AMCAT Data Analysis

In [79]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [80]:
```
data=pd.read_csv(r"C:\Users\ADMIN\Downloads\data.csv")
```

1.Display Top 10 Rows Of The Dataset

In [81]:
```
data.head(10)
```

Out[81]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10p
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	
5	train	1027655	300000.0	6/1/14 0:00	present	system engineer	Hyderabad	m	7/2/92 0:00	
6	train	947847	300000.0	8/1/14 0:00	5/1/15 0:00	java software engineer	Banglore	m	2/1/93 0:00	
7	train	912934	400000.0	7/1/14 0:00	7/1/15 0:00	mechanical engineer	Bangalore	m	5/27/92 0:00	
8	train	552574	600000.0	7/1/13 0:00	present	electrical engineer	Noida	m	9/17/91 0:00	
9	train	1203363	230000.0	7/1/14 0:00	present	project engineer	Kolkata	m	6/13/93 0:00	

10 rows × 39 columns

2.Displaying the Last 10 Rows of The Dataset

In [82]:
```
data.tail(10)
```

Out[82]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	
3988	train	466661	200000.0	2/1/13 0:00	7/1/13 0:00	data analyst	Bangalore	f	5/
3989	train	1204604	300000.0	9/1/14 0:00	present	software engineer	Bangalore	m	11/
3990	train	204287	480000.0	2/1/12 0:00	present	senior systems engineer	Hyderabad	f	9
3991	train	230873	630000.0	7/1/11 0:00	10/1/14 0:00	systems analyst	Bangalore	m	5/
3992	train	344407	800000.0	4/1/14 0:00	4/1/15 0:00	manager	Rajkot	m	6/
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/

10 rows × 39 columns

3.Find Shape of Our Dataset(Number of Rows and Number of Columns)

In [83]:
```
data.shape
```

Out[83]: (3998, 39)

4.Information about Our Dataset like Number of Rows,Number of Columns,Datatype and

Memory Usage

In [84]:
```
data.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 [85]:
```
data.describe()
```

Out[85]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	C
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	39
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	

8 rows × 27 columns

In [86]:
```
data.columns
```

Out[86]:
```
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 [87]:
```
data.isnull().sum()
```

Out[87]:
```
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 [88]:
```
data.duplicated().sum()
```

Out[88]:
```
0
```

In [89]:
```
data.nunique()
```

Out[89]:
```
Unnamed: 0               1
ID                    3998
Salary                 177
DOJ                     81
DOL                     67
Designation            419
JobCity                339
Gender                   2
DOB                   1872
10percentage           851
10board                275
12graduation            16
12percentage           801
12board                340
CollegeID             1350
CollegeTier              2
Degree                   4
Specialization          46
collegeGPA            1282
CollegeCityID         1350
CollegeCityTier          2
CollegeState            26
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          149
extraversion           154
nueroticism            217
openess_to_experience  142
dtype: int64
```

In [90]:
```
(data.isnull().sum()/(len(data)))*100
```

Out[90]:
```
Unnamed: 0              0.0
ID                      0.0
Salary                  0.0
DOJ                     0.0
DOL                     0.0
Designation             0.0
JobCity                 0.0
Gender                  0.0
DOB                     0.0
10percentage            0.0
10board                 0.0
12graduation            0.0
12percentage            0.0
12board                 0.0
CollegeID               0.0
CollegeTier             0.0
Degree                  0.0
Specialization          0.0
collegeGPA              0.0
CollegeCityID           0.0
CollegeCityTier         0.0
CollegeState            0.0
GraduationYear          0.0
English                 0.0
Logical                 0.0
Quant                   0.0
Domain                  0.0
ComputerProgramming     0.0
ElectronicsAndSemicon   0.0
ComputerScience         0.0
MechanicalEngg          0.0
ElectricalEngg          0.0
TelecomEngg             0.0
CivilEngg               0.0
conscientiousness       0.0
agreeableness           0.0
extraversion            0.0
nueroticism             0.0
openess_to_experience   0.0
dtype: float64
```

In [91]:
```
list(data.Gender.unique())
```

Out[91]:
```
['f', 'm']
```

In [92]:
```
list(data.JobCity.unique())
```

Out[92]:
```
['Bangalore',
 'Indore',
 'Chennai',
 'Gurgaon',
 'Manesar',
 'Hyderabad',
 'Banglore',
 'Noida',
 'Kolkata',
 'Pune',
 '-1',
 'mohali',
 'Jhansi',
 'Delhi',
 'Hyderabad ',
 'Bangalore ',
 'noida',
 'delhi',
 'Bhubaneswar',
```

In [93]:
```
data.describe(include='O')
```

Out[93]:

	Unnamed: 0	DOJ	DOL	Designation	JobCity	Gender	DOB	10board	12board	
count	3998	3998	3998	3998	3998	3998	3998	3998	3998	
unique	1	81	67	419	339	2	1872	275	340	
top	train	7/1/14 0:00	present	software engineer	Bangalore	m	1/1/91 0:00	cbse	cbse	B
freq	3998	199	1875	539	627	3041	11	1395	1400	

In [94]:
```
data[data.duplicated(keep='first')]
```

Out[94]:

Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	Cor

0 rows × 39 columns

In [95]:
```
data.drop_duplicates(keep='first',inplace=True)
```

In [96]:
```
categorical_cols=data.select_dtypes(include=['object']).columns
numerical_cols = data.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:")
print(categorical_cols)
print("Numerical Variables:")
print(numerical_cols)
```

```
Categorical Variables:
Index(['Unnamed: 0', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DO
B',
       '10board', '12board', 'Degree', 'Specialization', 'CollegeState'],
      dtype='object')
Numerical Variables:
['ID', 'Salary', '10percentage', '12graduation', '12percentage', 'CollegeI
D', 'CollegeTier', 'collegeGPA', 'CollegeCityID', 'CollegeCityTier', 'Gradua
tionYear', 'English', 'Logical', 'Quant', 'Domain', 'ComputerProgramming',
'ElectronicsAndSemicon', 'ComputerScience', 'MechanicalEngg', 'ElectricalEng
g', 'TelecomEngg', 'CivilEngg', 'conscientiousness', 'agreeableness', 'extra
version', 'nueroticism', 'openess_to_experience']
```

In [97]:
```
len(data[data['collegeGPA']<=10])
```

Out[97]:
```
12
```

In [98]:
```
data=data[data['GraduationYear']>0]
```

In [99]:
```
len(data[data['collegeGPA']>10])
```

Out[99]:
```
3985
```

In [100]:
```
column=list(data.drop(columns=['ID','CollegeID','CollegeCityID'],axis=1).sele
column
```

Out[100]:
```
['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']
```

In [101]:
```
out_dict={}
for i in column:
    Q1 = data[i].quantile(0.05)
    Q3 = data[i].quantile(0.95)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[i] < lower_bound) | (data[i] > upper_bound)]
    out_dict[i]=outliers
```

In [102]:
```
out_dict['12graduation']
```

Out[102]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10per
59	train	536053	120000.0	9/1/09 0:00	4/1/13 0:00	software engineer	Bangalore	m	10/30/77 0:00	

1 rows × 39 columns

In [103]:
```
out_dict['collegeGPA']
```

Out[103]:

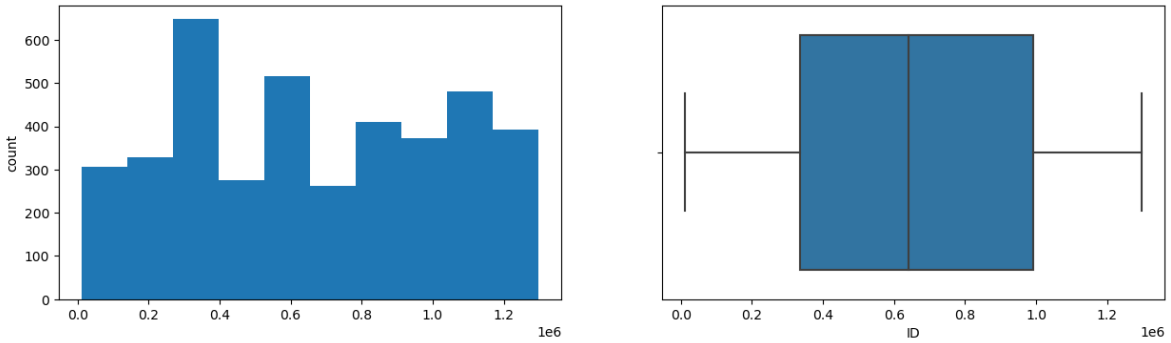
	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOE
7	train	912934	400000.0	7/1/14 0:00	7/1/15 0:00	mechanical engineer	Bangalore	m	5/27/92 0:0(
138	train	964319	195000.0	10/1/14 0:00	1/1/15 0:00	business development managerde	coimbatore	m	5/4/9' 0:0(
788	train	249853	180000.0	5/1/12 0:00	6/1/13 0:00	electrical project engineer	Jowai	m	1/12/89 0:0(
1419	train	1262900	180000.0	10/1/14 0:00	4/1/15 0:00	java software engineer	Chennai	m	6/14/93 0:0(
1439	train	299447	360000.0	8/1/11 0:00	present	assistant professor	AM	m	12/11/88 0:0(
1767	train	813008	180000.0	6/1/14 0:00	8/1/14 0:00	it technician	Bhopal	m	9/21/92 0:0(
2151	train	262814	145000.0	2/1/12 0:00	4/1/13 0:00	web developer	New Delhi	m	6/18/88 0:0(
2229	train	868740	240000.0	1/1/15 0:00	4/1/15 0:00	product development engineer	Chennai	m	5/1/92 0:0(
2293	train	407736	490000.0	10/1/12 0:00	12/1/14 0:00	software engineer	-1	f	3/18/90 0:0(
2662	train	240465	470000.0	7/1/11 0:00	3/1/15 0:00	systems engineer	Kolkata	m	2/15/90 0:0(
2691	train	385442	820000.0	7/1/14 0:00	3/1/15 0:00	software engineer	New Delhi	m	10/28/90 0:0(
3308	train	287976	250000.0	8/1/11 0:00	11/1/12 0:00	engineer	Aurangabad	m	6/7/85 0:0(

12 rows × 39 columns

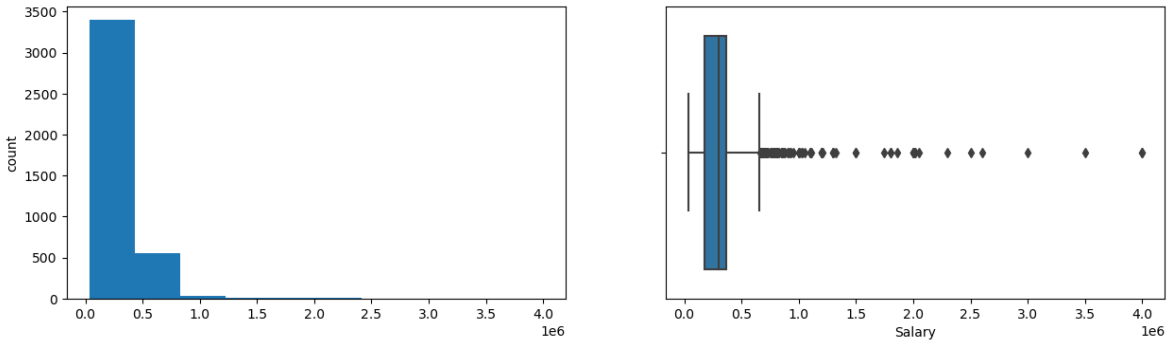
Univariate Analysis

In [104]:
```
for col in numerical_cols:
    print(col)
    print('Skew :', round(data[col].skew(), 2))
    plt.figure(figsize = (15, 4))
    plt.subplot(1, 2, 1)
    data[col].hist(grid=False)
    plt.ylabel('count')
    plt.subplot(1, 2, 2)
    sns.boxplot(x=data[col])
    plt.show()
```

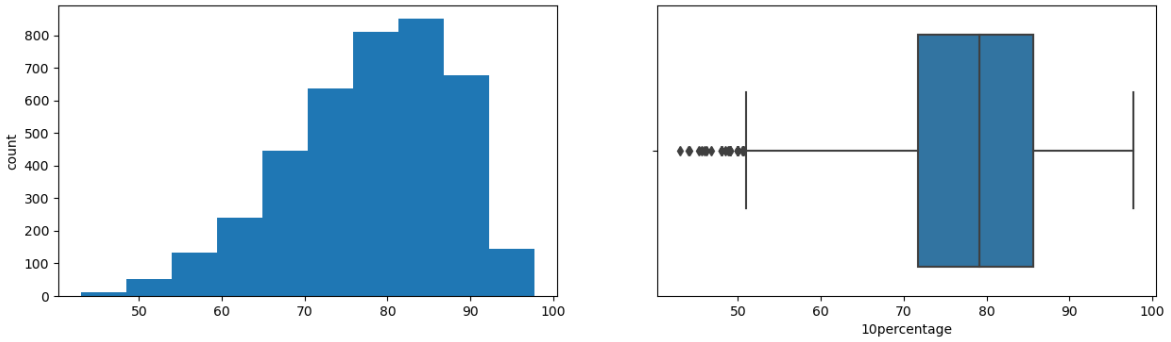
```
ID
Skew : 0.06
```



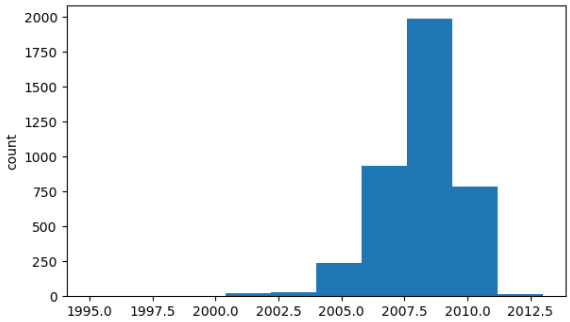

