EDA PROJECT

import all required packages

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

In [2]:
```
path=r"C:\Users\Sruth\Downloads\data.xlsx - Sheet1.csv"
df=pd.read_csv(path)
df
```

Out[2]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	G
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	

3998 rows × 39 columns

shape

which determines how many rows and columns present in dataframe

In [3]:
```
# there are 3998 rows and 39 columns present in data frame

df.shape
```

Out[3]: (3998, 39)

size

In [4]:
```
# size gives over all elements in data set

df.size
```

Out[4]: 155922

head

In [5]:
```
# by default head gives first 5 rows of dataframe

df.head()
```

Out[5]:

	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 0
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4 0
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3 0
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5 0
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27 0

5 rows × 39 columns

tail

In [6]:
```
# tail gives last five rows of dataframe

df.tail()
```

Out[6]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Ge
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	

5 rows × 39 columns

dtypes -->checking the columns that are which types -->categorical columns(object) -->numerical columns(int,float)

In [7]:
```
df.dtypes
```

Out[7]:
```
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
```

Checking whether missing values are present

In [8]:
```
# there is no missing values are present in data
df.isnull().sum()
```

Out[8]:
```
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
```

categorical columns

In [9]:
```
cat_cols=df.select_dtypes(include='object').columns
cat_cols
```

Out[9]:
```
Index(['Unnamed: 0', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10board', '12board', 'Degree', 'Specialization', 'CollegeState'],
      dtype='object')
```

In [10]:
```
# there are 12 categorical columns

len(cat_cols)
```

Out[10]:
```
12
```

numerical columns

In [11]:
```
num_cols=df.select_dtypes(exclude='object').columns
num_cols
```

Out[11]:
```
Index(['ID', 'Salary', '10percentage', '12graduation', '12percentage',
       'CollegeID', 'CollegeTier', 'collegeGPA', 'CollegeCityID',
       'CollegeCityTier', 'GraduationYear', 'English', 'Logical', 'Quant',
       'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience'],
      dtype='object')
```

In [12]:
```
# there are 26 numerical columns

len(num_cols)
```

Out[12]:
```
27
```

Univariate_analysis

Here dropped an ID column which contain unqiue items

In [13]:
```
numerical_cols=df.drop(columns=['ID'])
numerical_cols
```

Out[13]:

	Unnamed: 0	Salary	DOJ	DOL	Designation	JobCity	Gender	
0	train	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2
1	train	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	1
2	train	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	
3	train	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	1
4	train	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2
...	...	...	...	...	...	...	...	
3993	train	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4
3994	train	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8
3995	train	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	
3996	train	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3
3997	train	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2

3998 rows × 38 columns

describe

In [14]:
```
numerical_cols.describe()
```

Out[14]:

	Salary	10percentage	12graduation	12percentage	CollegeID	College
count	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	3998.00
mean	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	1.92
std	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	0.26
min	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	1.00
25%	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	2.00
50%	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	2.00
75%	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	2.00
max	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	2.00

8 rows × 26 columns

skewness for each numeric column

In [15]:
```
for i in num_cols[1:]:
    skew=round(df[i].skew(),2)
    print(f"skew of  column {i} is '{skew}'")
```

```
skew of  column Salary is '6.45'
skew of  column 10percentage is '-0.59'
skew of  column 12graduation is '-0.96'
skew of  column 12percentage is '-0.03'
skew of  column CollegeID is '0.65'
skew of  column CollegeTier is '-3.25'
skew of  column collegeGPA is '-1.25'
skew of  column CollegeCityID is '0.65'
skew of  column CollegeCityTier is '0.87'
skew of  column GraduationYear is '-63.07'
skew of  column English is '0.19'
skew of  column Logical is '-0.22'
skew of  column Quant is '-0.02'
skew of  column Domain is '-1.92'
skew of  column ComputerProgramming is '-0.78'
skew of  column ElectronicsAndSemicon is '1.2'
skew of  column ComputerScience is '1.53'
skew of  column MechanicalEngg is '4.03'
skew of  column ElectricalEngg is '5.06'
skew of  column TelecomEngg is '3.04'
skew of  column CivilEngg is '10.32'
skew of  column conscientiousness is '-0.53'
skew of  column agreeableness is '-1.2'
skew of  column extraversion is '-0.52'
skew of  column nueroticism is '0.17'
skew of  column openess_to_experience is '-1.51'
```

kurtosis for each numeric column

In [16]:
```
for i in num_cols[1:]:
    kurt=round(df[i].kurt(),2)
    print(f"kurtosis of  column {i} is '{kurt}'")
```

```
kurtosis of  column Salary is '80.93'
kurtosis of  column 10percentage is '-0.11'
kurtosis of  column 12graduation is '1.95'
kurtosis of  column 12percentage is '-0.63'
kurtosis of  column CollegeID is '-0.77'
kurtosis of  column CollegeTier is '8.55'
kurtosis of  column collegeGPA is '10.23'
kurtosis of  column CollegeCityID is '-0.77'
kurtosis of  column CollegeCityTier is '-1.24'
kurtosis of  column GraduationYear is '3984.37'
kurtosis of  column English is '-0.25'
kurtosis of  column Logical is '-0.22'
kurtosis of  column Quant is '-0.1'
kurtosis of  column Domain is '3.9'
kurtosis of  column ComputerProgramming is '-0.67'
kurtosis of  column ElectronicsAndSemicon is '-0.21'
kurtosis of  column ComputerScience is '0.69'
kurtosis of  column MechanicalEngg is '15.02'
kurtosis of  column ElectricalEngg is '24.88'
kurtosis of  column TelecomEngg is '7.81'
kurtosis of  column CivilEngg is '109.04'
kurtosis of  column conscientiousness is '0.12'
kurtosis of  column agreeableness is '3.39'
kurtosis of  column extraversion is '0.64'
kurtosis of  column nueroticism is '-0.19'
kurtosis of  column openess_to_experience is '5.79'
```

covariance matrix

In [17]:
```
numerical_cols.cov(numeric_only=True)
```

Out[17]:

	Salary	10percentage	12graduation	12percentage	
Salary	4.525724e+10	371684.456159	-56771.565543	398412.208861	-1.
10percentage	3.716845e+05	97.025700	4.397119	69.710710	9.
12graduation	-5.677157e+04	4.397119	2.734391	4.714096	2.
12percentage	3.984122e+05	69.710710	4.714096	120.998528	1.
CollegeID	-1.212565e+08	997.261052	2017.184639	1179.911862	2.
CollegeTier	-1.000579e+04	-0.325618	0.012009	-0.290720	8.
collegeGPA	2.260530e+05	25.143547	1.161489	31.096999	6.
CollegeCityID	-1.212565e+08	997.261052	2017.184639	1179.911862	2.
CollegeCityTier	1.500495e+03	0.527071	-0.002287	0.657967	1.
GraduationYear	-6.813186e+04	-4.330051	0.761563	-4.532195	-2.
English	3.978683e+06	362.593281	25.669208	245.744098	-1.
Logical	3.309785e+06	270.138109	15.195251	232.515001	-1.
Quant	6.000532e+06	382.660566	0.278868	420.294392	-6.
Domain	1.043457e+04	0.362683	-0.026476	0.382008	-1.
ComputerProgramming	5.053027e+06	108.421537	-16.297969	182.558565	-3.
ElectronicsAndSemicon	2.240082e+04	132.769360	-1.541392	203.850411	-1.
ComputerScience	-3.755551e+06	-32.687912	85.047828	-83.932520	8.
MechanicalEngg	3.856527e+05	48.678119	5.753526	40.621034	-4.
ElectricalEngg	-8.868903e+05	64.203771	17.923053	61.660745	9.
TelecomEngg	-5.061435e+05	50.998564	4.069324	50.980677	1.
CivilEngg	2.935306e+05	10.833465	-0.286551	2.383094	1.
conscientiousness	-1.403798e+04	0.685534	0.175763	0.659664	3.
agreeableness	1.150482e+04	1.267615	0.064135	1.077373	-2.
extraversion	-2.067186e+03	-0.043855	0.097479	-0.078347	2.
nueroticism	-1.172182e+04	-1.314998	-0.123908	-1.045919	-4.
openess_to_experience	-2.425976e+03	0.364343	-0.025119	0.070213	-5.

26 rows × 26 columns

correlation

In [18]:
```
correlation_data=df.corr(numeric_only=True)
correlation_data
```

Out[18]:

	ID	Salary	10percentage	12graduation	12percentage
ID	1.000000	-0.247294	0.044547	0.673102	0.00706
Salary	-0.247294	1.000000	0.177373	-0.161383	0.17025
10percentage	0.044547	0.177373	1.000000	0.269957	0.64337
12graduation	0.673102	-0.161383	0.269957	1.000000	0.25916
12percentage	0.007069	0.170254	0.643378	0.259166	1.00000
CollegeID	0.284540	-0.118690	0.021082	0.254021	0.02233
CollegeTier	0.035160	-0.179332	-0.126042	0.027691	-0.10077
collegeGPA	0.047144	0.130103	0.312538	0.086001	0.34613
CollegeCityID	0.284540	-0.118690	0.021082	0.254021	0.02233
CollegeCityTier	-0.035977	0.015384	0.116707	-0.003016	0.13046
GraduationYear	0.027539	-0.010053	-0.013799	0.014457	-0.01293
English	0.135505	0.178219	0.350780	0.147925	0.21288
Logical	0.102215	0.179275	0.316014	0.105887	0.24357
Quant	-0.055134	0.230627	0.317640	0.001379	0.31241
Domain	-0.125639	0.104656	0.078563	-0.034163	0.07409
ComputerProgramming	0.018859	0.115665	0.053600	-0.047995	0.08081
ElectronicsAndSemicon	-0.115601	0.000665	0.085179	-0.005891	0.11711
ComputerScience	0.482626	-0.100720	-0.018933	0.293439	-0.04353
MechanicalEngg	-0.026147	0.018475	0.050364	0.035459	0.03763
ElectricalEngg	0.104454	-0.047598	0.074419	0.123751	0.06400
TelecomEngg	-0.049272	-0.022691	0.049378	0.023470	0.04420
CivilEngg	-0.017871	0.037639	0.030002	-0.004727	0.00591
conscientiousness	0.175557	-0.064148	0.067657	0.103329	0.05829
agreeableness	0.024837	0.057423	0.136645	0.041182	0.10399
extraversion	0.120979	-0.010213	-0.004679	0.061956	-0.00748
nueroticism	-0.146289	-0.054685	-0.132496	-0.074369	-0.09436
openess_to_experience	0.031359	-0.011312	0.036692	-0.015069	0.00633

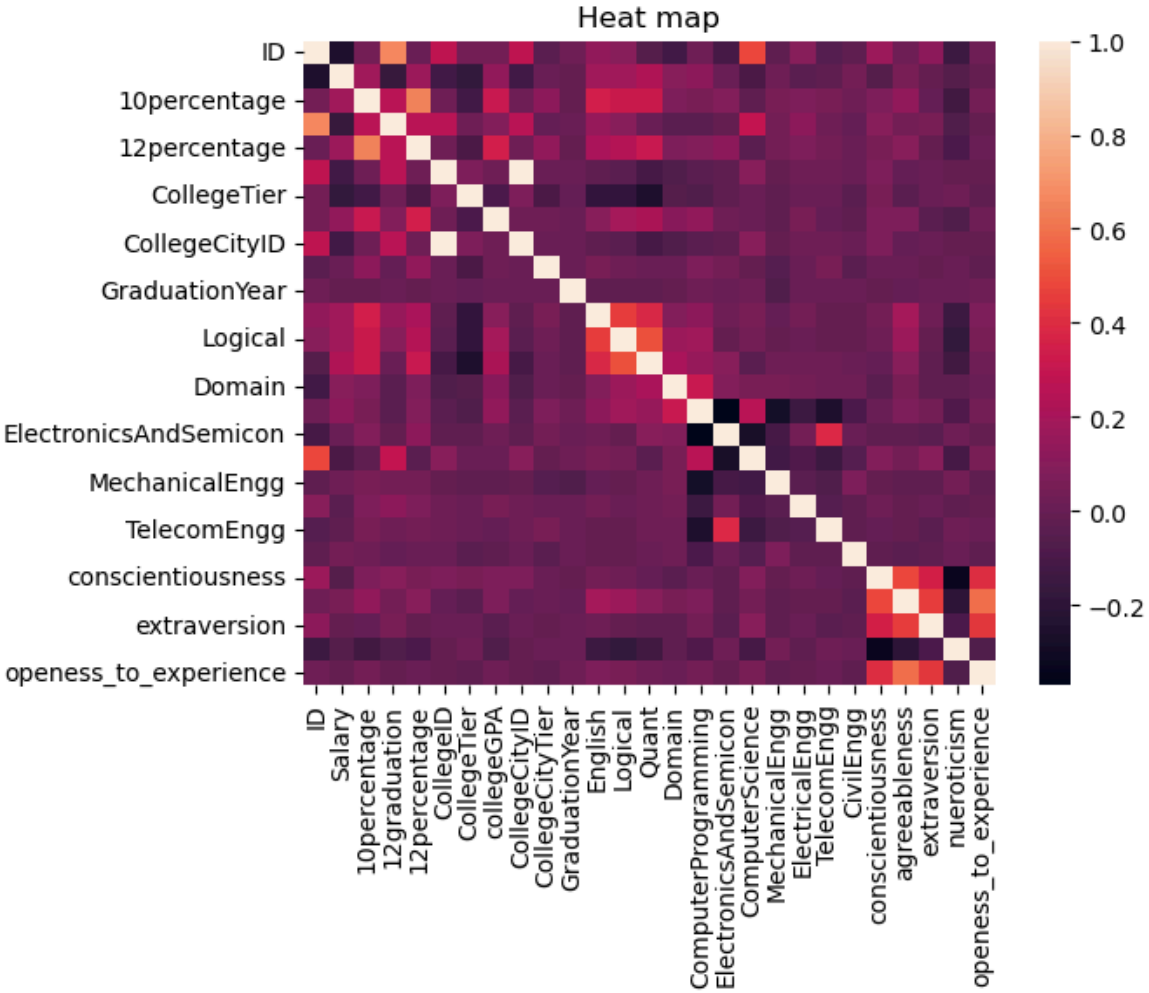
27 rows × 27 columns

Heat_map

- correlation tells about how much relation between two variables
- denotes with r , r varies **-1 to +1**

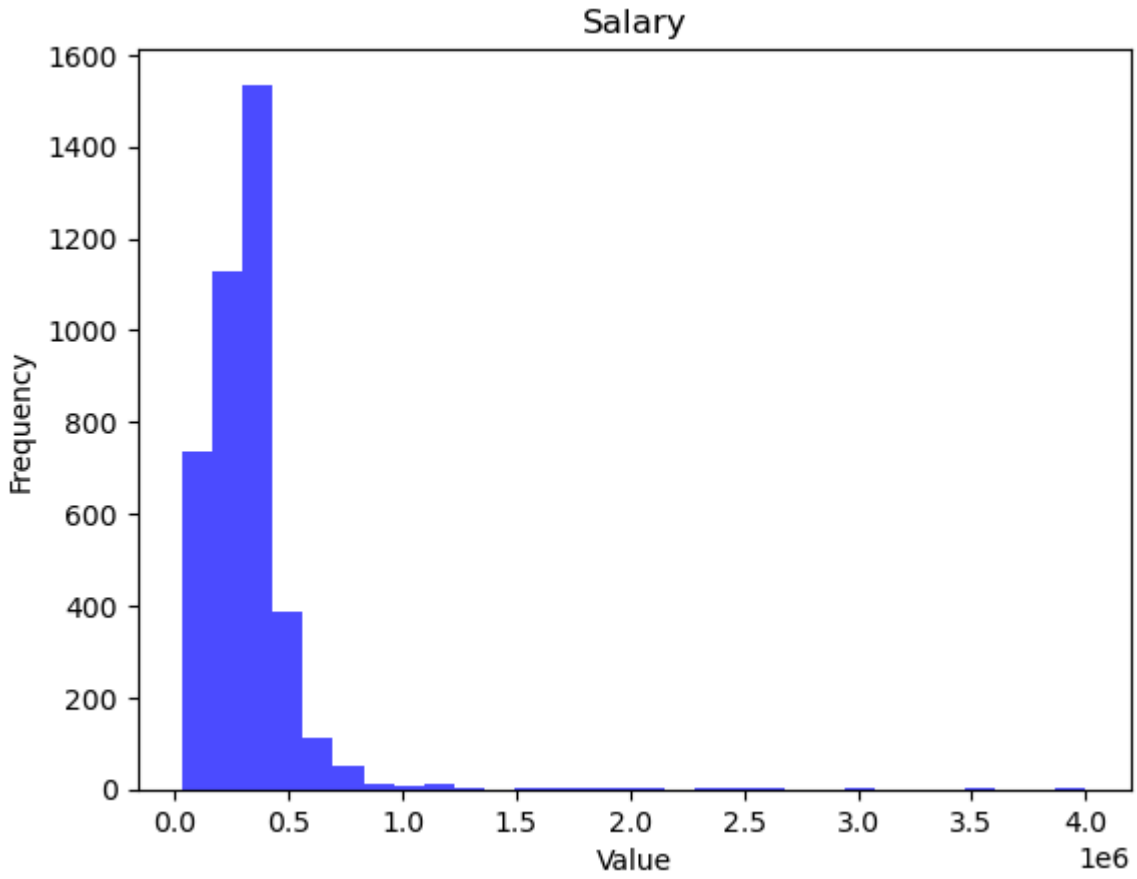
- **-1 to 0** indicates **negative relation**
- **0 to 1** indivates **positive relation**
- **0** indicates **no relation**

