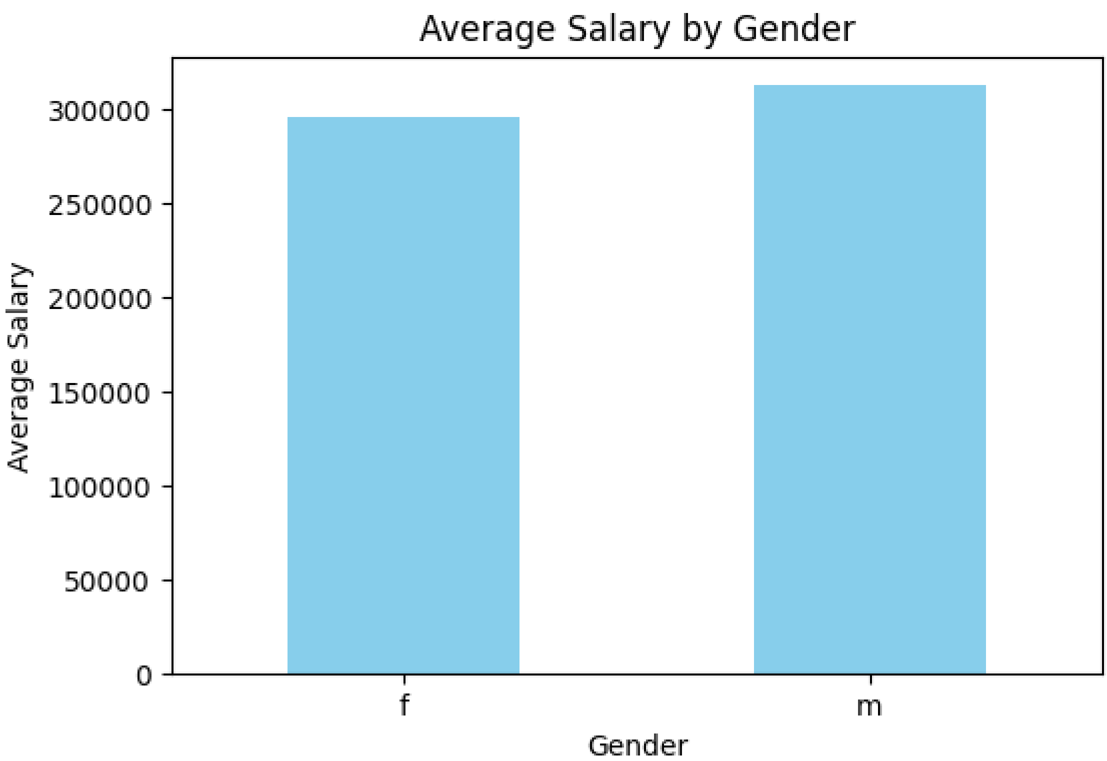


Males occupied 3 times more than females in every specialization.

3.3) Categorical vs Numerical data.

In [103…
```
group = df1.groupby('Gender')
avg_salary_by_gender = group['Salary'].mean()
```

In [104…
```
plt.figure(figsize=(6, 4))
avg_salary_by_gender.plot(kind='bar', color='skyblue')
plt.xlabel('Gender')
plt.ylabel('Average Salary')
plt.title('Average Salary by Gender')
plt.xticks(rotation=0)
plt.show()
```



The avg salary for both male and female are approximately equal.

In [ ]:
```

```