```
Salary
Skew : 6.45
```

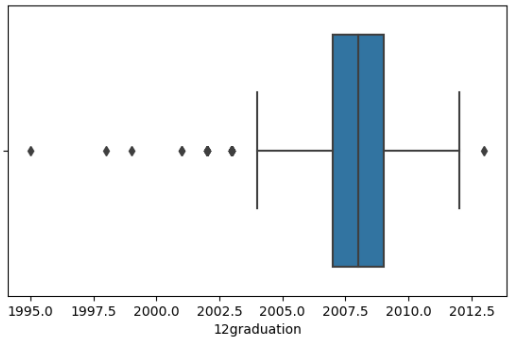



```
10percentage
Skew : -0.59
```

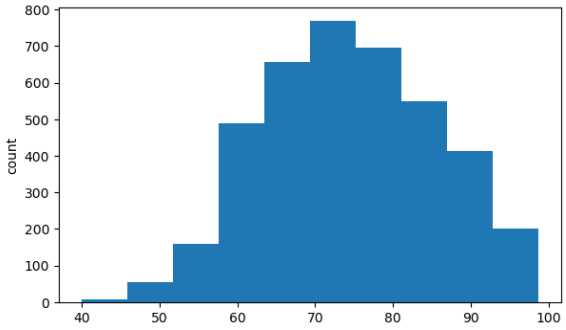


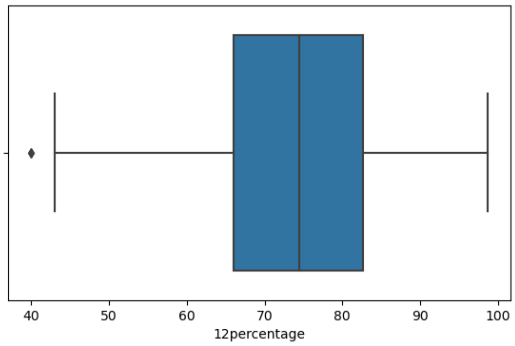