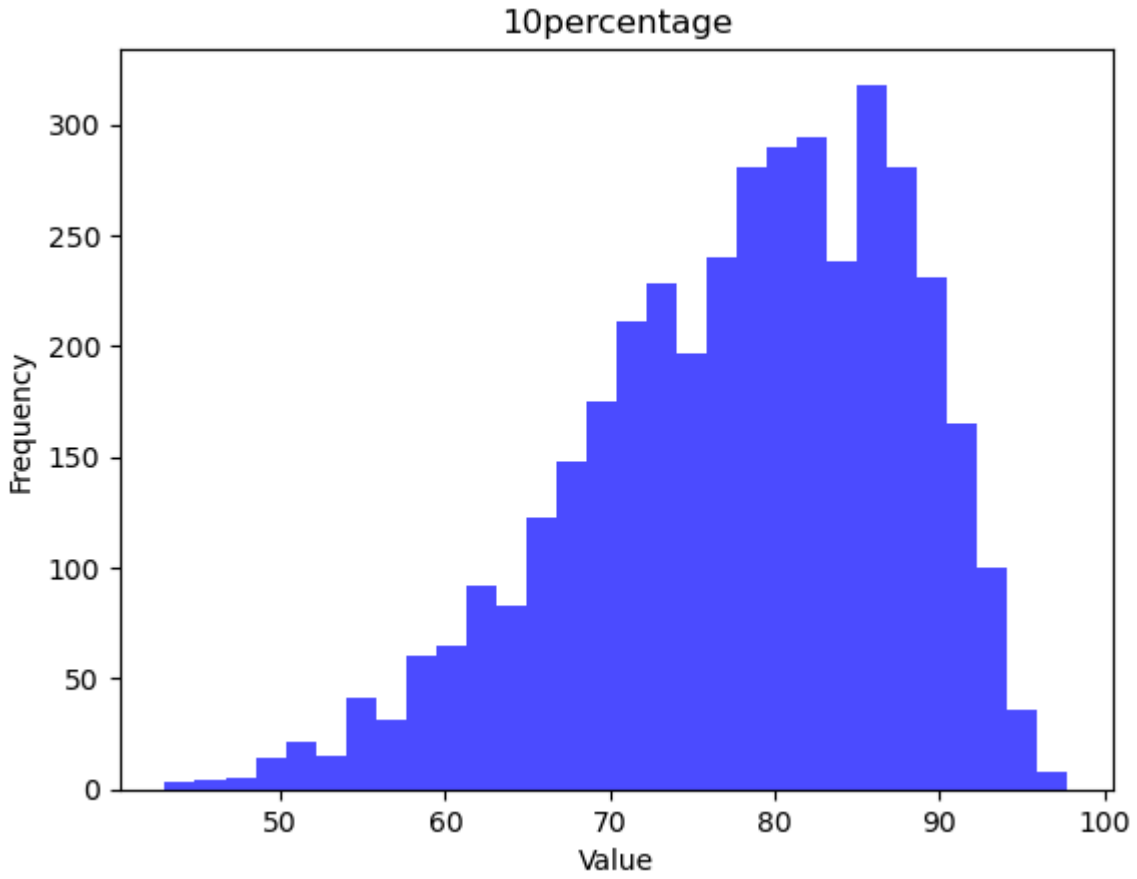
In [19]:
```
sns.heatmap(correlation_data)
plt.title('Heat map')
plt.show()
```

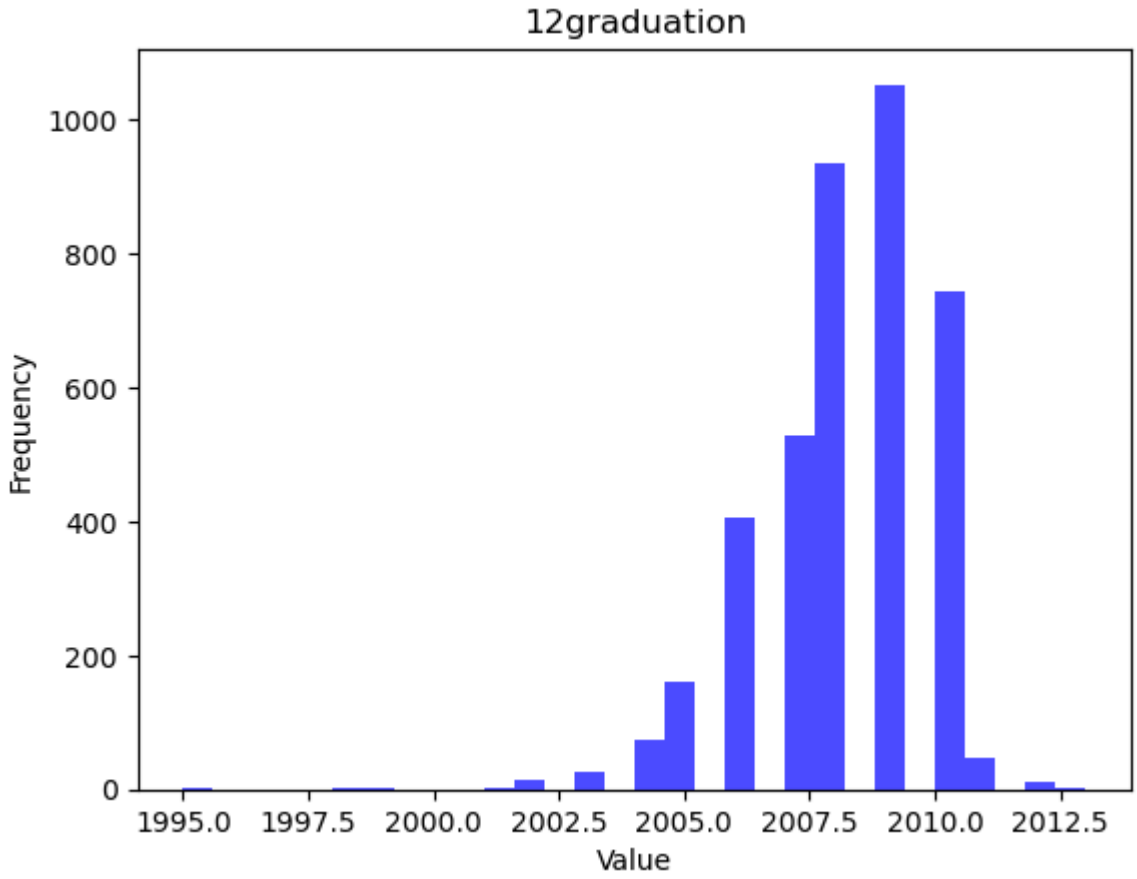


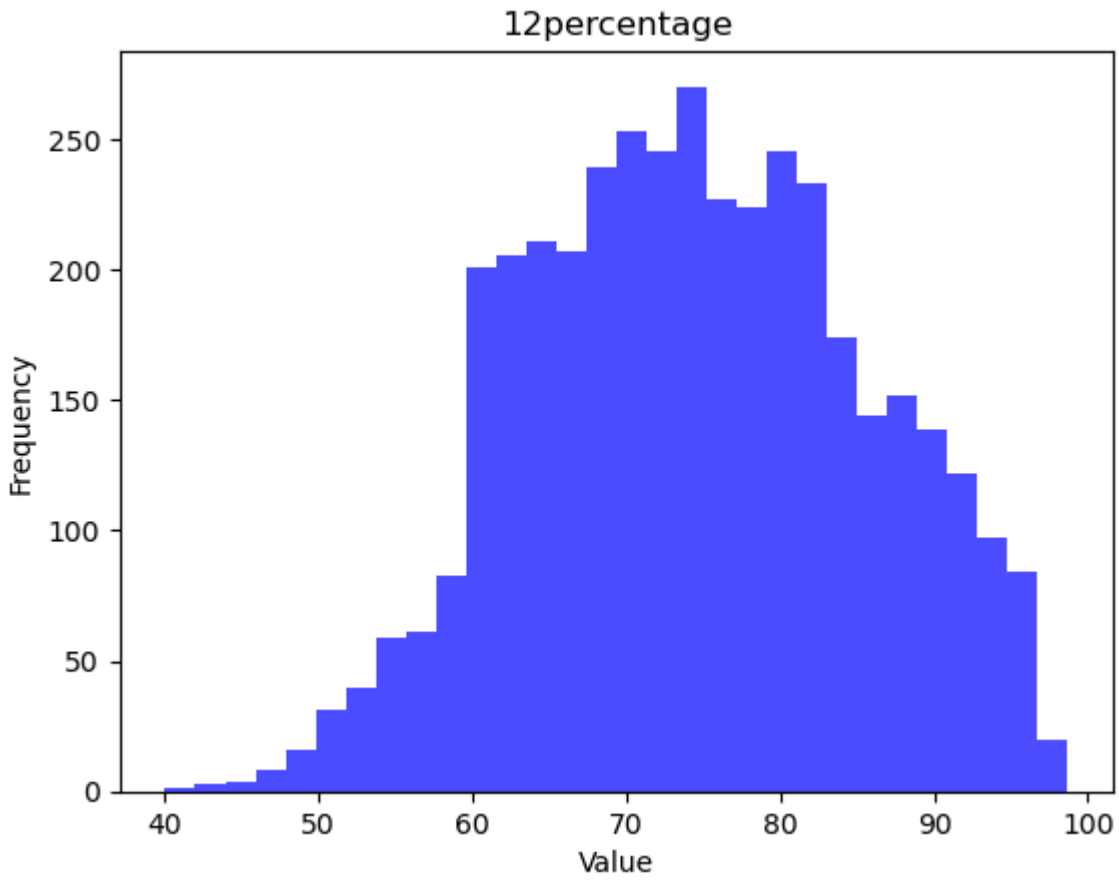
Histogram analysis for numerical columns

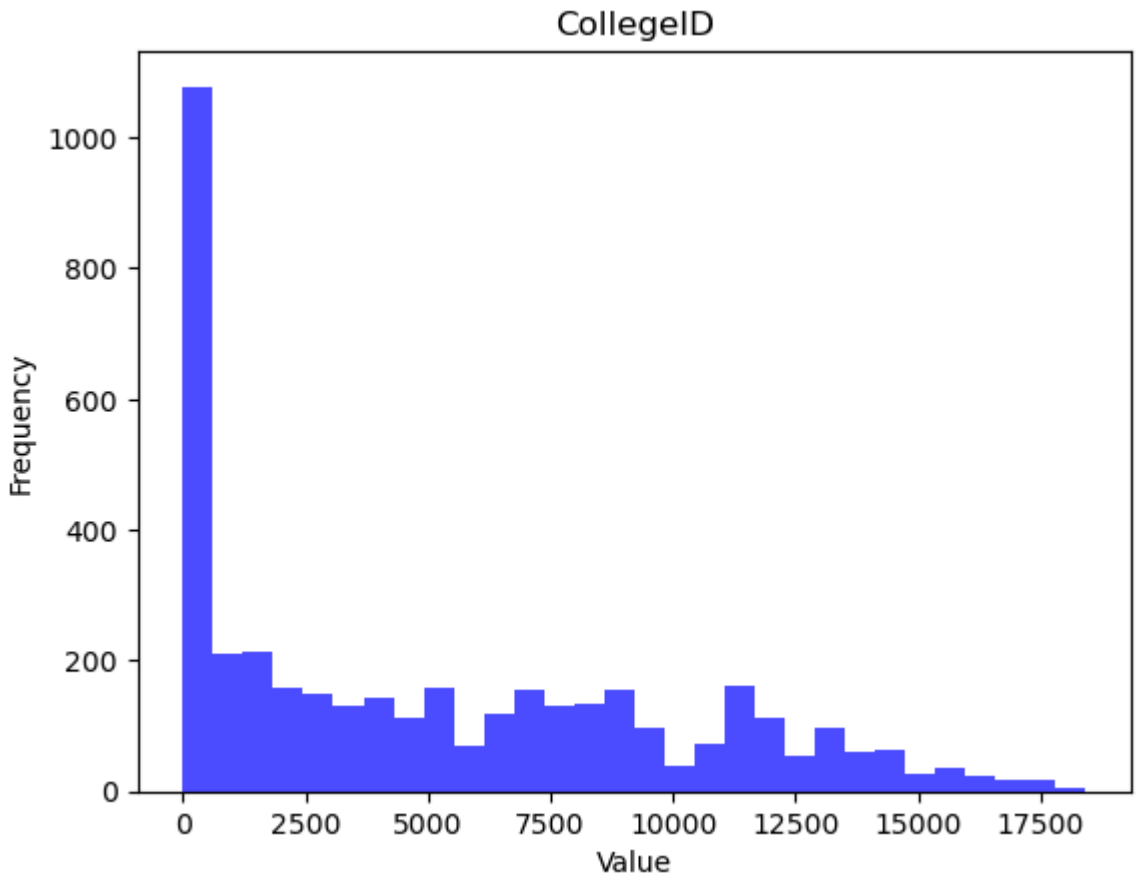
In [22]:
```
for i in num_cols[1:]:
    plt.hist(numerical_cols[i],bins=30,alpha=0.7,color='blue')
    plt.title(i)
    plt.xlabel('Value')
    plt.ylabel("Frequency")
    plt.show()
```

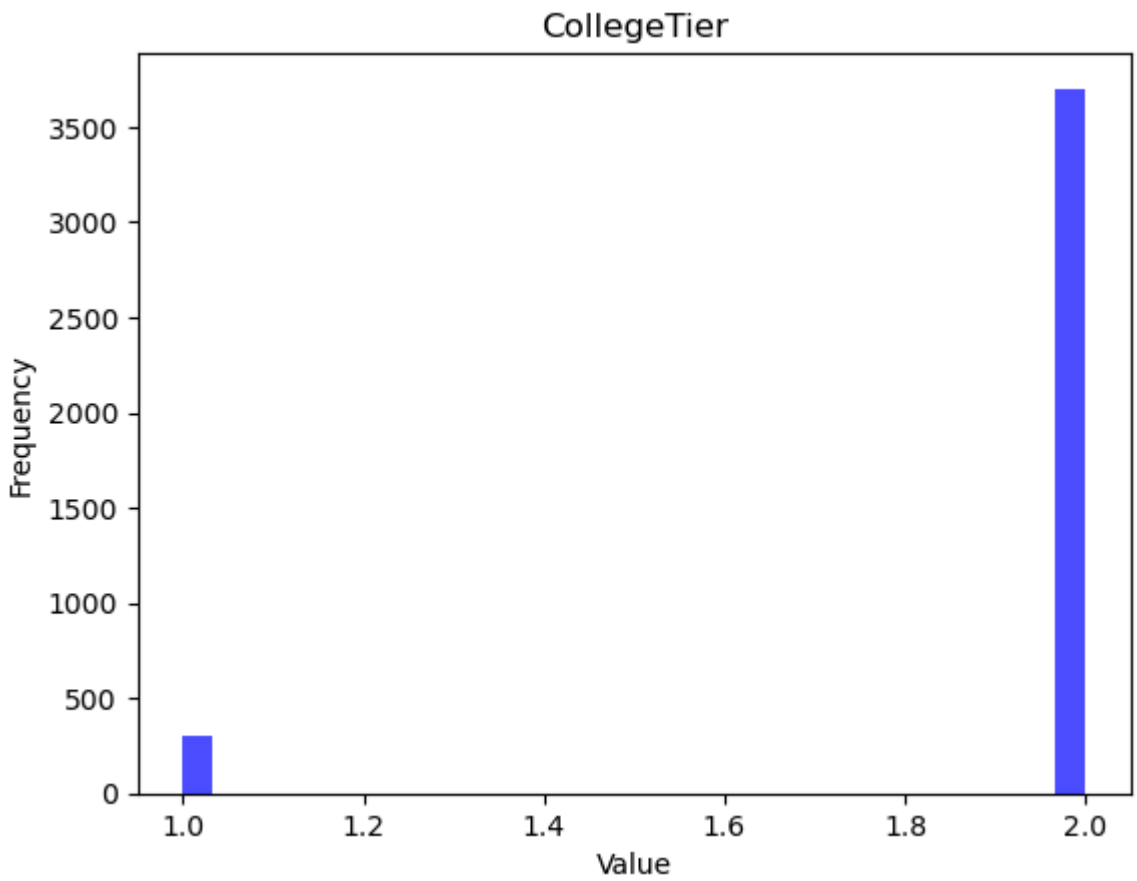

Salary
1600
1400
1200
1000
800
600
400
200
0
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Value
1e6