```
12graduation
Skew : -0.96
```



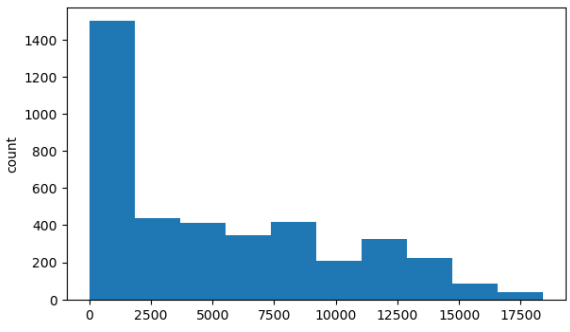


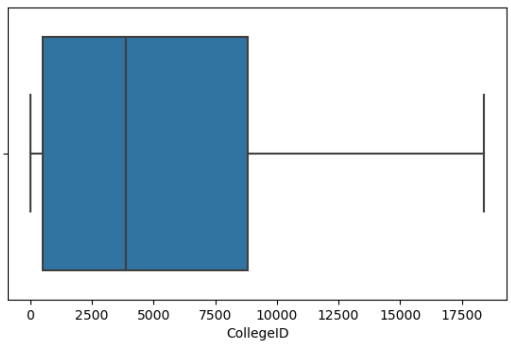
12percentage
Skew : -0.03



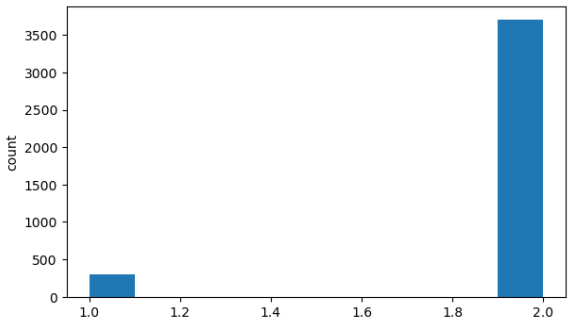


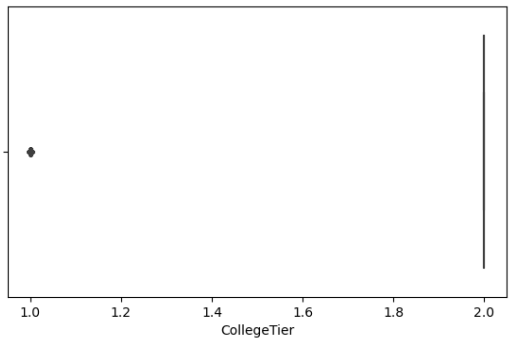
CollegeID
Skew : 0.65



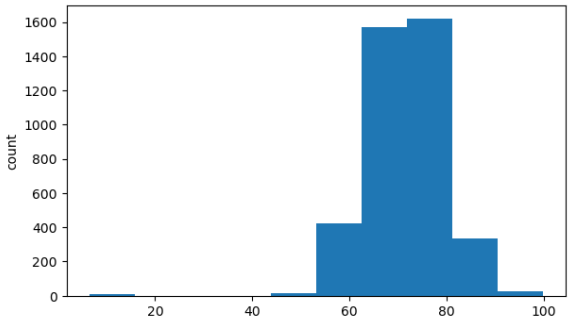


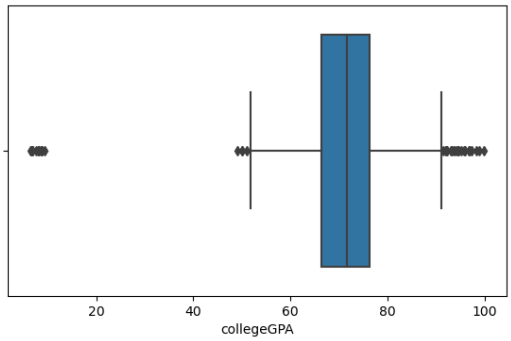
CollegeTier
Skew : -3.25



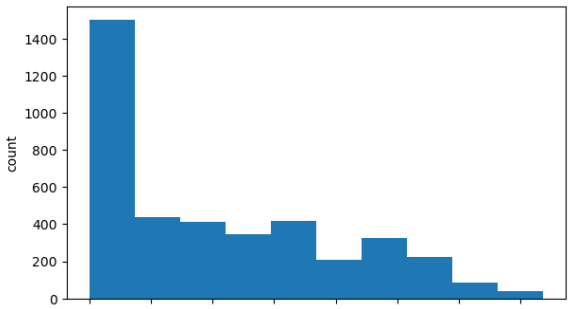


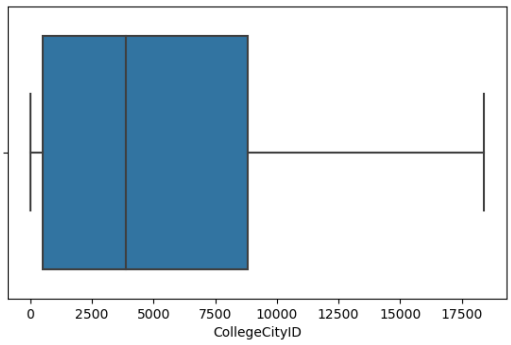
collegeGPA
Skew : -1.25



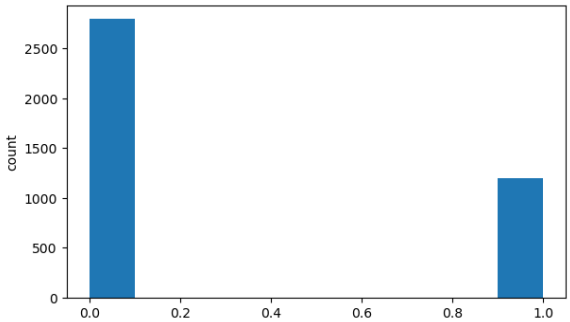


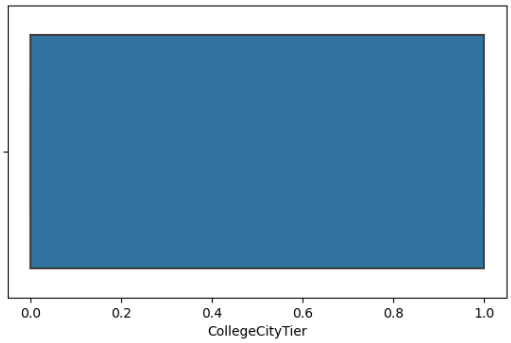
CollegeCityID
Skew : 0.65



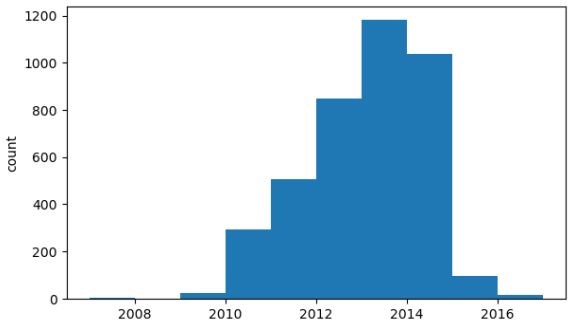


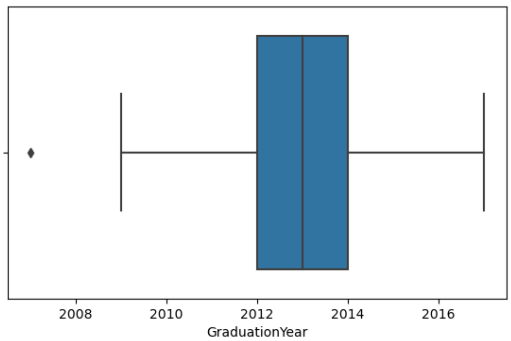
CollegeCityTier
Skew : 0.87



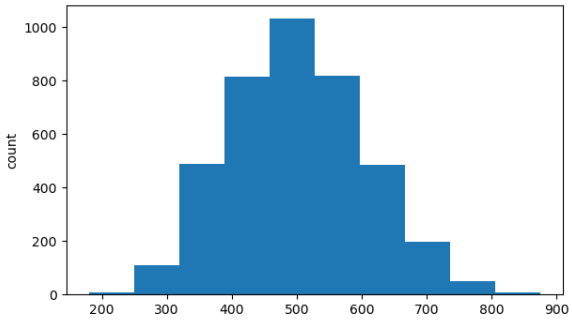


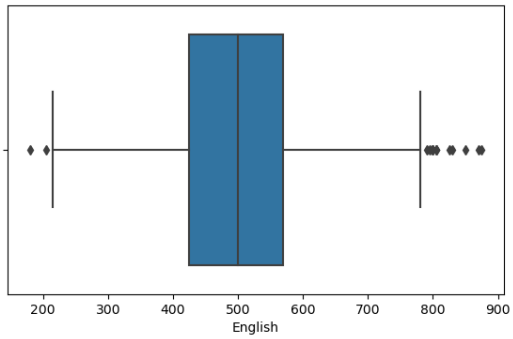
GraduationYear
Skew : -0.41



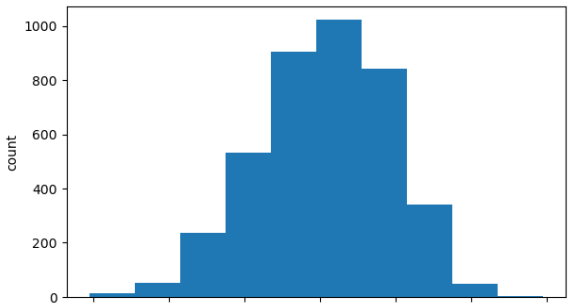