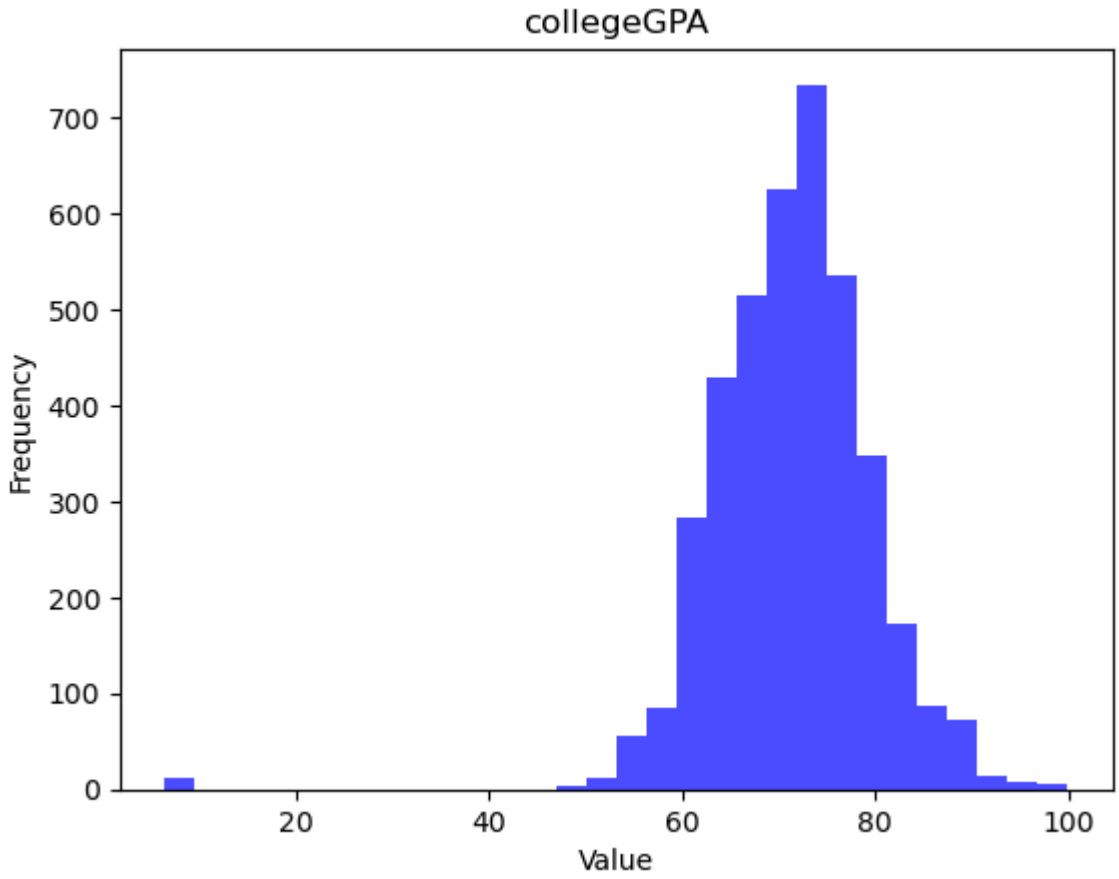

10percentage
300
250
200
150
100
50
0
Frequency
50
60
70
80
90
100
Value

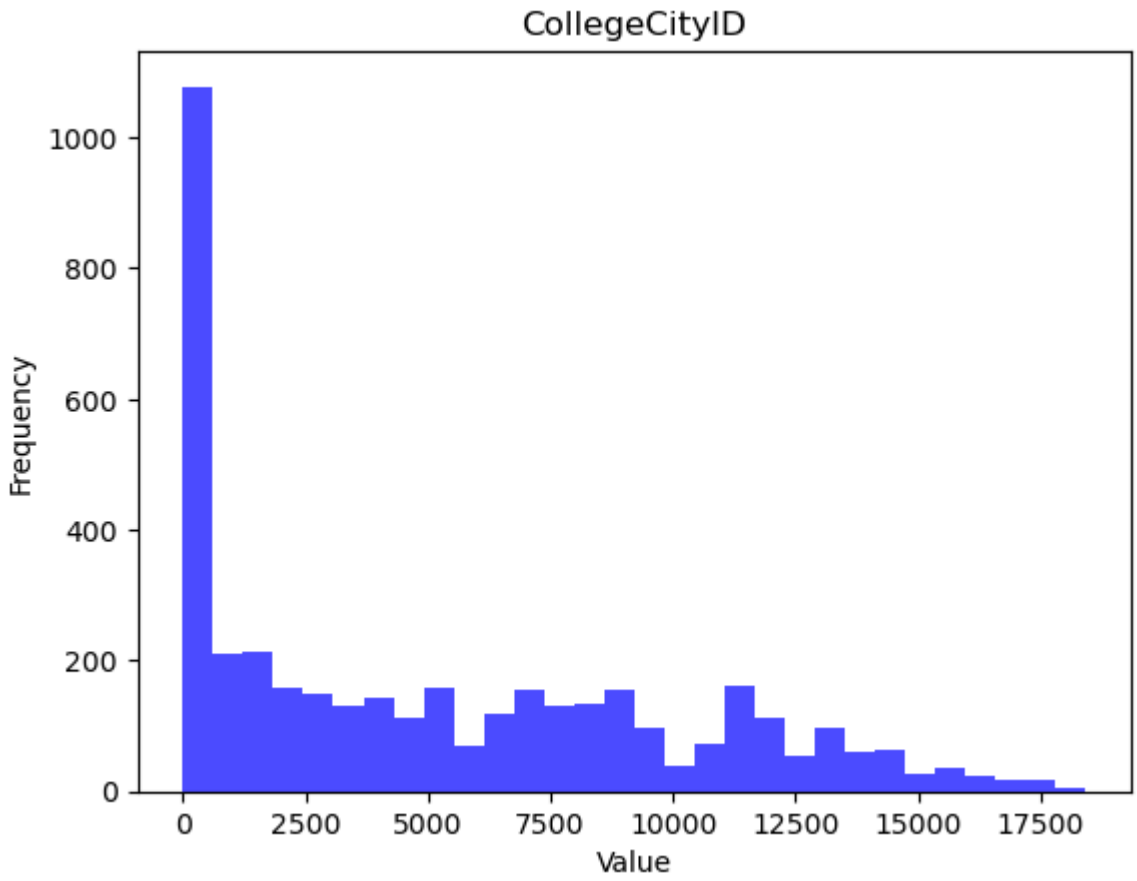

12graduation
Frequency
1000
800
600
400
200
0
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
Value

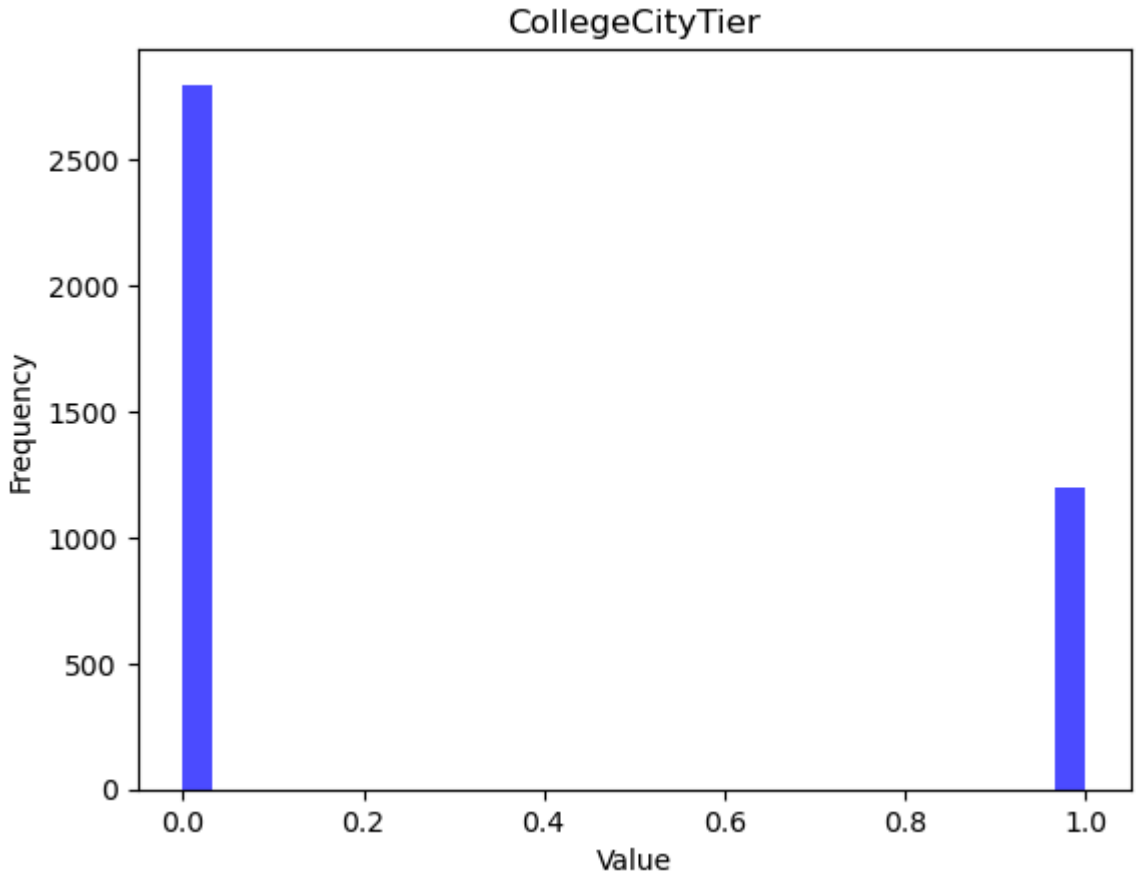

12percentage
Frequency
250
200
150
100
50
0
40
50
60
70
80
90
100
Value

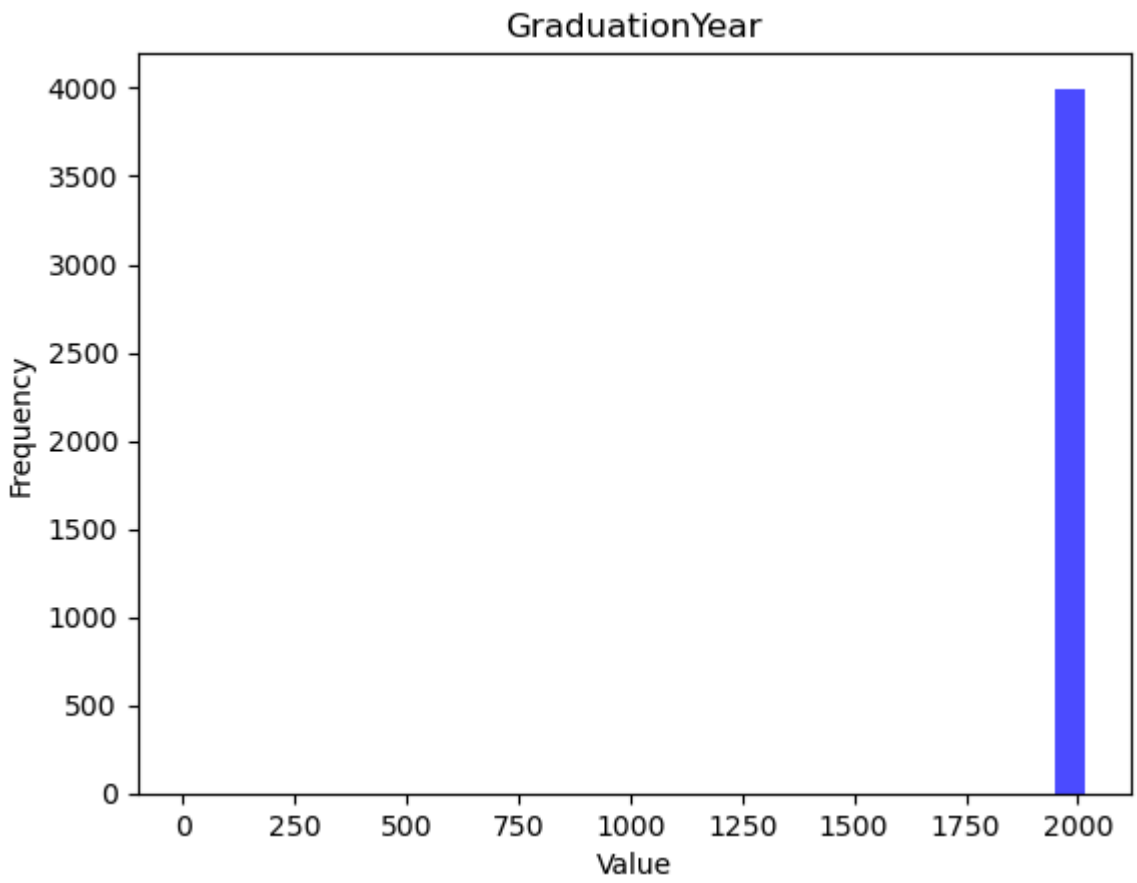

CollegeID
Frequency
1000
800
600
400
200
0
0
2500
5000
7500
10000
12500
15000
17500
Value

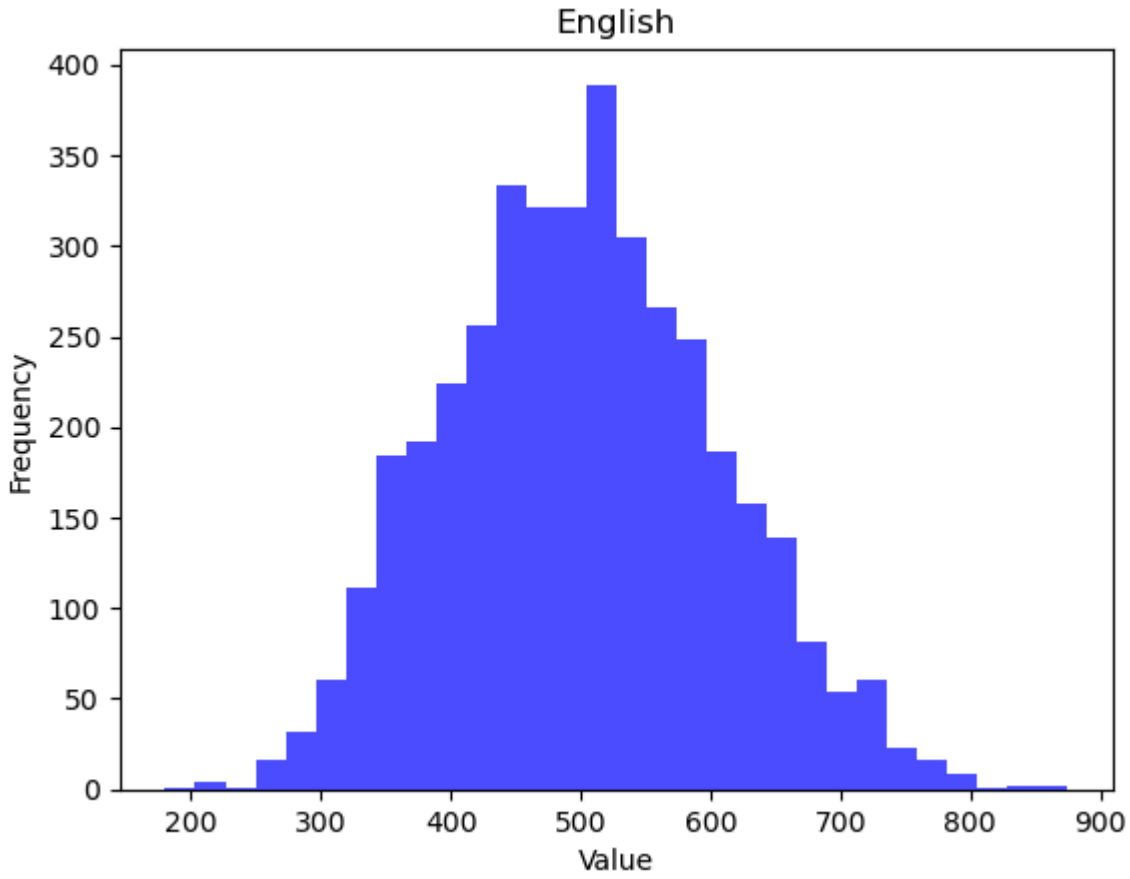

CollegeTier
Frequency
3500
3000
2500
2000
1500
1000
500
0
1.0
1.2
1.4
1.6
1.8
2.0
Value