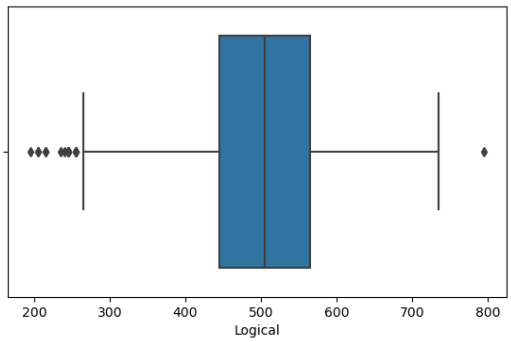
English
Skew : 0.19



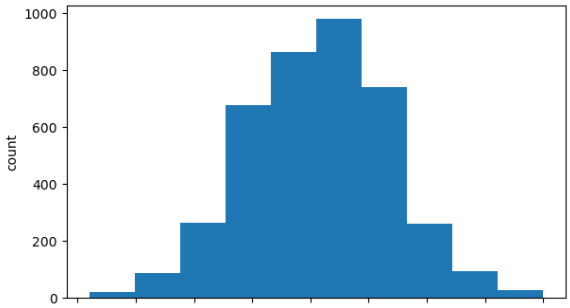


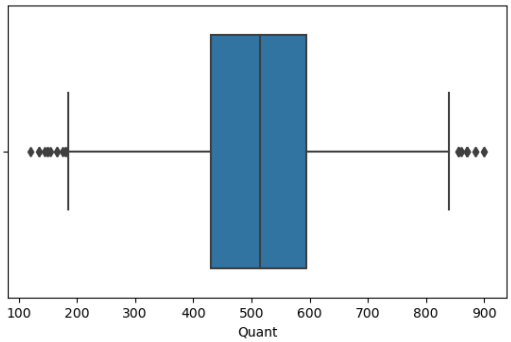
Logical
Skew : -0.22



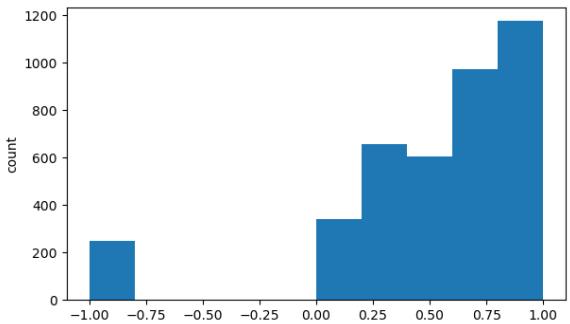


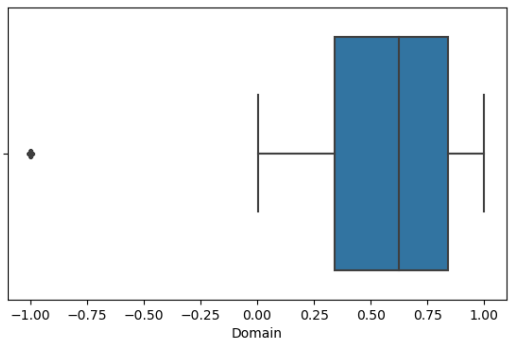
Quant
Skew : -0.02



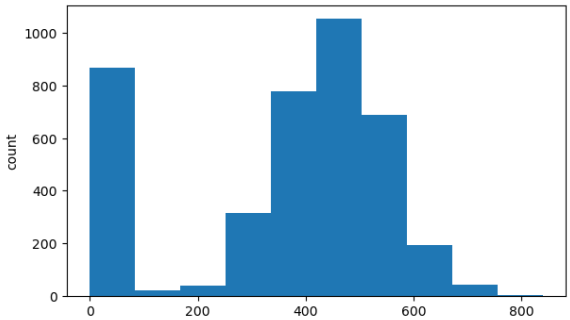


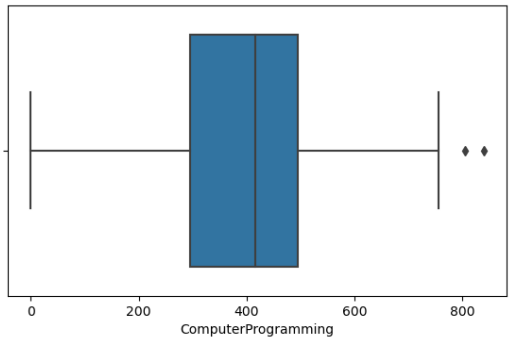
Domain
Skew : -1.92



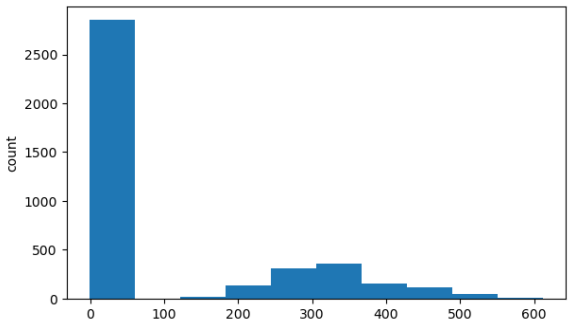


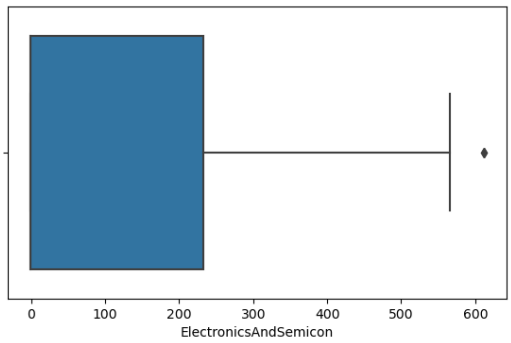
ComputerProgramming
Skew : -0.78



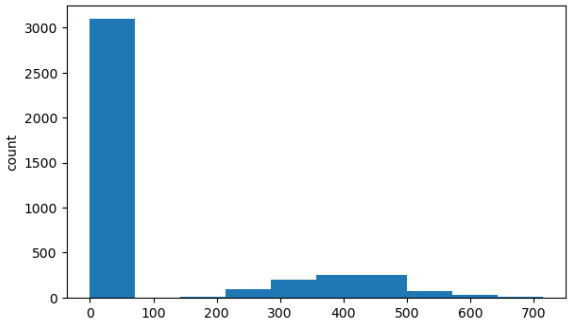


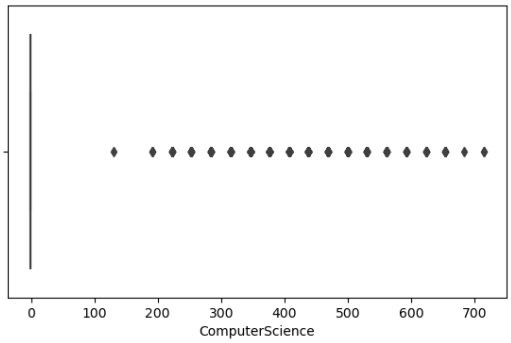
ElectronicsAndSemicon
Skew : 1.2



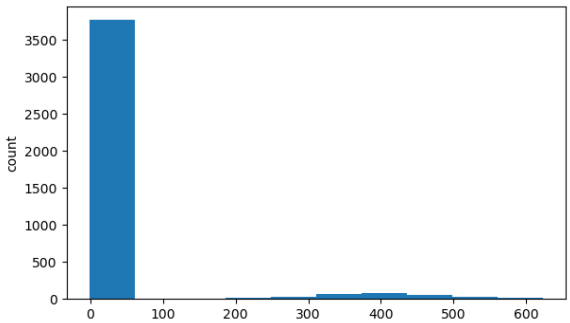


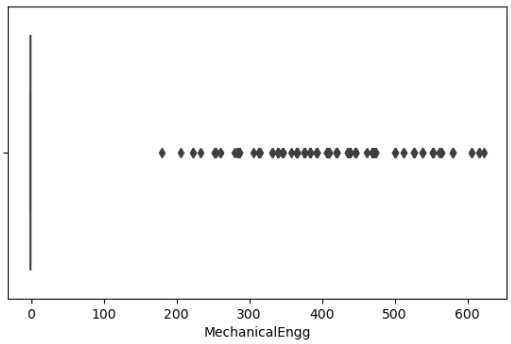
ComputerScience
Skew : 1.53



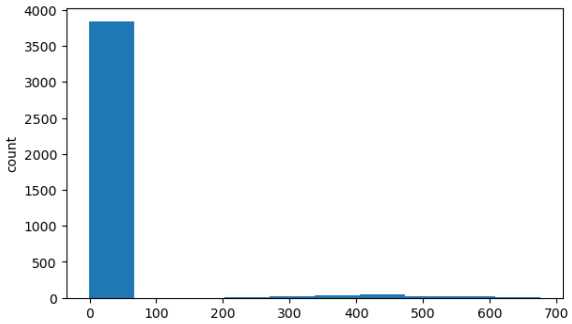


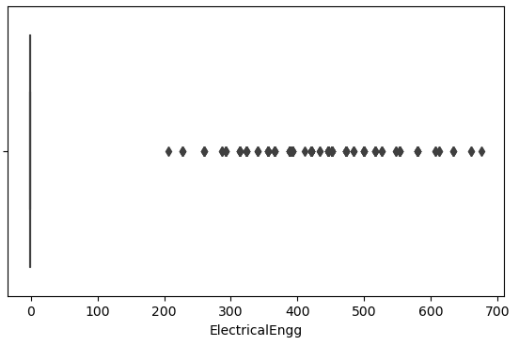
MechanicalEngg
Skew : 4.04