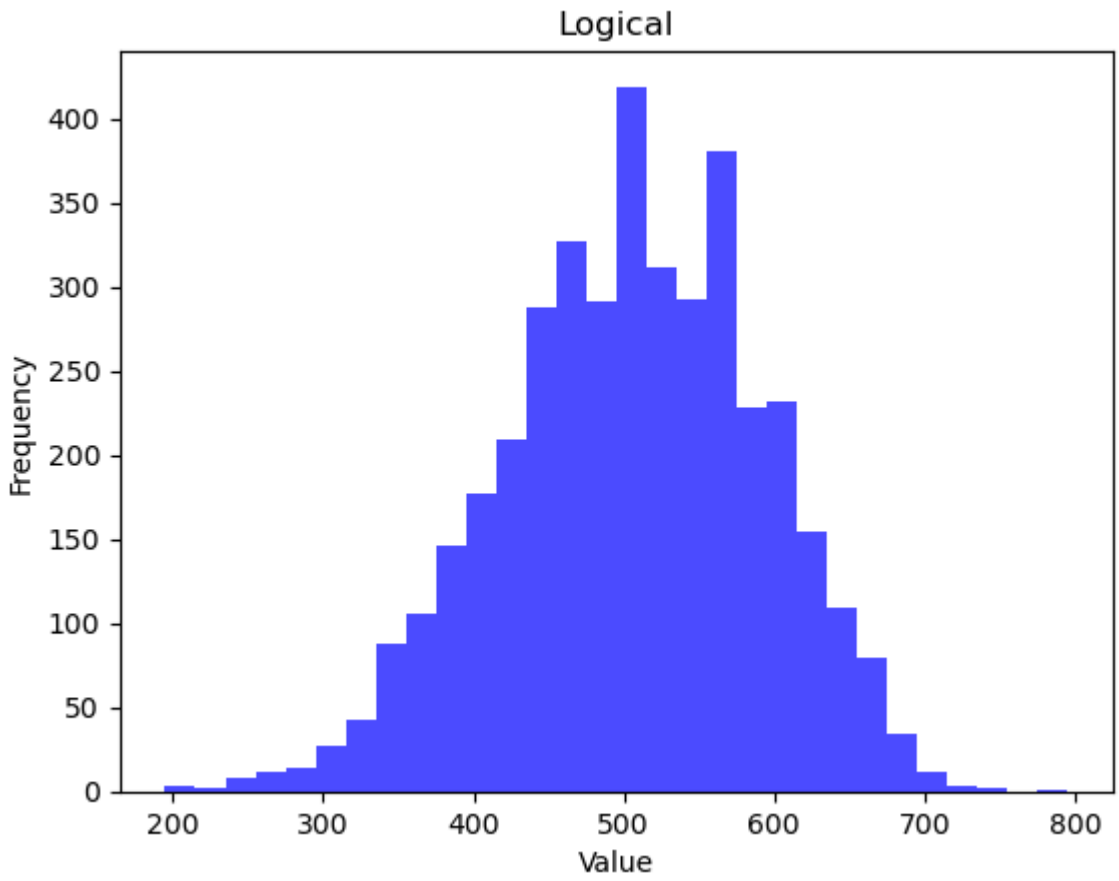

collegeGPA
Frequency
Value
0
100
200
300
400
500
600
700
20
40
60
80
100

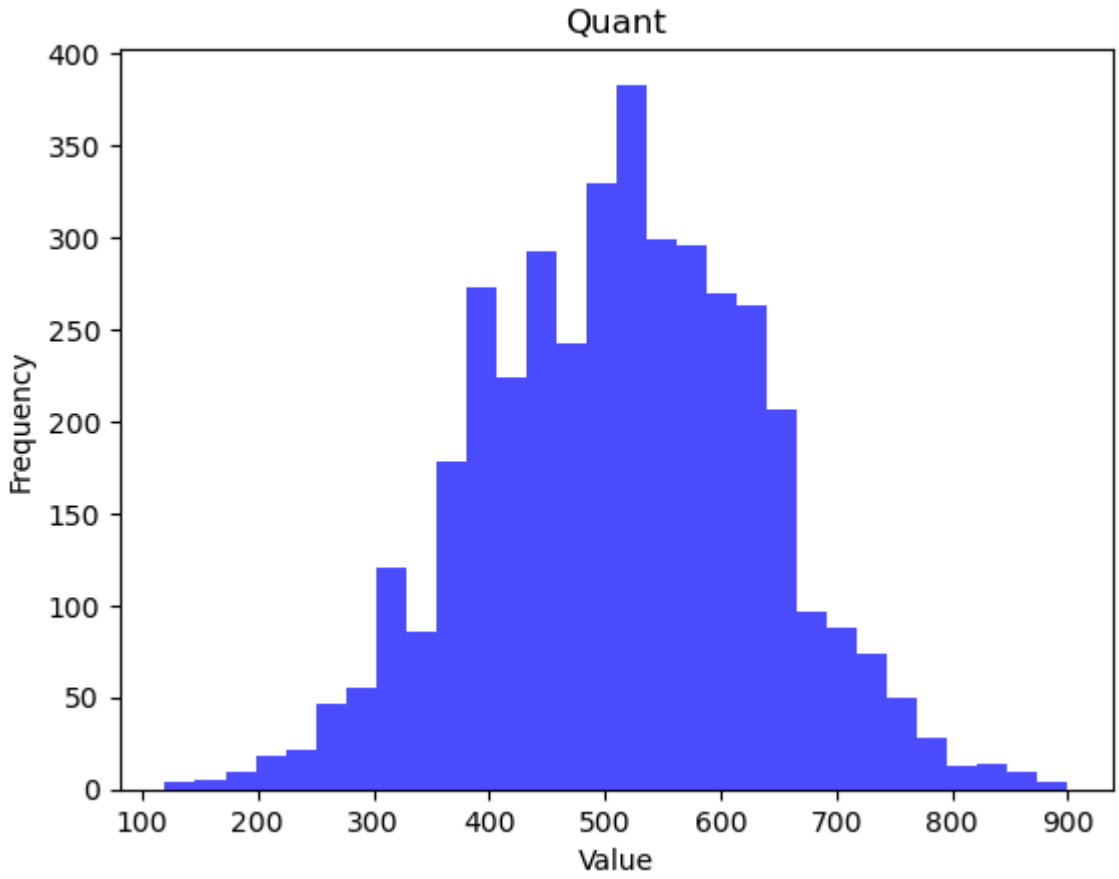

CollegeCityID
Frequency
Value
0
200
400
600
800
1000
0
2500
5000
7500
10000
12500
15000
17500

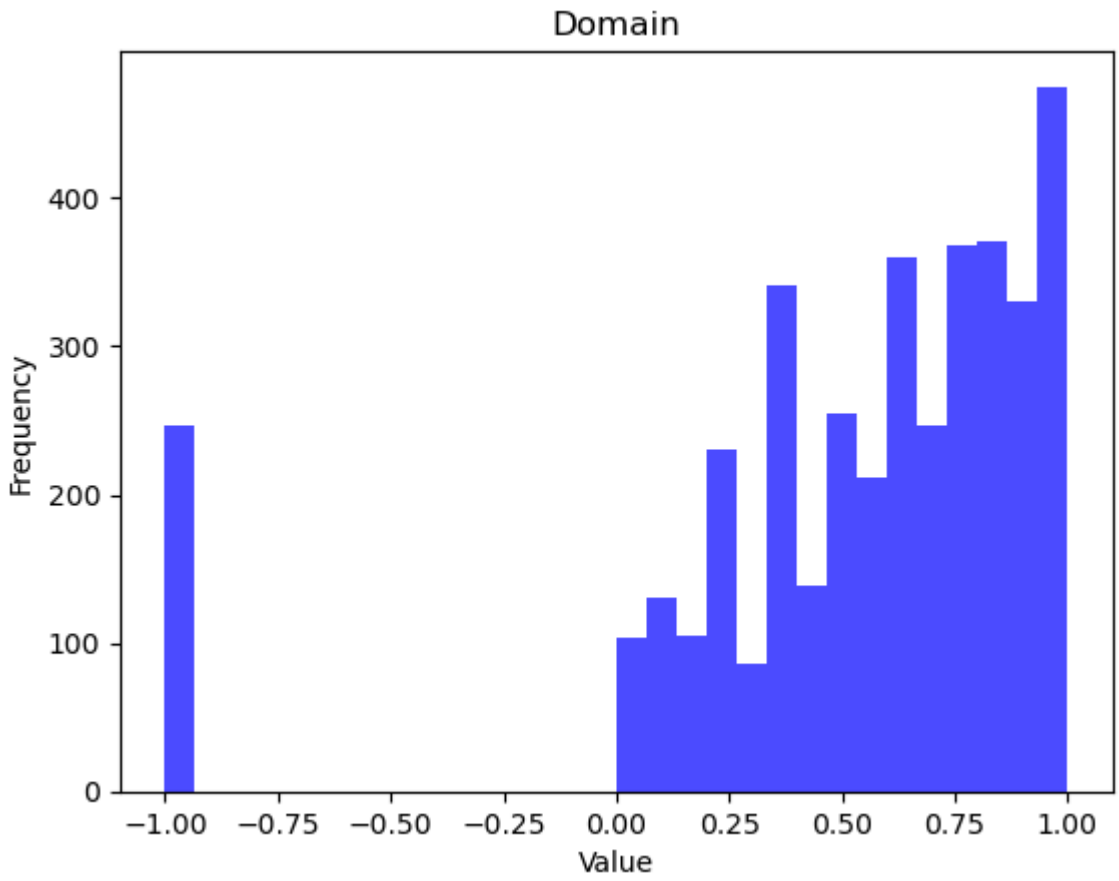

CollegeCityTier
Frequency
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
Value

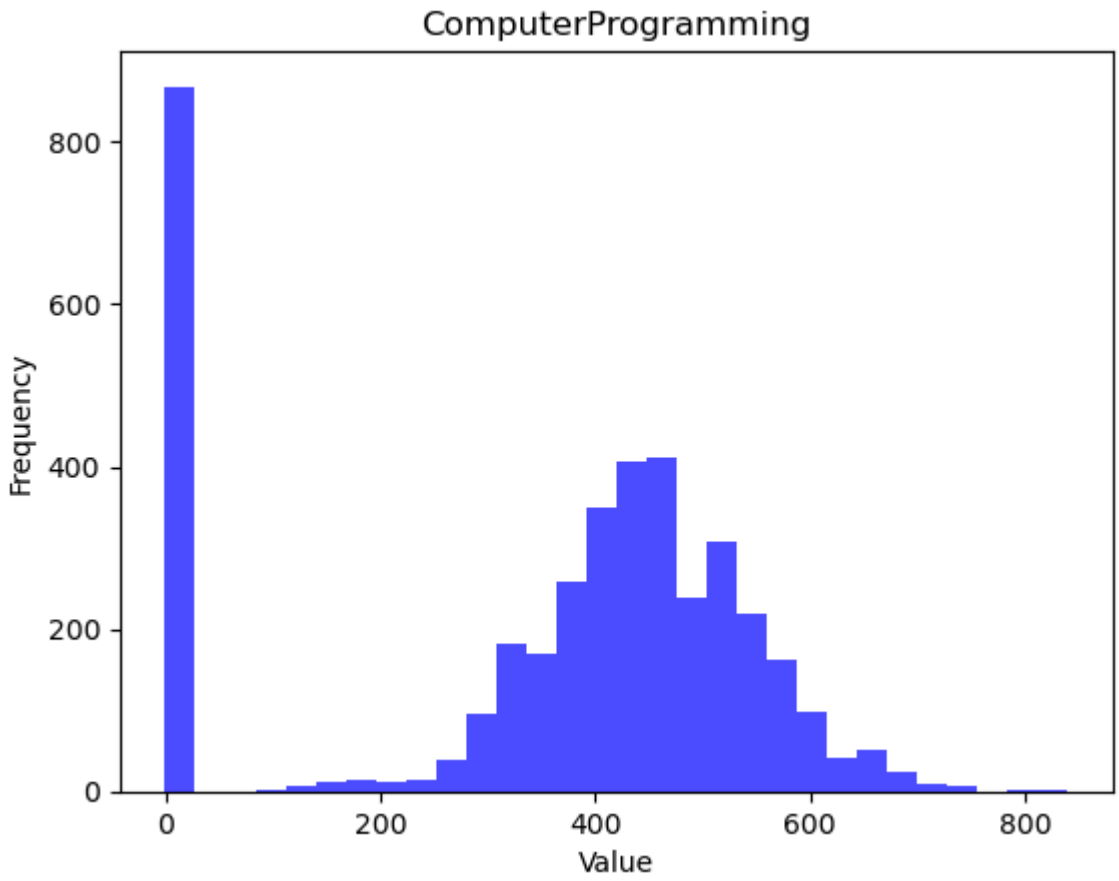

GraduationYear
Frequency
4000
3500
3000
2500
2000
1500
1000
500
0
0
250
500
750
1000
1250
1500
1750
2000
Value

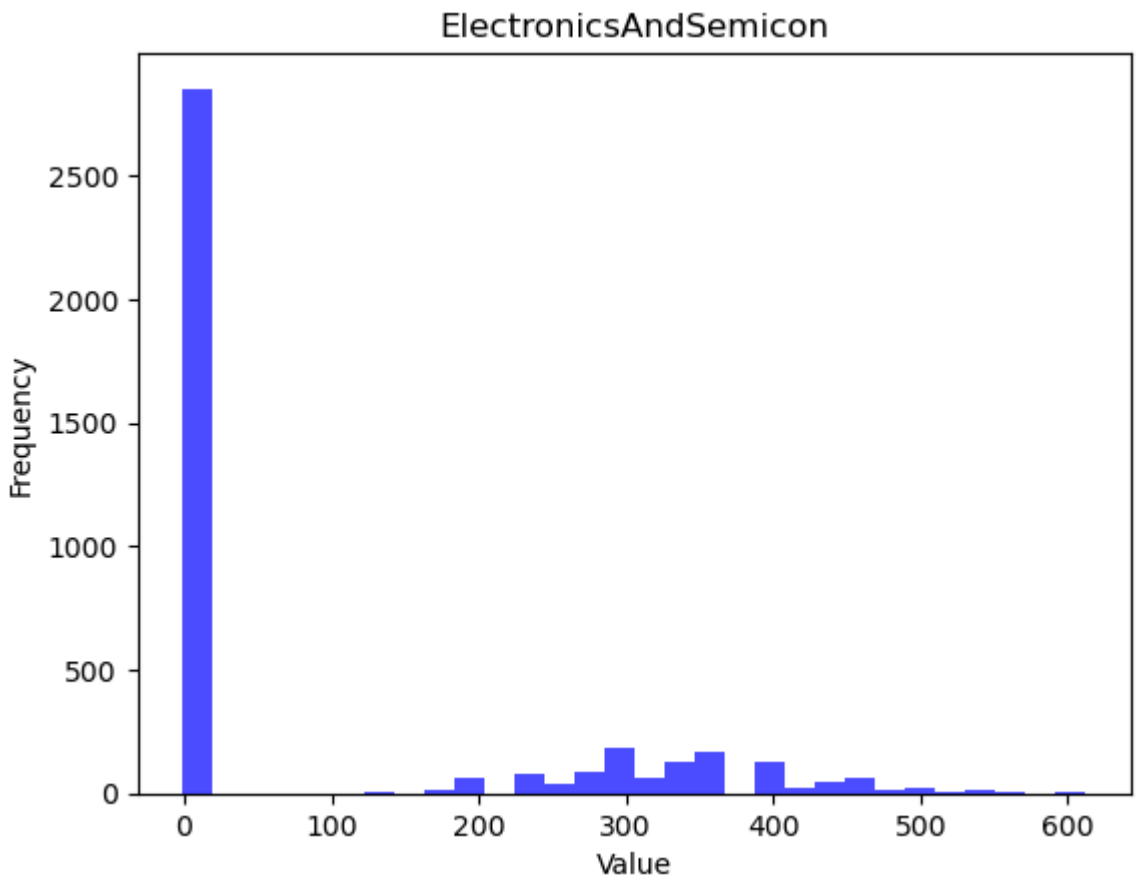

English
Frequency
Value
0
50
100
150
200
250
300
350
400
200
300
400
500
600
700
800
900