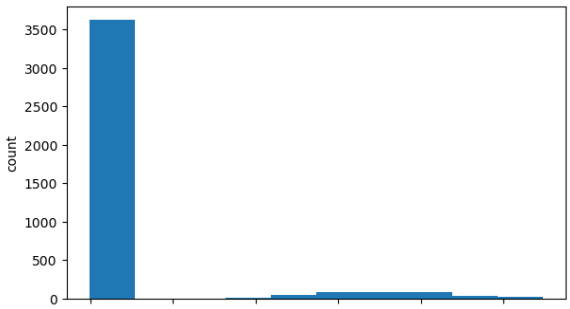


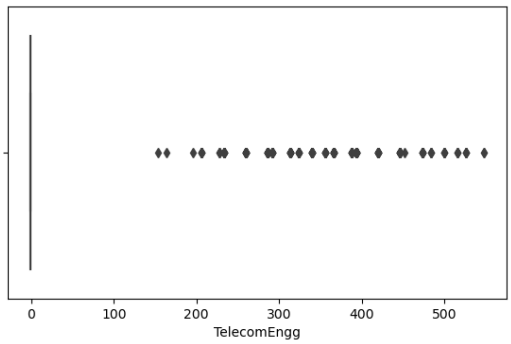
ElectricalEngg
Skew : 5.06



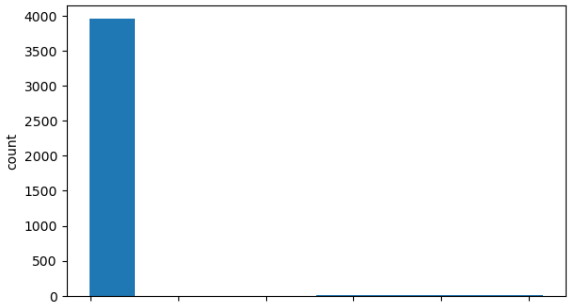


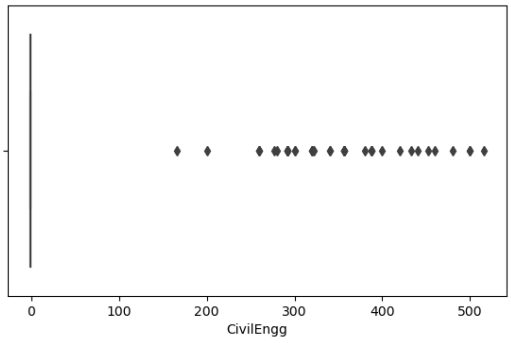
TelecomEngg
Skew : 3.04



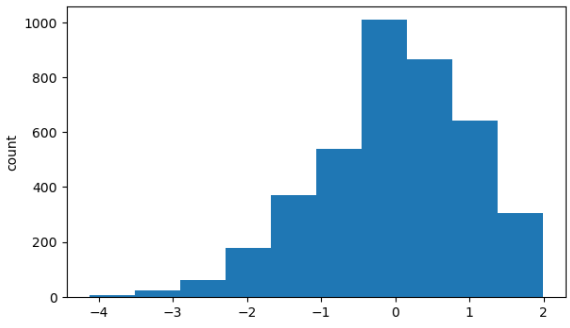


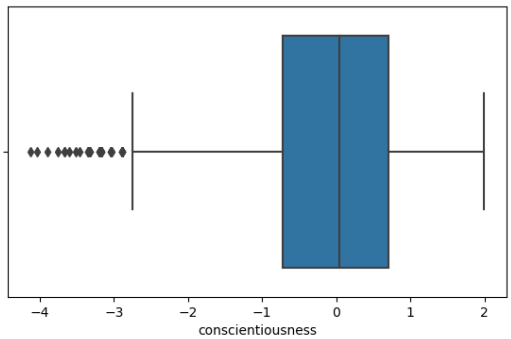
CivilEngg
Skew : 10.31



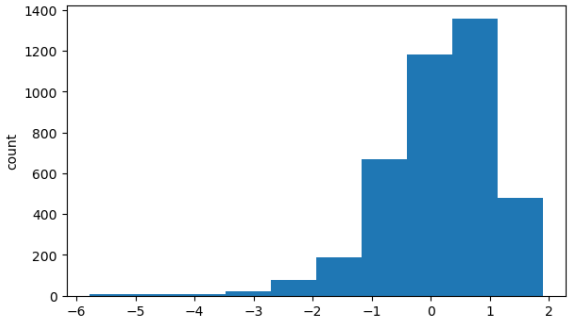


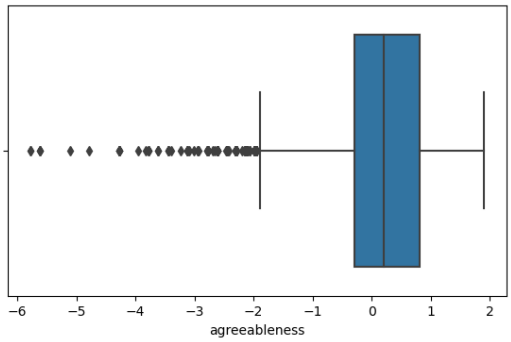
conscientiousness
Skew : -0.53



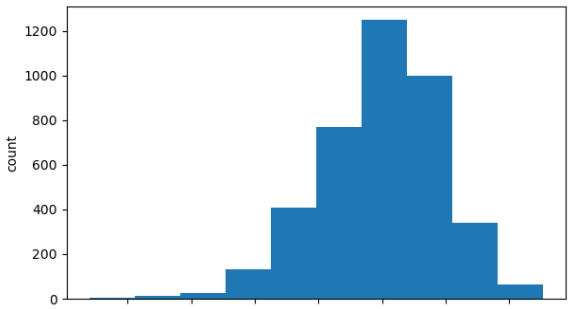


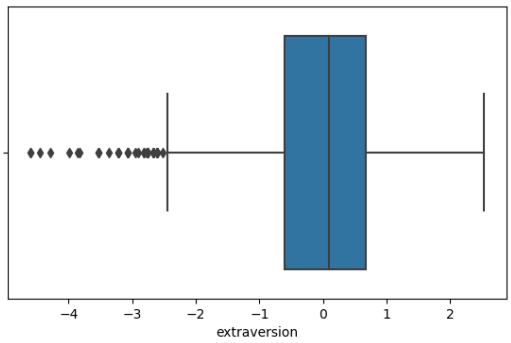
agreeableness
Skew : -1.2



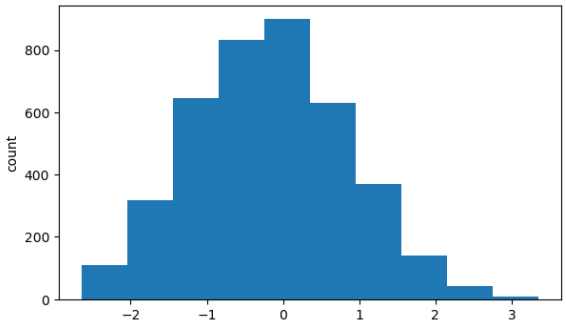


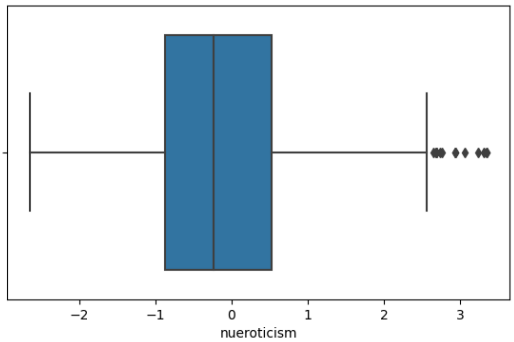
extraversion
Skew : -0.52



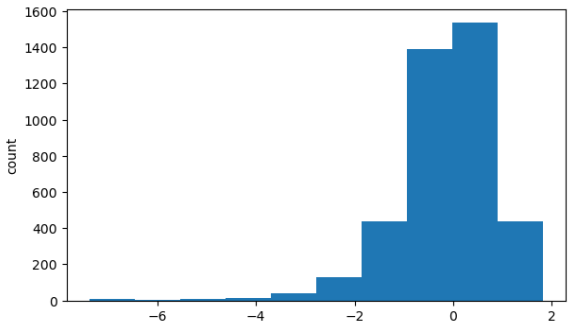


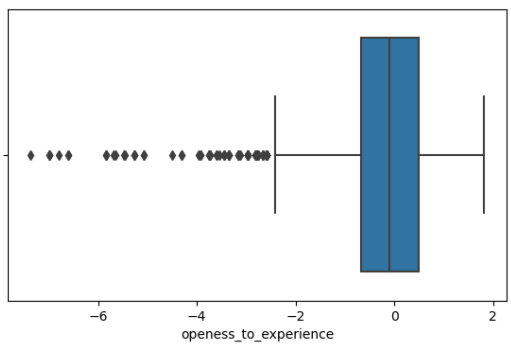
nueroticism
Skew : 0.17





openess_to_experience
Skew : -1.51

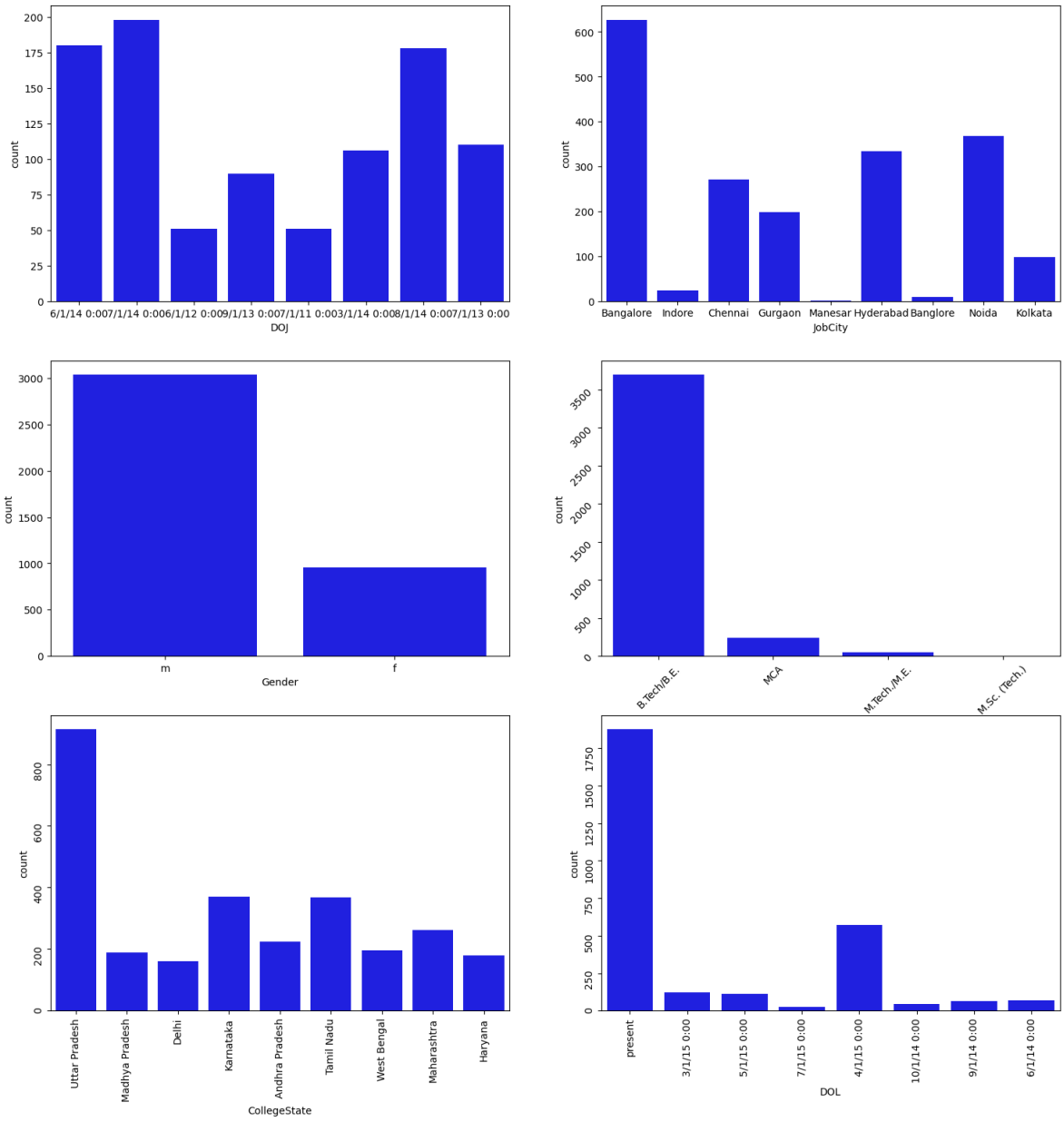




```
The skew() function typically refers to a statistical function that
calculates the skewness of a dataset
```

In [105]:
```
fig, axes = plt.subplots(3, 2, figsize = (18, 18))
fig.suptitle('Bar plot for all categorical variables in the dataset')
sns.countplot(ax = axes[0, 0], x = 'DOJ', data = data, color = 'blue',
              order = data['DOJ'].head(10).value_counts().index);
sns.countplot(ax = axes[0, 1], x = 'JobCity', data = data, color = 'blue',
              order = data['JobCity'].head(10).value_counts().index);
sns.countplot(ax = axes[1, 0], x = 'Gender', data = data, color = 'blue',
              order = data['Gender'].value_counts().index);
sns.countplot(ax = axes[1, 1], x = 'Degree', data = data, color = 'blue',
              order = data['Degree'].value_counts().index);
sns.countplot(ax = axes[2, 0], x = 'CollegeState', data = data, color = 'blue
              order = data['CollegeState'].head(20).value_counts().index);
sns.countplot(ax = axes[2, 1], x = 'DOL', data = data, color = 'blue',
              order = data['DOL'].head(20).value_counts().index);
axes[1][1].tick_params(labelrotation=45);
axes[2][0].tick_params(labelrotation=90);
axes[2][1].tick_params(labelrotation=90);
```

Bar plot for all categorical variables in the dataset



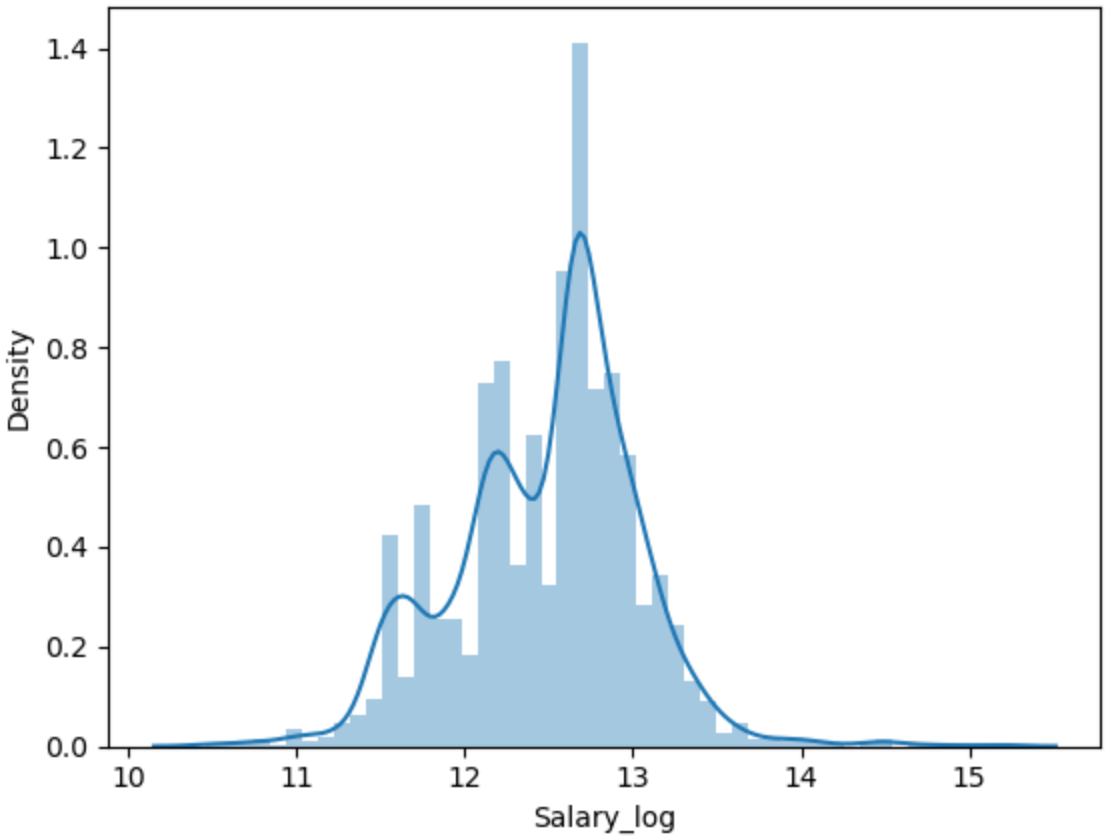
These are the Histogram plots for Numerical columns

In [106]:
```
# Function for log transformation of the column
def log_transform(data,col):
    for colname in col:
        if (data[colname] == 1.0).all():
            data[colname + '_log'] = np.log(data[colname]+1)
        else:
            data[colname + '_log'] = np.log(data[colname])
    data.info()
```

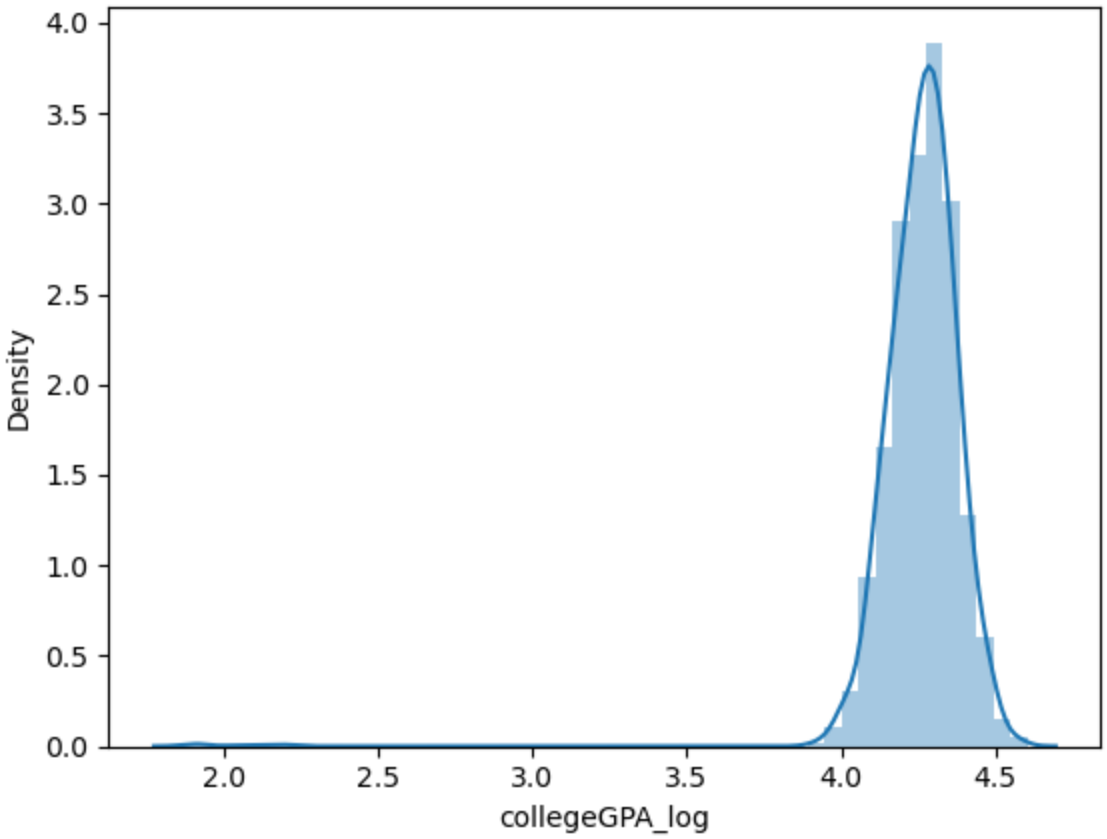
In [107]:
```
log_transform(data,['Salary','collegeGPA'])
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3997 entries, 0 to 3997
Data columns (total 41 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Unnamed: 0             3997 non-null   object 
 1   ID                     3997 non-null   int64  
 2   Salary                 3997 non-null   float64
 3   DOJ                    3997 non-null   object 
 4   DOL                    3997 non-null   object 
 5   Designation            3997 non-null   object 
 6   JobCity                3997 non-null   object 
 7   Gender                 3997 non-null   object 
 8   DOB                    3997 non-null   object 
 9   10percentage           3997 non-null   float64
 10  10board                3997 non-null   object 
 11  12graduation           3997 non-null   int64  
 12  12percentage           3997 non-null   float64
 13  12board                3997 non-null   object 
 14  CollegeID              3997 non-null   int64  
 15  CollegeTier            3997 non-null   int64  
 16  Degree                 3997 non-null   object 
 17  Specialization         3997 non-null   object 
 18  collegeGPA             3997 non-null   float64
 19  CollegeCityID          3997 non-null   int64  
 20  CollegeCityTier        3997 non-null   int64  
 21  CollegeState           3997 non-null   object 
 22  GraduationYear         3997 non-null   int64  
 23  English                3997 non-null   int64  
 24  Logical                3997 non-null   int64  
 25  Quant                  3997 non-null   int64  
 26  Domain                 3997 non-null   float64
 27  ComputerProgramming    3997 non-null   int64  
 28  ElectronicsAndSemicon  3997 non-null   int64  
 29  ComputerScience        3997 non-null   int64  
 30  MechanicalEngg         3997 non-null   int64  
 31  ElectricalEngg         3997 non-null   int64  
 32  TelecomEngg            3997 non-null   int64  
 33  CivilEngg              3997 non-null   int64  
 34  conscientiousness      3997 non-null   float64
 35  agreeableness          3997 non-null   float64
 36  extraversion           3997 non-null   float64
 37  nueroticism            3997 non-null   float64
 38  openess_to_experience  3997 non-null   float64
 39  Salary_log             3997 non-null   float64
 40  collegeGPA_log         3997 non-null   float64
dtypes: float64(12), int64(17), object(12)
memory usage: 1.3+ MB
```

In [108]:
```
#Log transformation of the feature 'Salary'
sns.distplot(data["Salary_log"], axlabel="Salary_log");
```

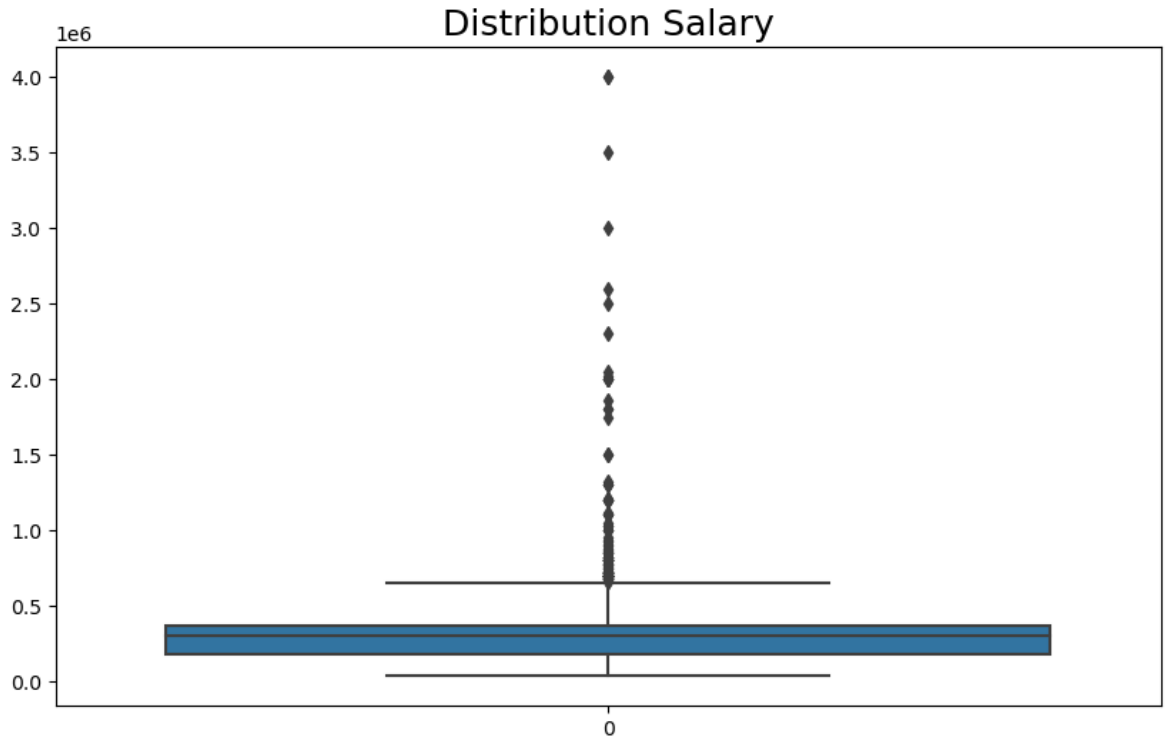


In [109]:
```
#Log transformation of the feature 'Price'
sns.distplot(data["collegeGPA_log"], axlabel="collegeGPA_log");
```



Boxplot

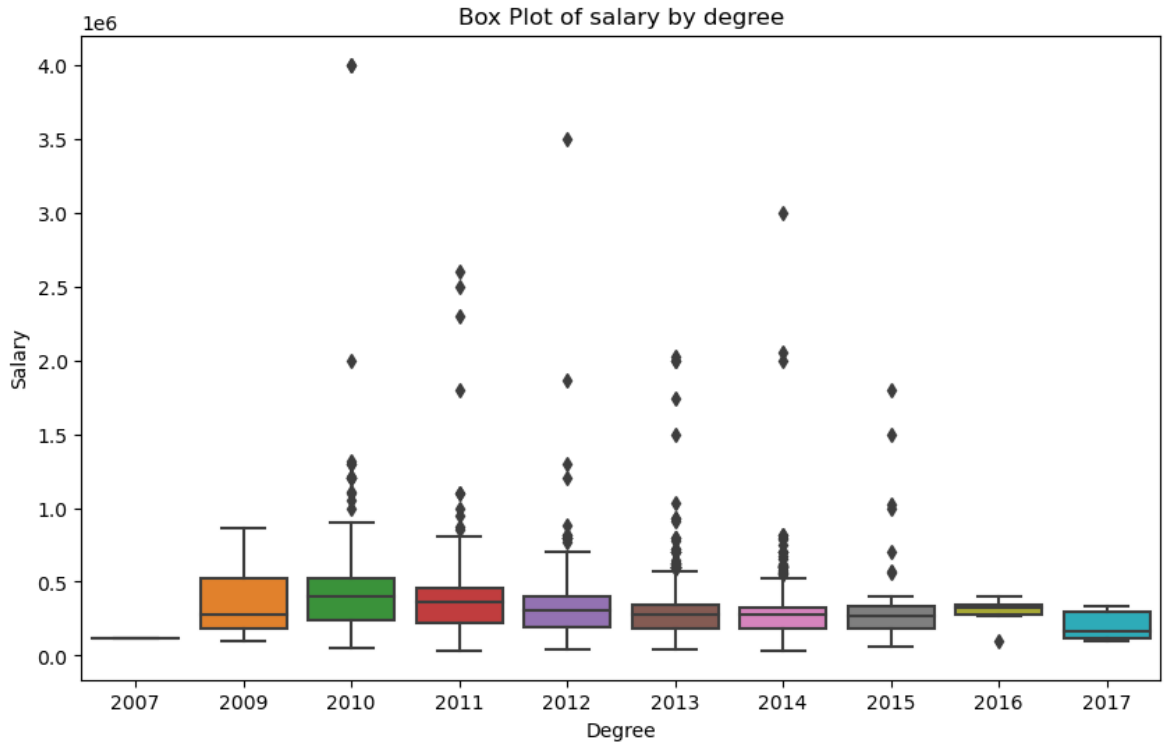
In [113]:
```
plt.figure(figsize = (10,6))
sns.boxplot(data.Salary)
plt.title('Distribution Salary',size=18)
plt.show()
```



In [114]:
```
data['Salary'].describe()
```

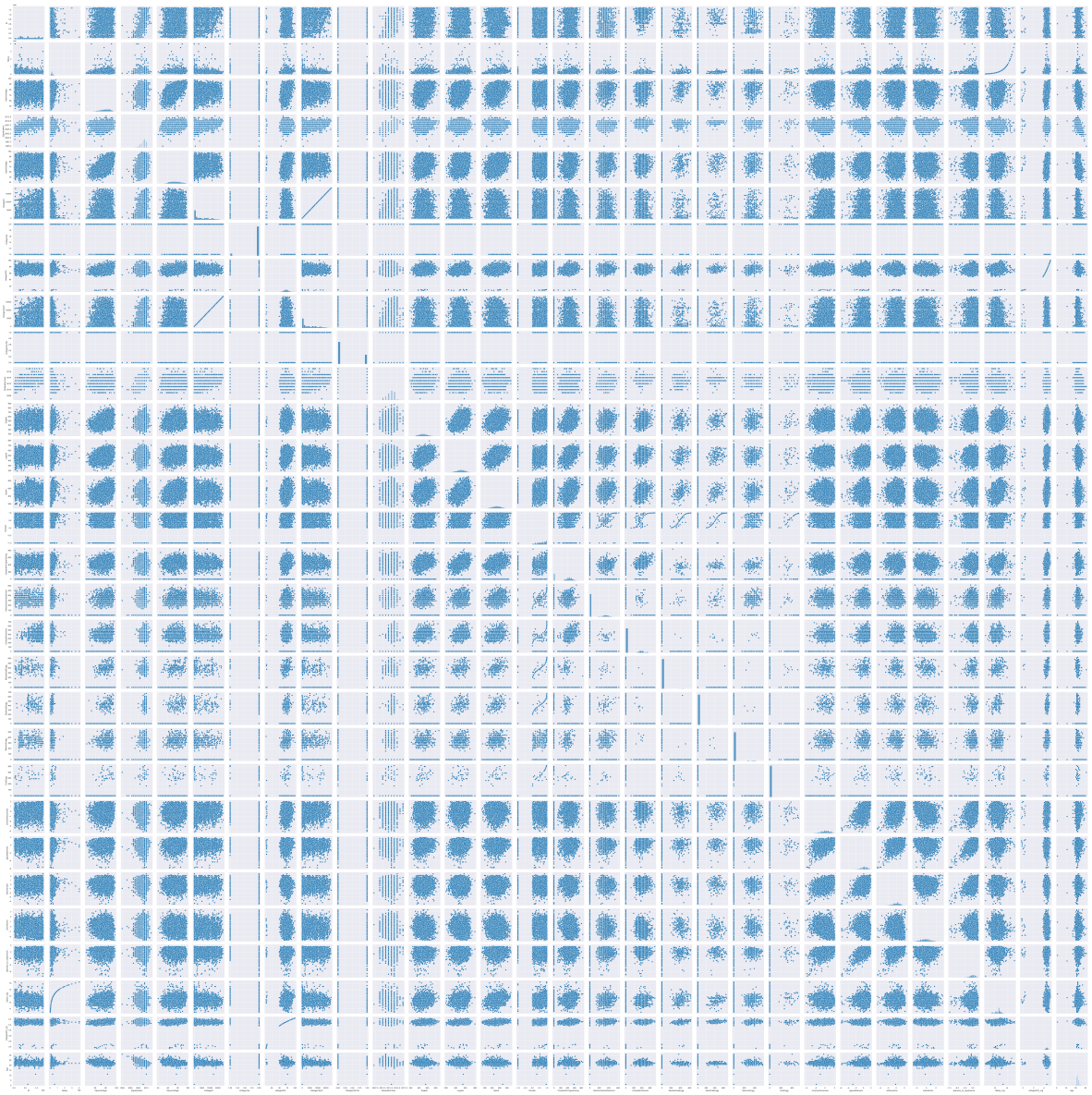
Out[114]:
```
count    3.997000e+03
mean     3.076955e+05
std      2.127639e+05
min      3.500000e+04
25%      1.800000e+05
50%      3.000000e+05
75%      3.700000e+05
max      4.000000e+06
Name: Salary, dtype: float64
```

In [115]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='GraduationYear', y='Salary')
plt.title('Box Plot of salary by degree')
plt.xlabel('Degree')
plt.ylabel('Salary')
plt.show()
```