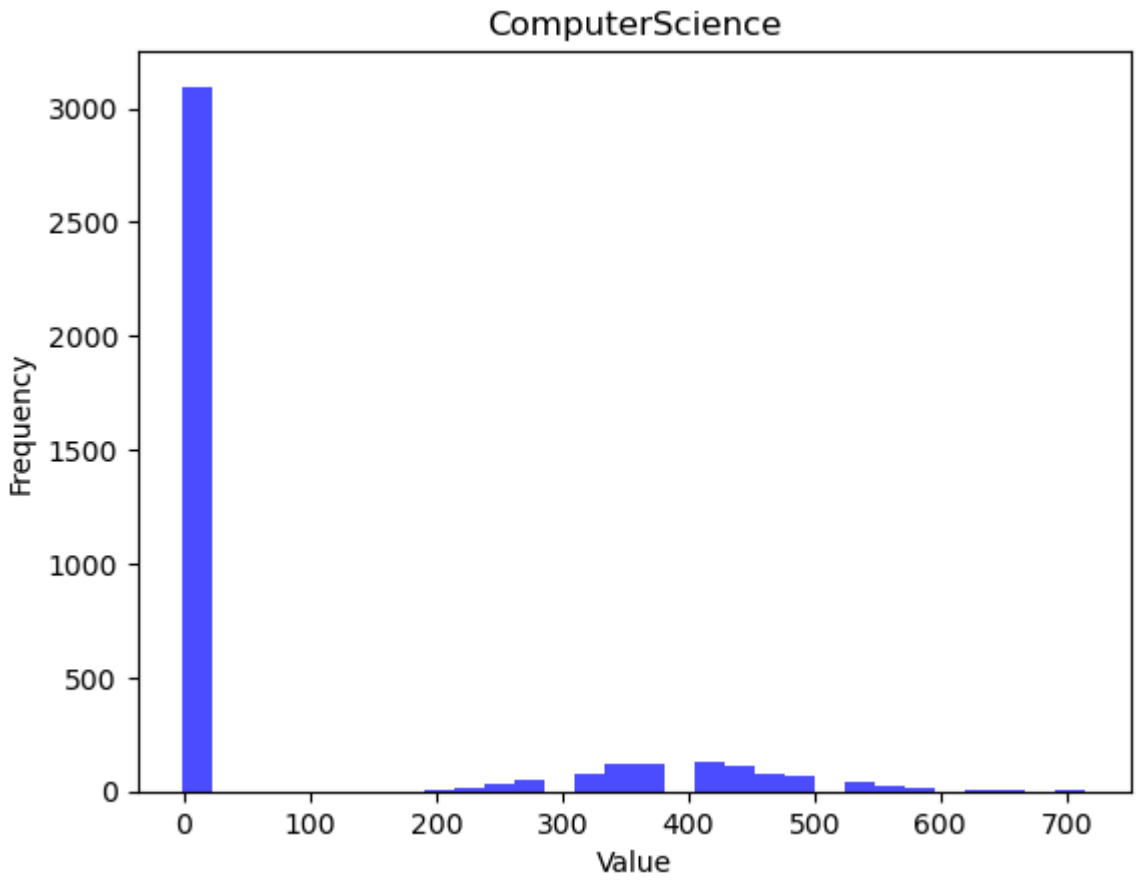

Logical
Frequency
Value
0
50
100
150
200
250
300
350
400
200
300
400
500
600
700
800

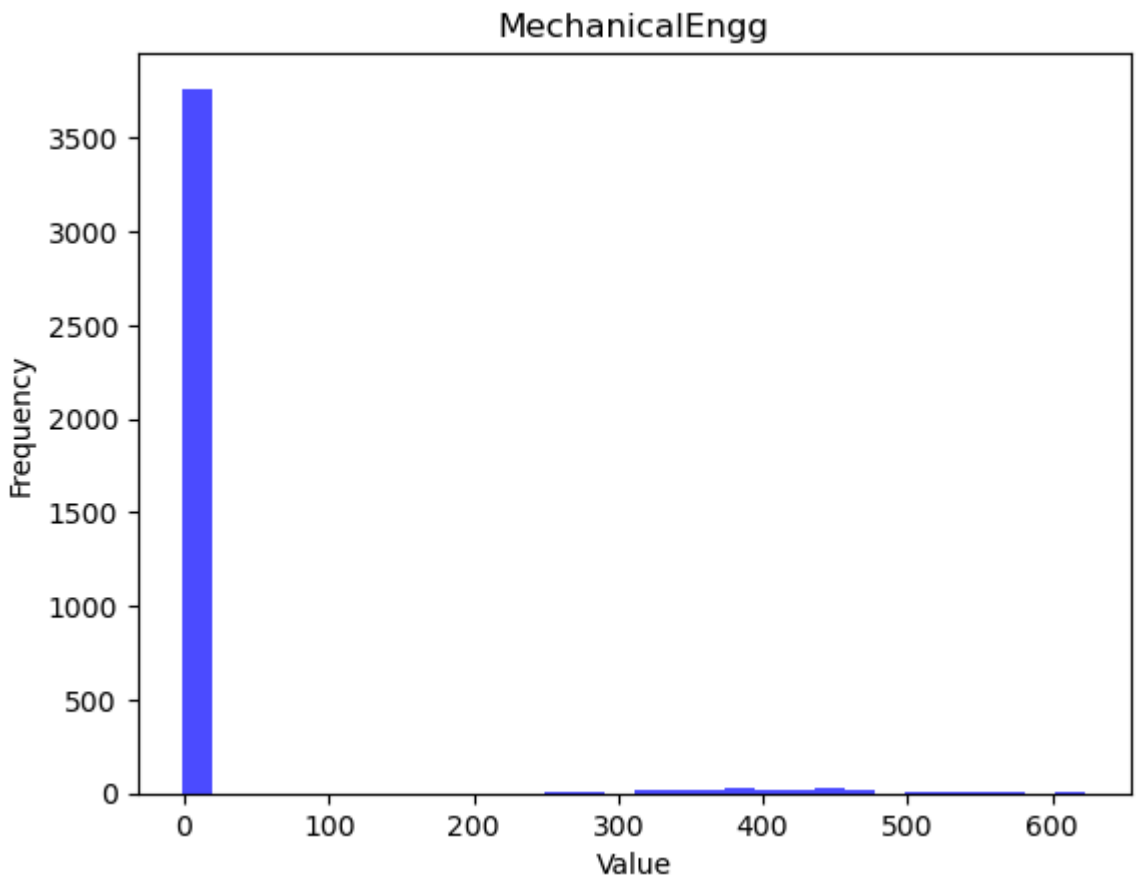

Quant
Frequency
Value
0
50
100
150
200
250
300
350
400
100
200
300
400
500
600
700
800
900

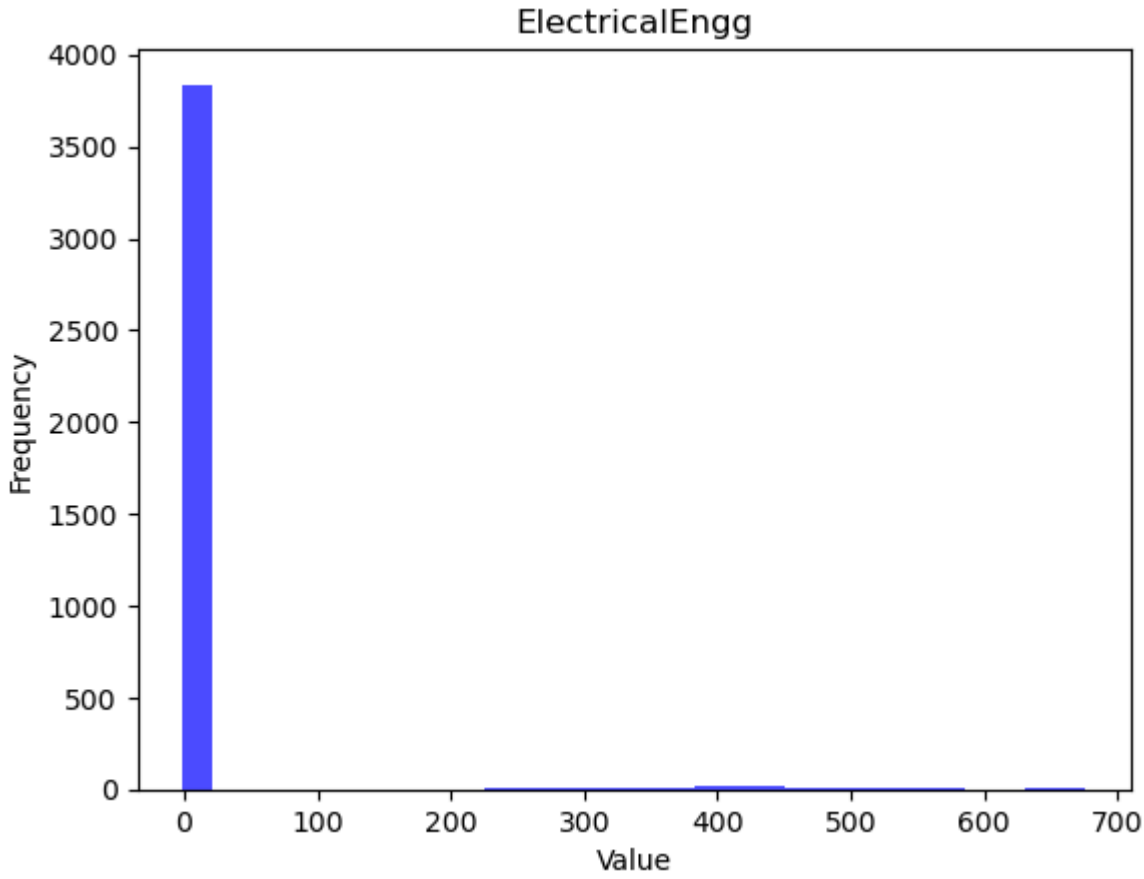

Domain
Frequency
Value
0
100
200
300
400
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

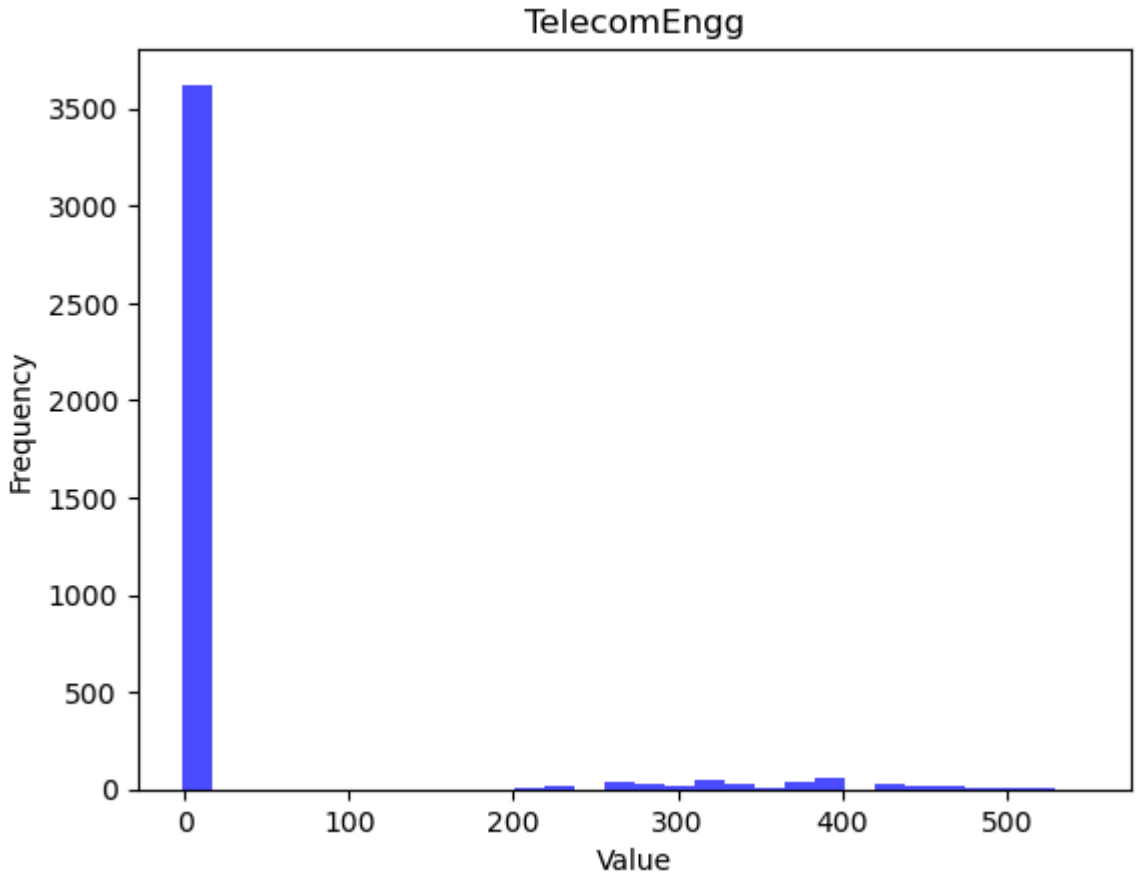

ComputerProgramming
Frequency
0
200
400
600
800
Value

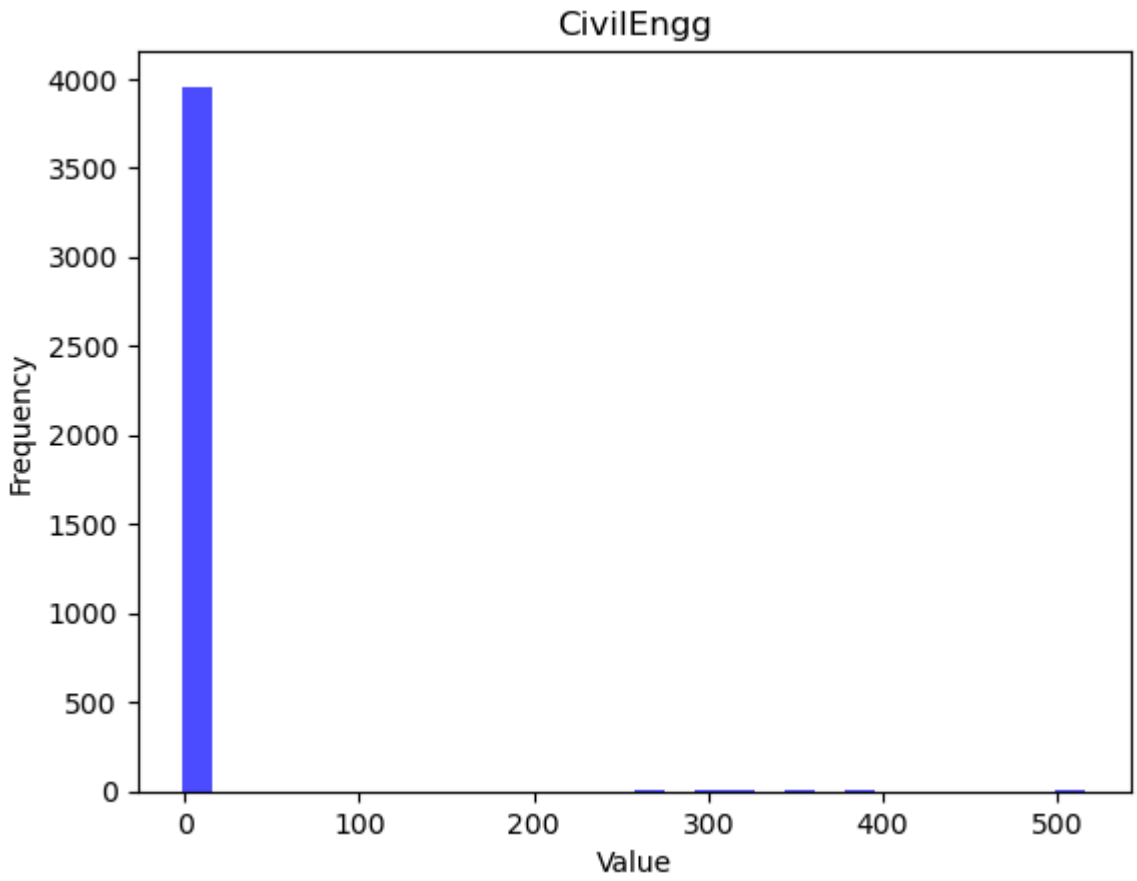

ElectronicsAndSemicon
Frequency
0
500
1000
1500
2000
2500
0
100
200
300
400
500
600
Value