In [140]:
```
plt.figure(figsize=(13,17))
sns.pairplot(data=data.drop(['Specialization','Degree'],axis=1))
plt.show()
```

```
<Figure size 1300x1700 with 0 Axes>
```

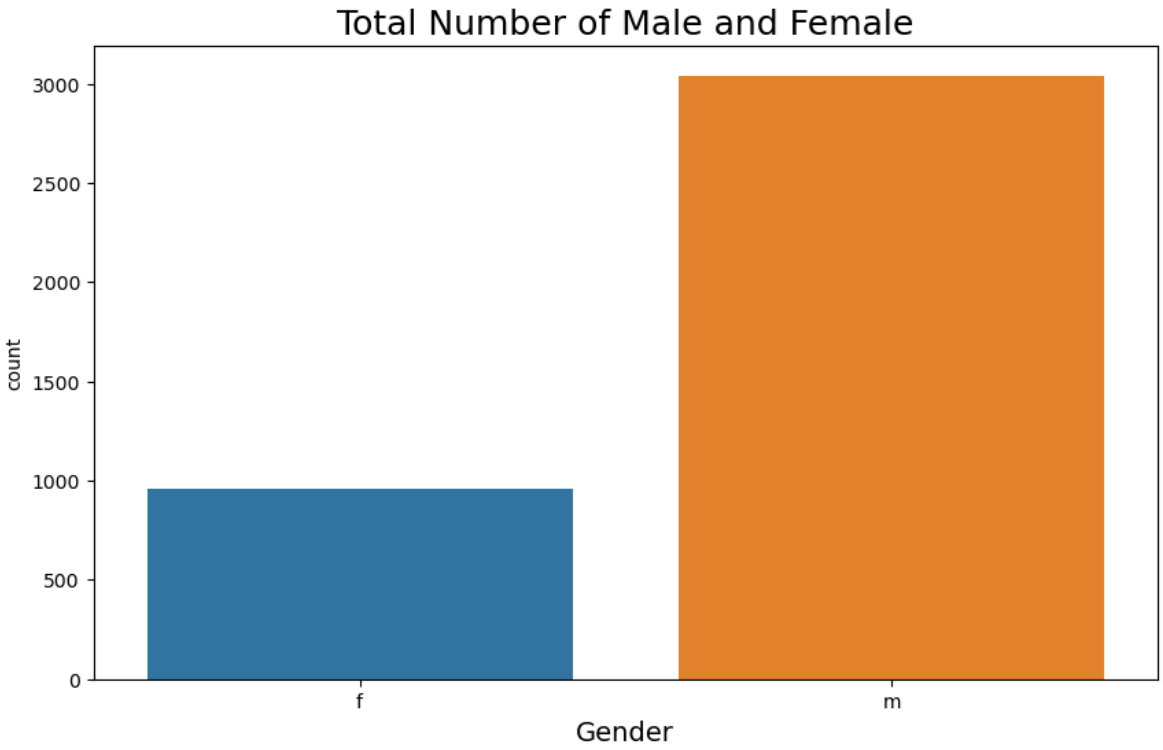


In [117]:
```
data['10percentage'].describe()
```

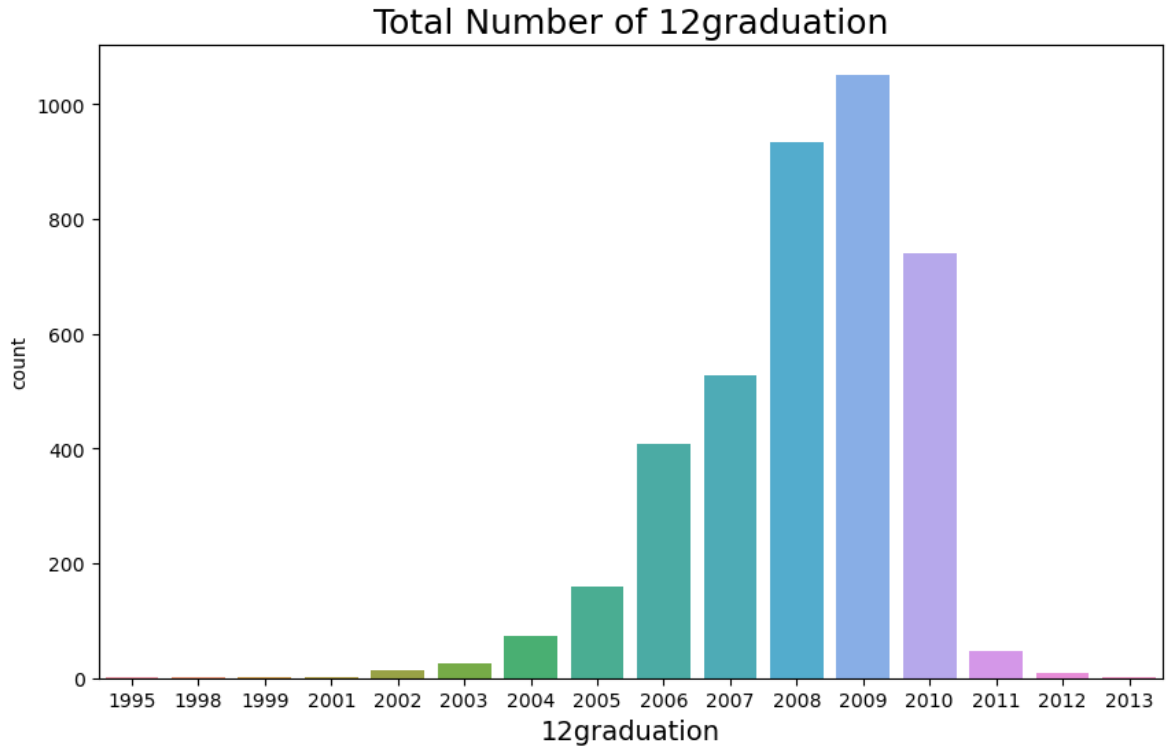
Out[117]:
```
count    3997.000000
mean       77.922672
std         9.849837
min        43.000000
25%        71.670000
50%        79.140000
75%        85.670000
max        97.760000
Name: 10percentage, dtype: float64
```

Countplot

In [118]:
```
plt.figure(figsize=(10,6))
sns.countplot(x = 'Gender', data = data)
plt.title('Total Number of Male and Female',size=18)
plt.xlabel('Gender',size=14)
plt.show()
```



In [119]:
```
plt.figure(figsize=(10,6))
sns.countplot(x = '12graduation', data = data)
plt.title('Total Number of 12graduation',size=18)
plt.xlabel('12graduation',size=14)
plt.show()
```



In [120]:
```
data['12graduation'].value_counts()
```

Out[120]:
```
12graduation
2009    1052
2008     935
2010     741
2007     528
2006     407
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
```

In [121]:
```
data['10board'].value_counts()
```

Out[121]:
```
10board
cbse                          1394
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: count, Length: 275, dtype: int64
```

In [122]:
```
data['12board'].value_counts()
```

Out[122]:
```
12board
cbse                                1399
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: count, Length: 340, dtype: int64
```

In [123]:
```
from datetime import datetime
```

In [124]:
```
datetime.now()
```

Out[124]:
```
datetime.datetime(2024, 2, 23, 19, 4, 45, 822155)
```

In [125]:
```
data['DOJ']=pd.to_datetime(data['DOJ'])
```

In [126]:
```
data['DOB']=pd.to_datetime(data['DOB'])
```

In [127]:
```
data['DOL']=data['DOL'].replace('present',pd.Timestamp.now())
```

In [128]:
```
data['DOL']=pd.to_datetime(data['DOL'])
```

In [129]:
```
data['DOJ']=pd.to_datetime(data['DOJ']).dt.date
data['DOL']=pd.to_datetime(data['DOL']).dt.date
```

In [130]:
```
data['DOJ']=pd.to_datetime(data['DOJ'])
data['DOL']=pd.to_datetime(data['DOL'])
```

In [131]:
```
data['Age']=data['DOJ']-data['DOB']
```

In [132]:
```
data['Age']=(data['Age']//365).astype('str')
```

In [133]:
```
import re
data['Age']=data['Age'].apply(lambda x: int(re.findall(r'[0-9]+',x)[0]))
```

In [134]:
```
data['Age']
```

Out[134]:
```
0       22
1       23
2       21
3       21
4       23
        ..
3993    24
3994    20
3995    22
3996    22
3997    21
Name: Age, Length: 3997, dtype: int64
```

Outliers

In [135]:
```
data[(data['Salary']< Q1-1.5* IQR) | (data['Salary']> Q3+1.5* IQR)]
```

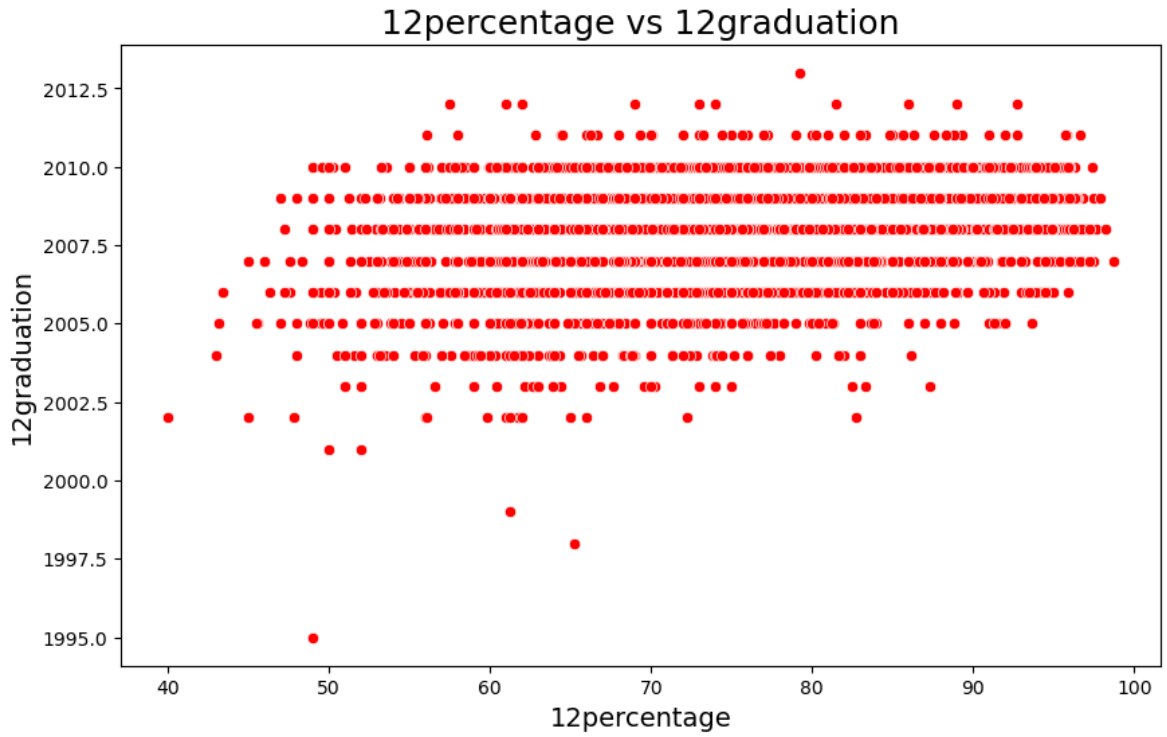
Out[135]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB
0	train	203097	420000.0	2012-06-01	2024-02-23	senior quality engineer	Bangalore	f	1990-02-19
1	train	579905	500000.0	2013-09-01	2024-02-23	assistant manager	Indore	m	1989-10-04
2	train	810601	325000.0	2014-06-01	2024-02-23	systems engineer	Chennai	f	1992-08-03
3	train	267447	1100000.0	2011-07-01	2024-02-23	senior software engineer	Gurgaon	m	1989-12-05
4	train	343523	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27
...	...	...	...	...	...	...	...	...	...
3993	train	47916	280000.0	2011-10-01	2012-10-01	software engineer	New Delhi	m	1987-04-15
3994	train	752781	100000.0	2013-07-01	2013-07-01	technical writer	Hyderabad	f	1992-08-27
3995	train	355888	320000.0	2013-07-01	2024-02-23	associate software engineer	Bangalore	m	1991-07-03
3996	train	947111	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20
3997	train	324966	400000.0	2013-02-01	2024-02-23	senior systems engineer	Chennai	f	1991-02-26

3997 rows × 42 columns

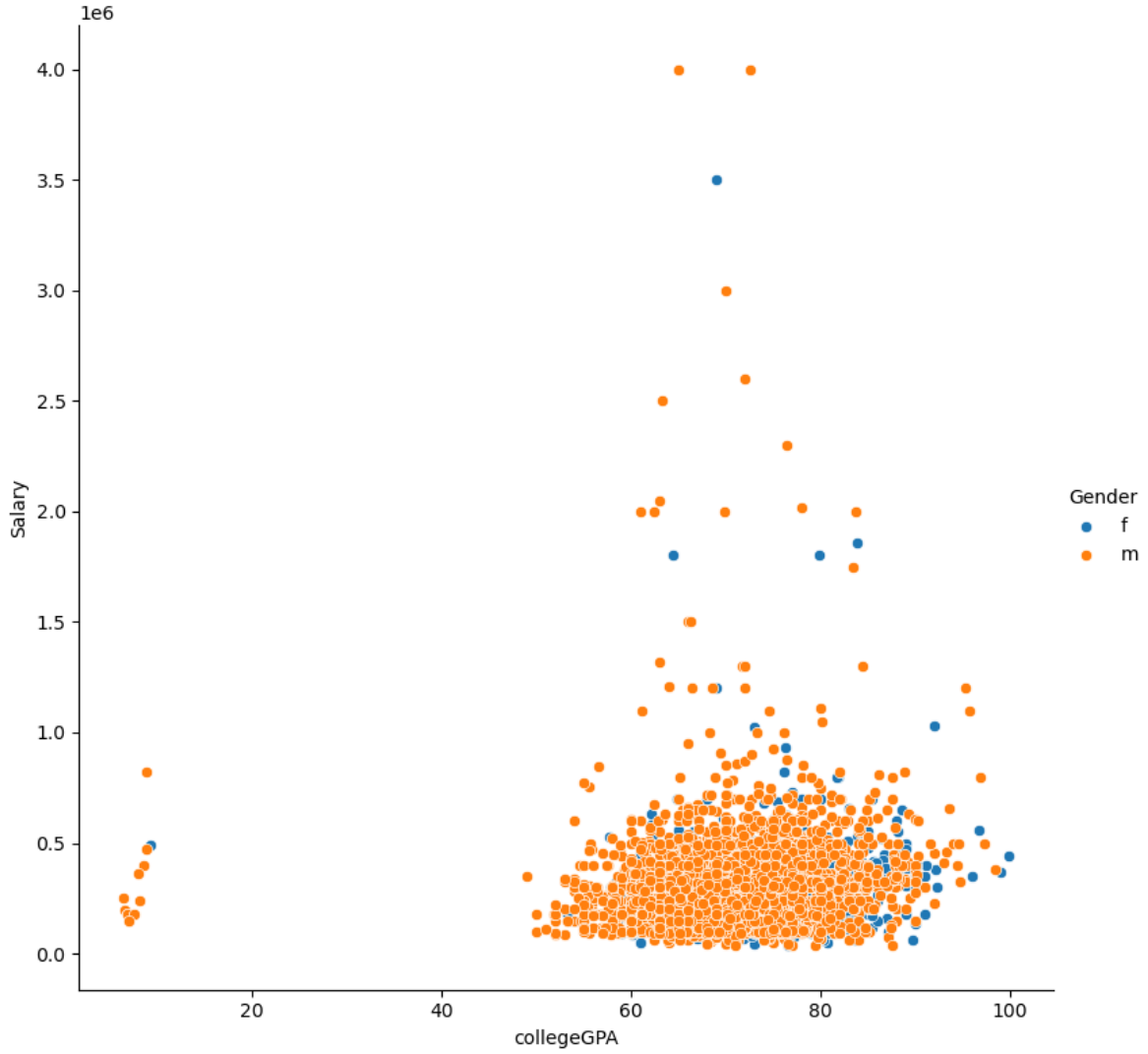
Scatterplot

In [136]:
```
plt.figure(figsize = (10,6))
sns.scatterplot(x='12percentage',y='12graduation',color='r',data=data)
plt.title('12percentage vs 12graduation',size=18)
plt.xlabel('12percentage',size=14)
plt.ylabel('12graduation',size=14)
plt.show()
```

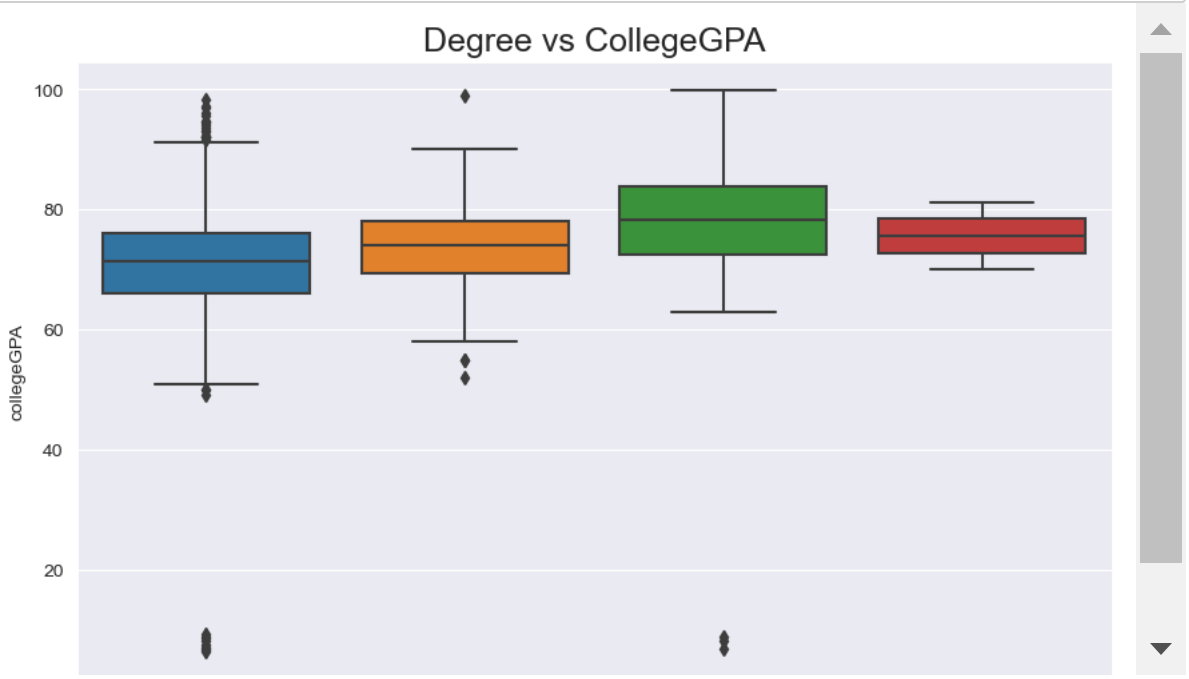


Boxplot

In [137]:
```
 sns.FacetGrid(data=data, hue='Gender', height=8) \
.map(sns.scatterplot, 'collegeGPA', 'Salary') \
.add_legend()
plt.show()
```



In [138]:
```
plt.figure(figsize = (10,6))
sns.set_style('darkgrid')
sns.boxplot(x='Degree',y='collegeGPA',data=data)
plt.title('Degree vs CollegeGPA',size=18);
```



In [139]:
```
data_numeric = data.apply(pd.to_numeric, errors='coerce')
data_numeric.drop(['CollegeTier', 'CollegeID'], axis=1, inplace=True)
plt.figure(figsize=(12, 7))
sns.heatmap(data_numeric.corr(), annot=True, vmin=-1, vmax=1)
plt.show()
```