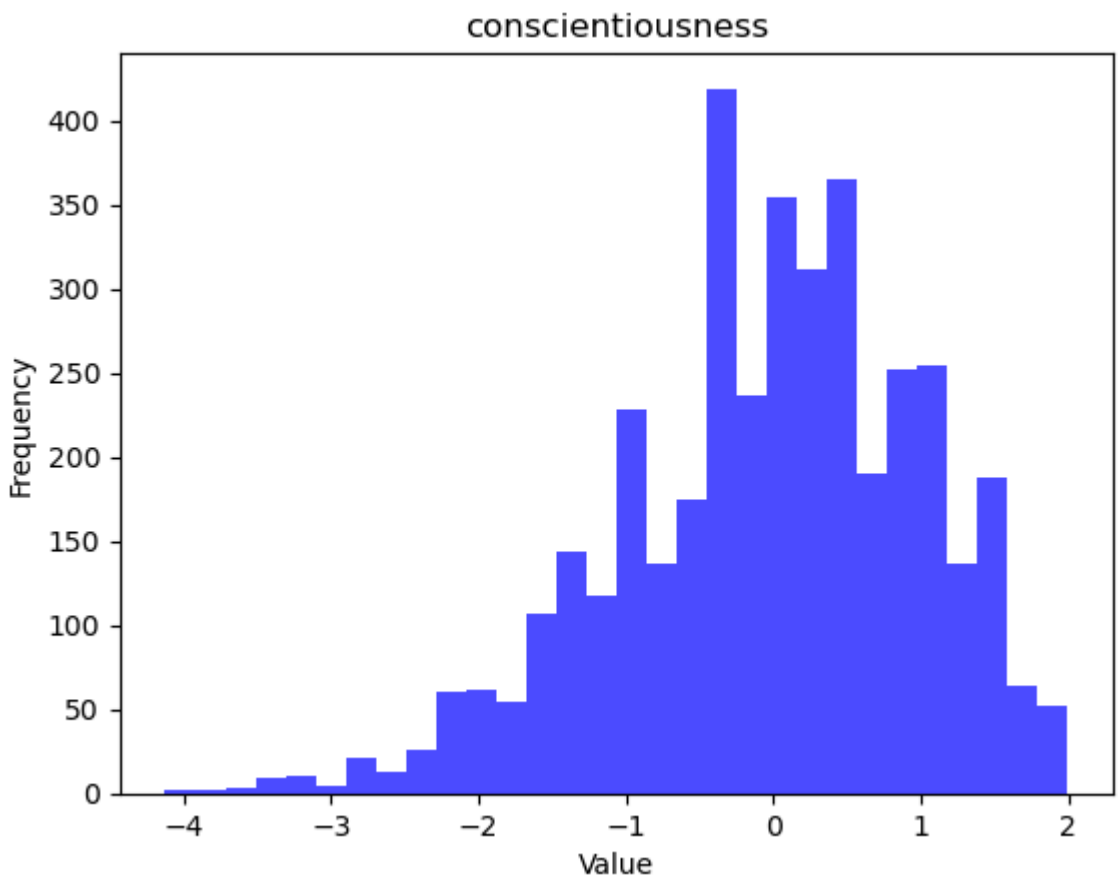

ComputerScience
Frequency
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500
600
700
Value

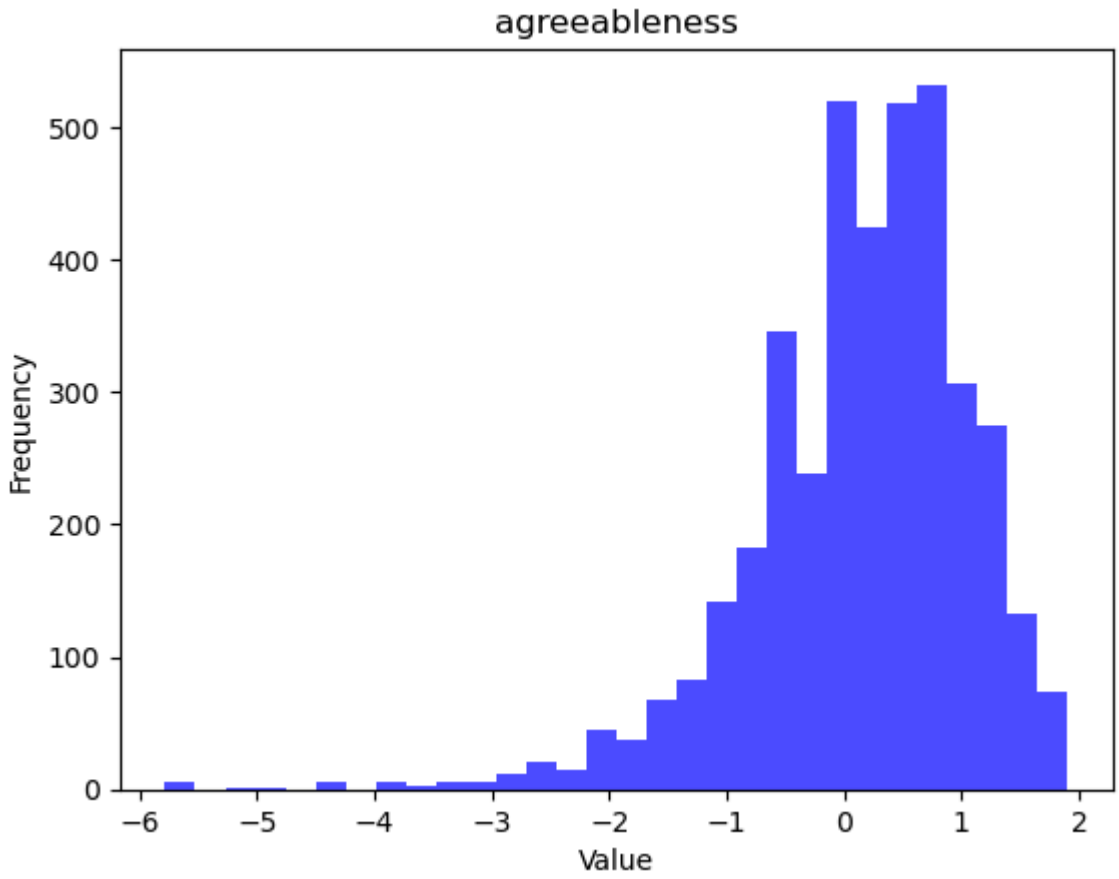

MechanicalEngg
Frequency
3500
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500
600
Value

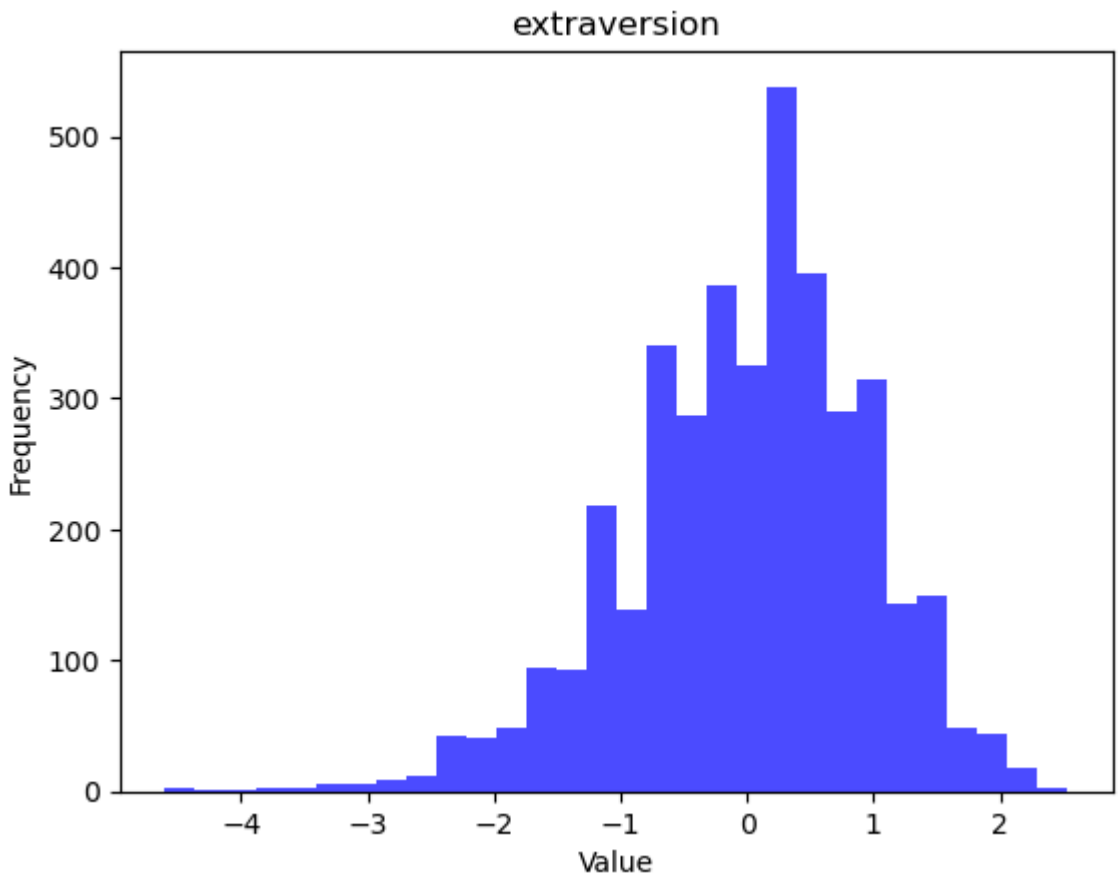

ElectricalEngg
Frequency
Value
0
500
1000
1500
2000
2500
3000
3500
4000
0
100
200
300
400
500
600
700

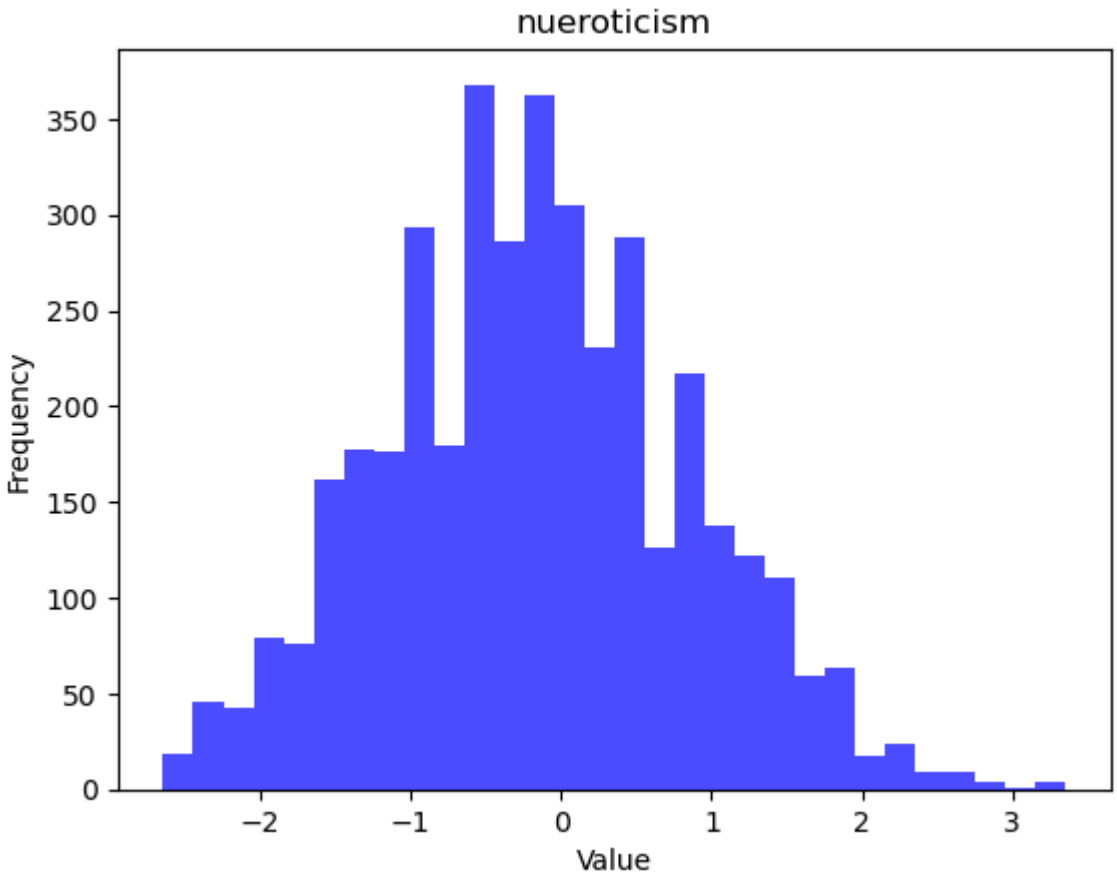

TelecomEngg
Frequency
Value
0
500
1000
1500
2000
2500
3000
3500
0
100
200
300
400
500

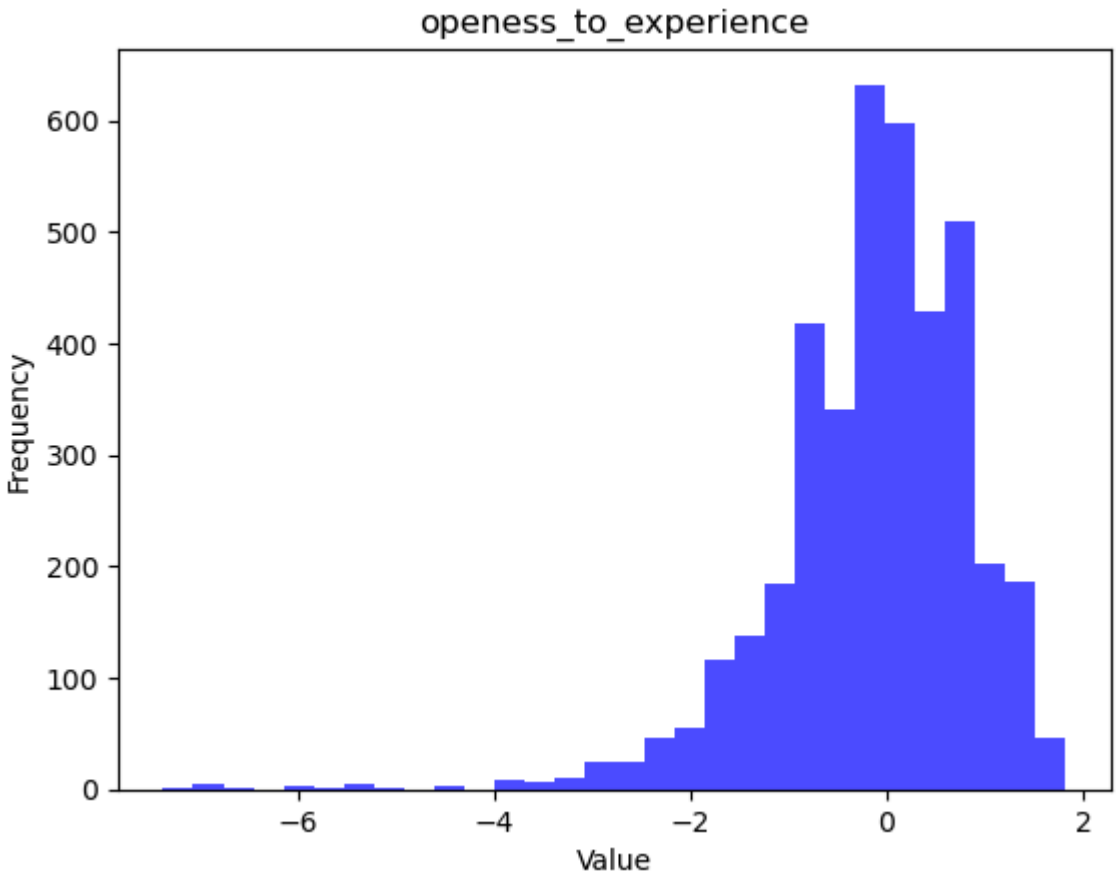

CivilEngg
Frequency
Value
0
500
1000
1500
2000
2500
3000
3500
4000
0
100
200
300
400
500


conscientiousness
Frequency
Value
0
50
100
150
200
250
300
350
400
−4
−3
−2
−1
0
1
2


agreeableness
Frequency
Value
0
100
200
300
400
500
−6
−5
−4
−3
−2
−1
0
1
2


extraversion
Frequency
Value
0
100
200
300
400
500
−4
−3
−2
−1
0
1
2

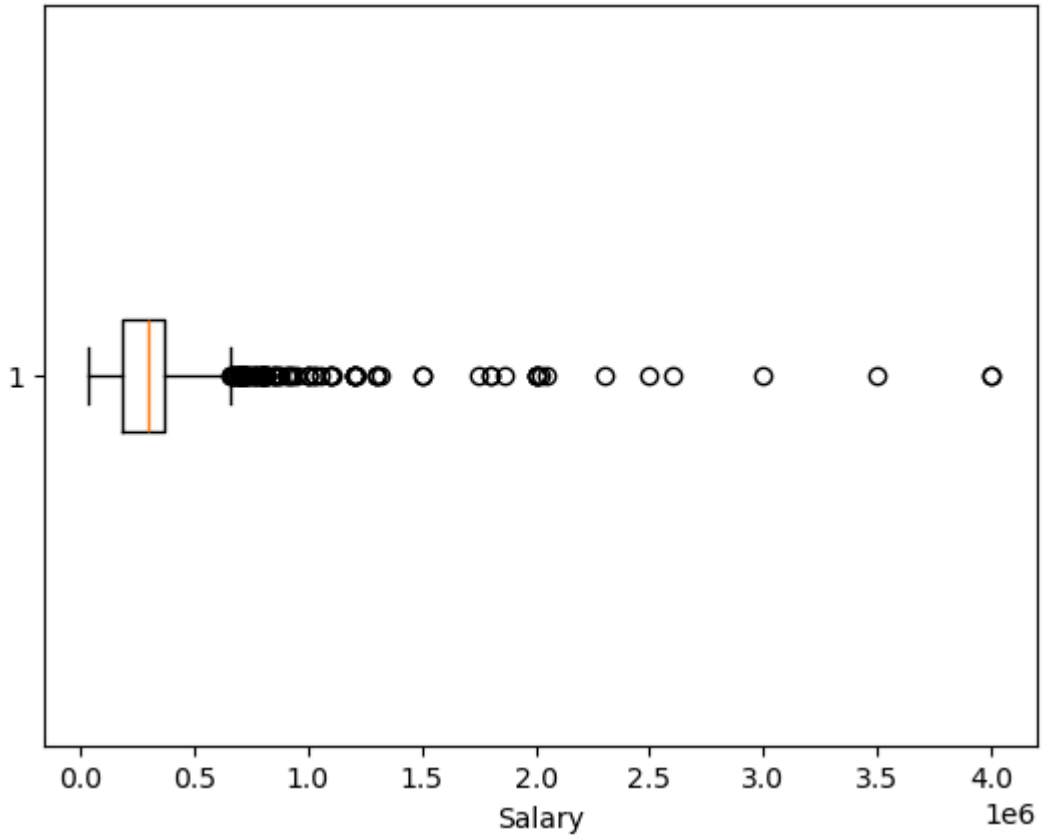


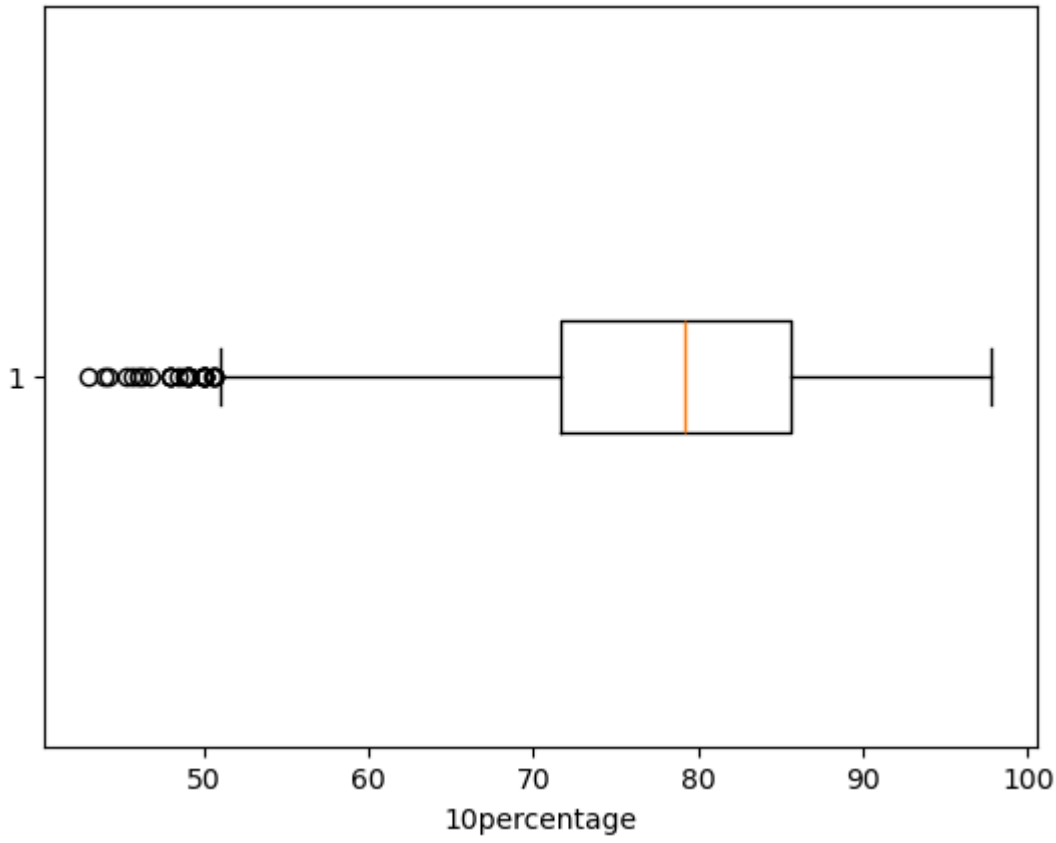


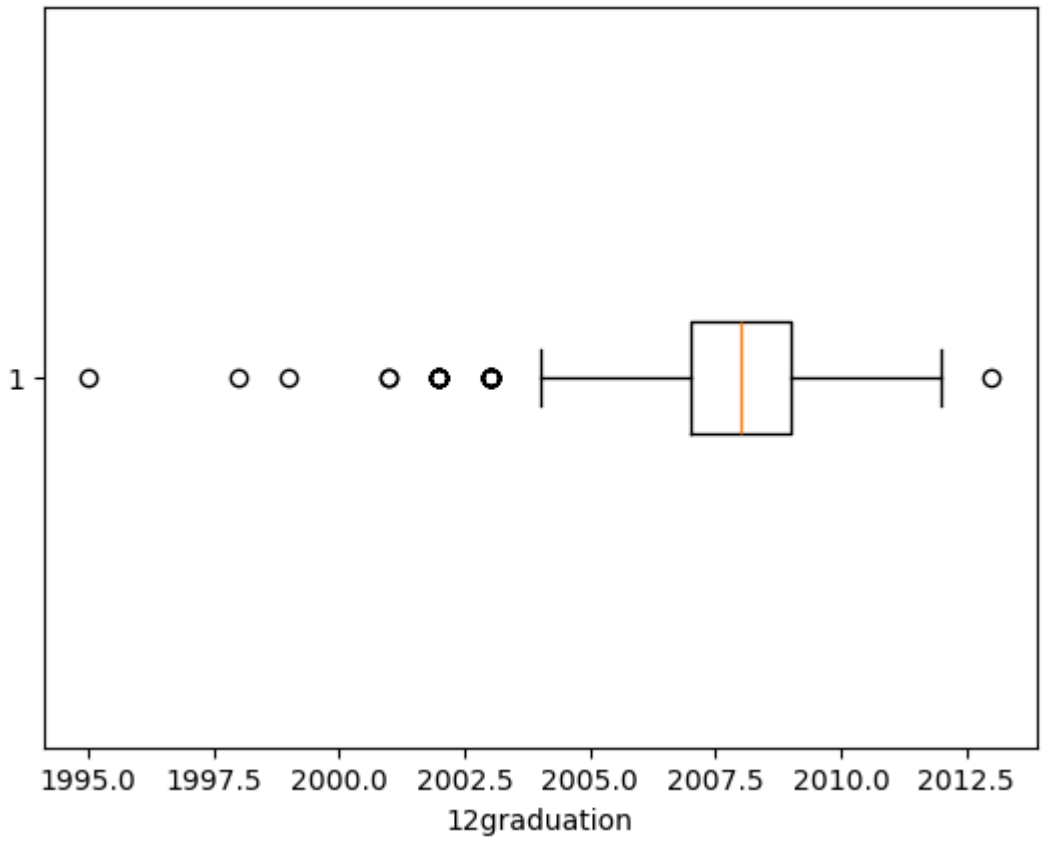
Boxplot

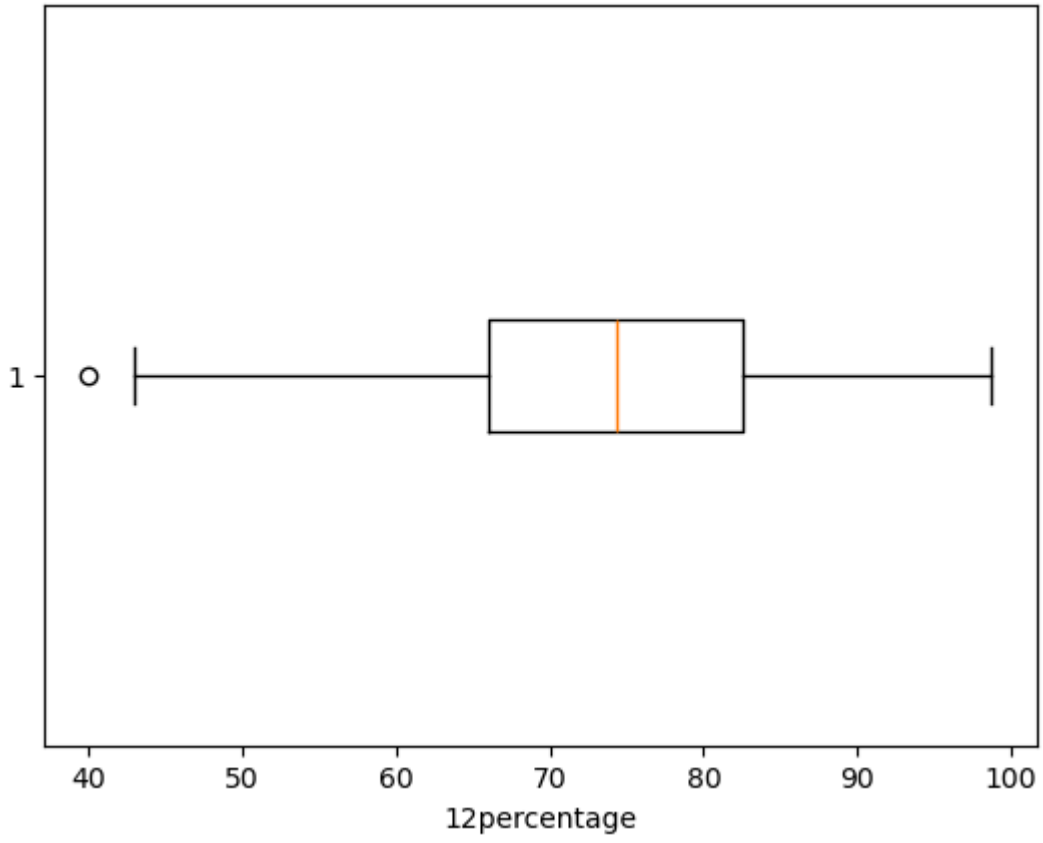
Outliers are present in each numerical column

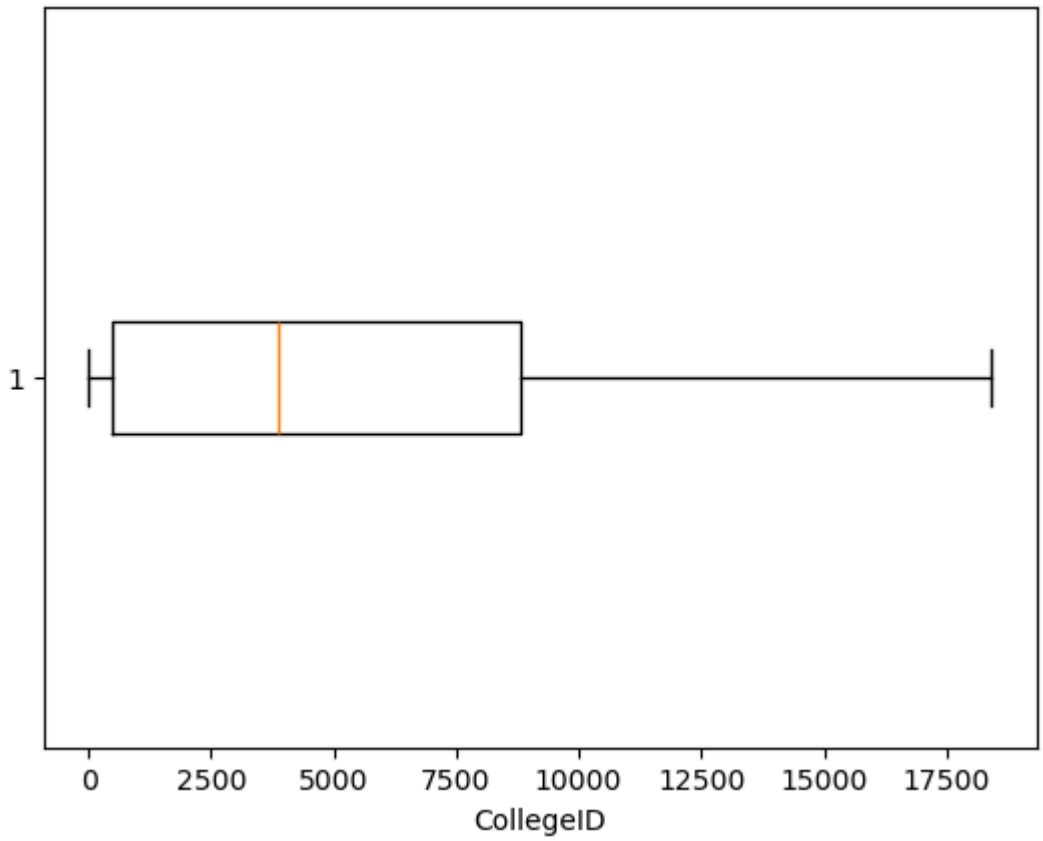
In [23]:
```
for i in num_cols[1:]:
    year_data=numerical_cols[i]
    plt.boxplot(year_data,vert=False)
    plt.xlabel(i)
    plt.show()
```

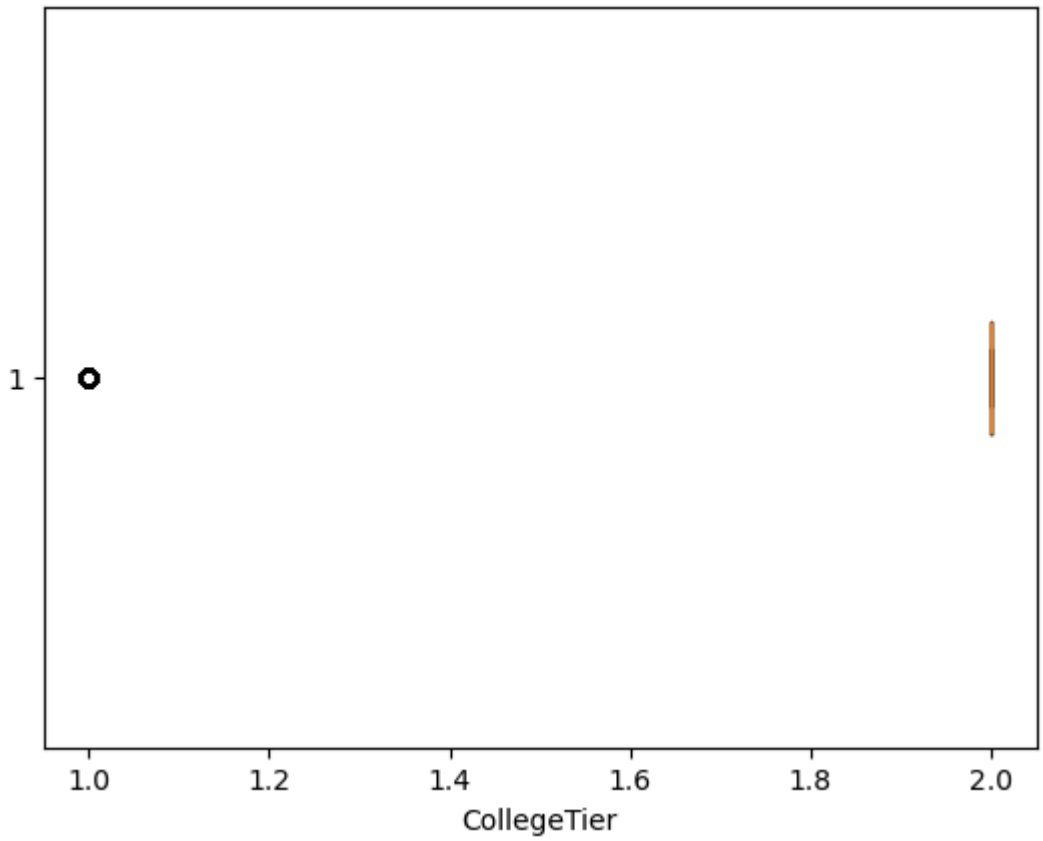


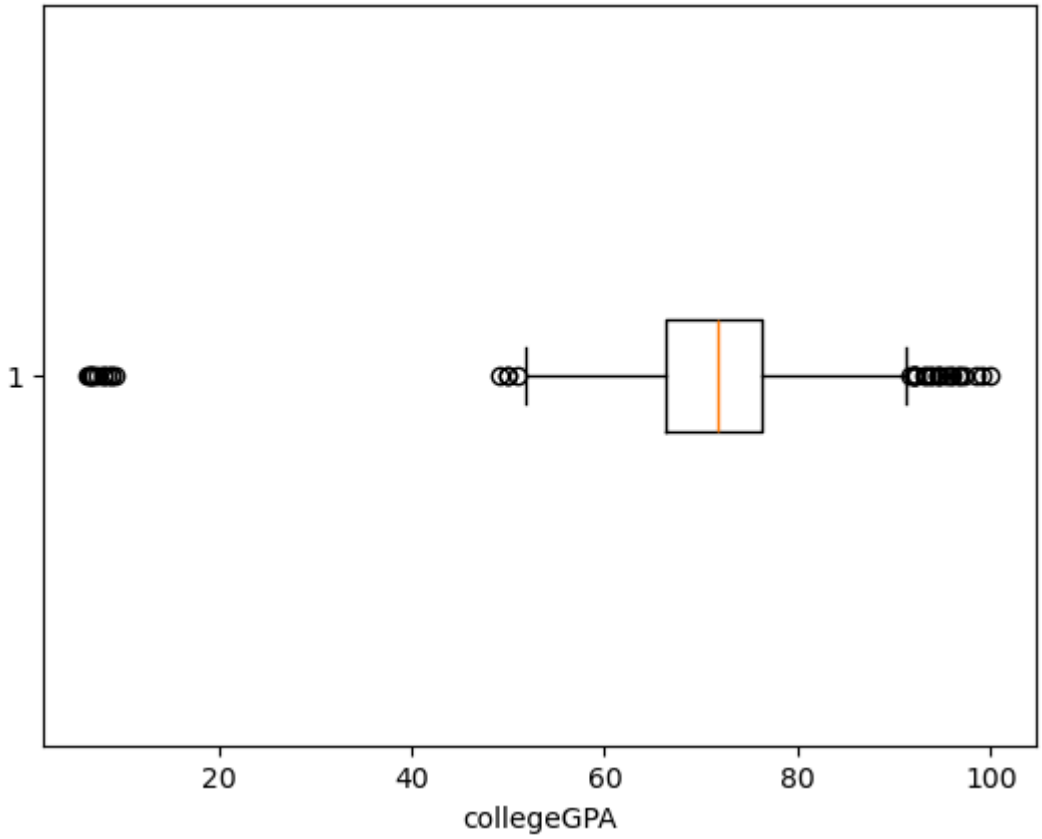


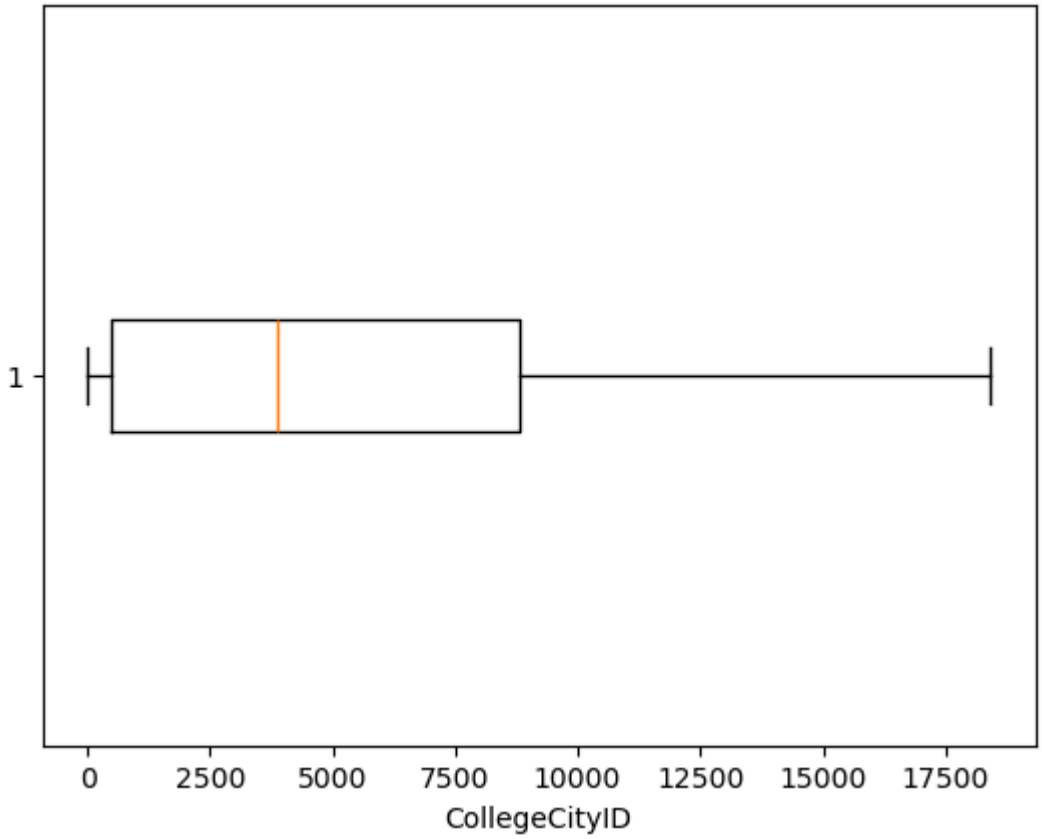

1
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
12graduation

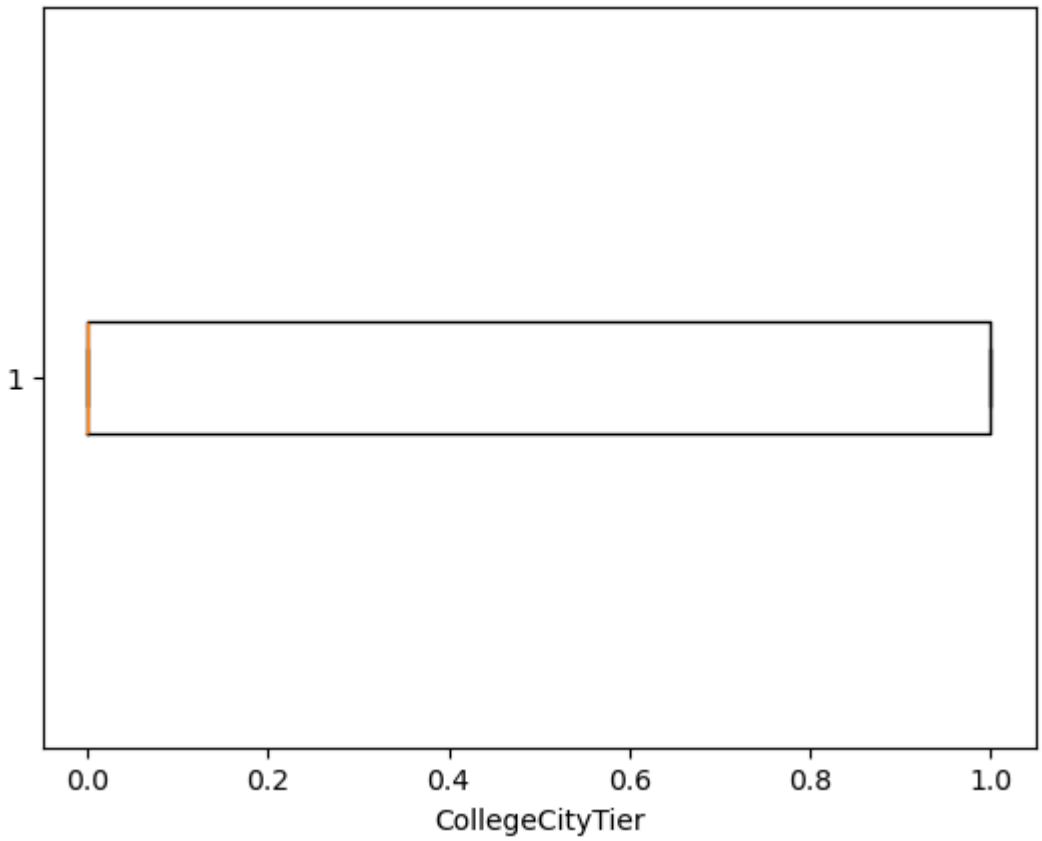

1
40
50
60
70
80
90
100
12percentage

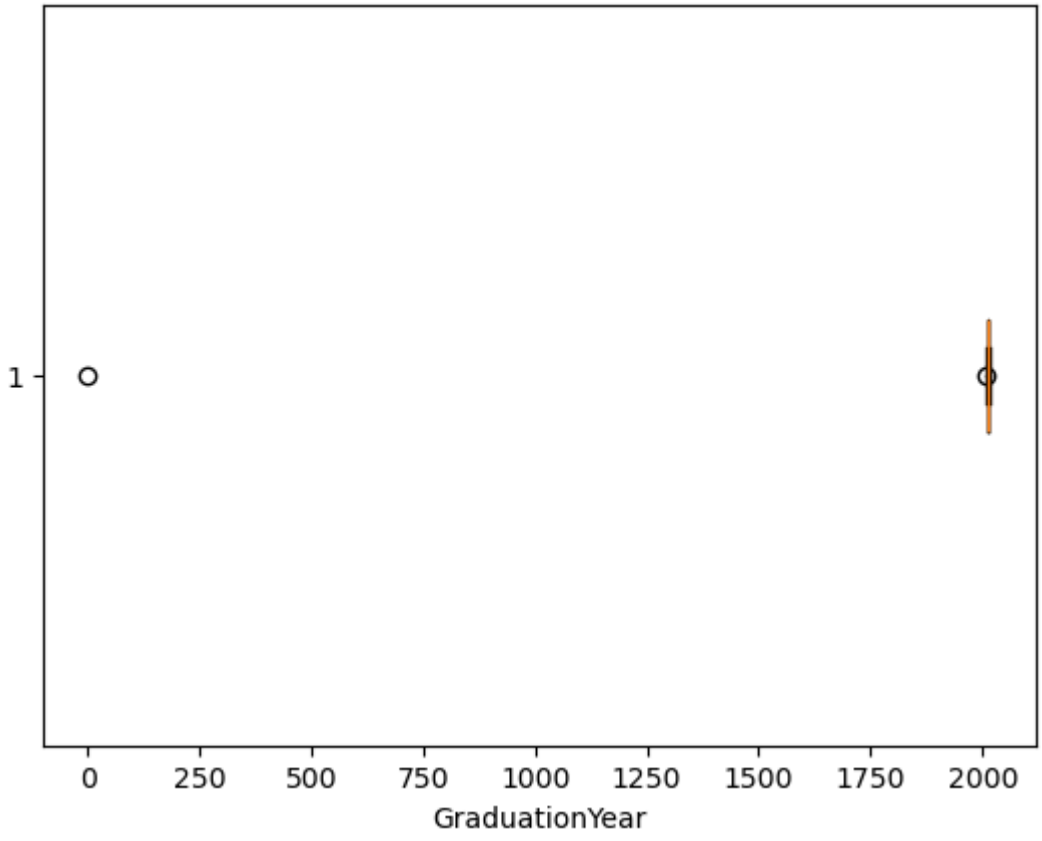

1
0
2500
5000
7500
10000
12500
15000
17500
CollegeID

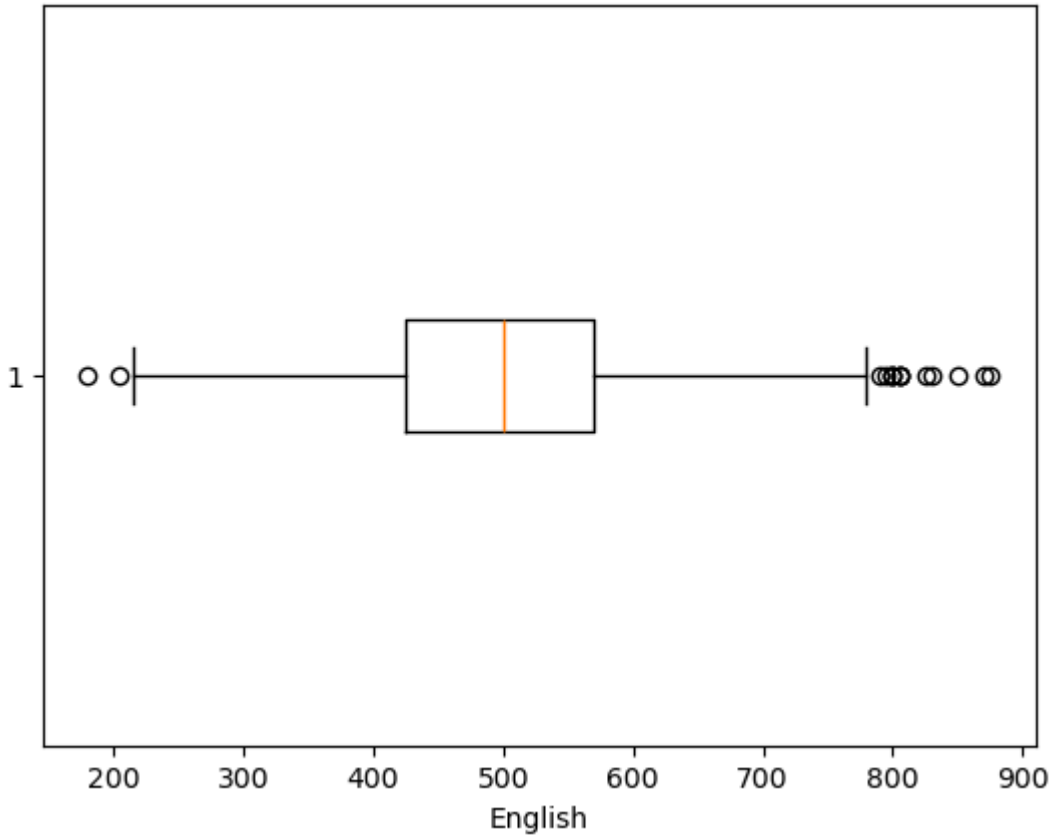

1
1.0
1.2
1.4
1.6
1.8
2.0
CollegeTier

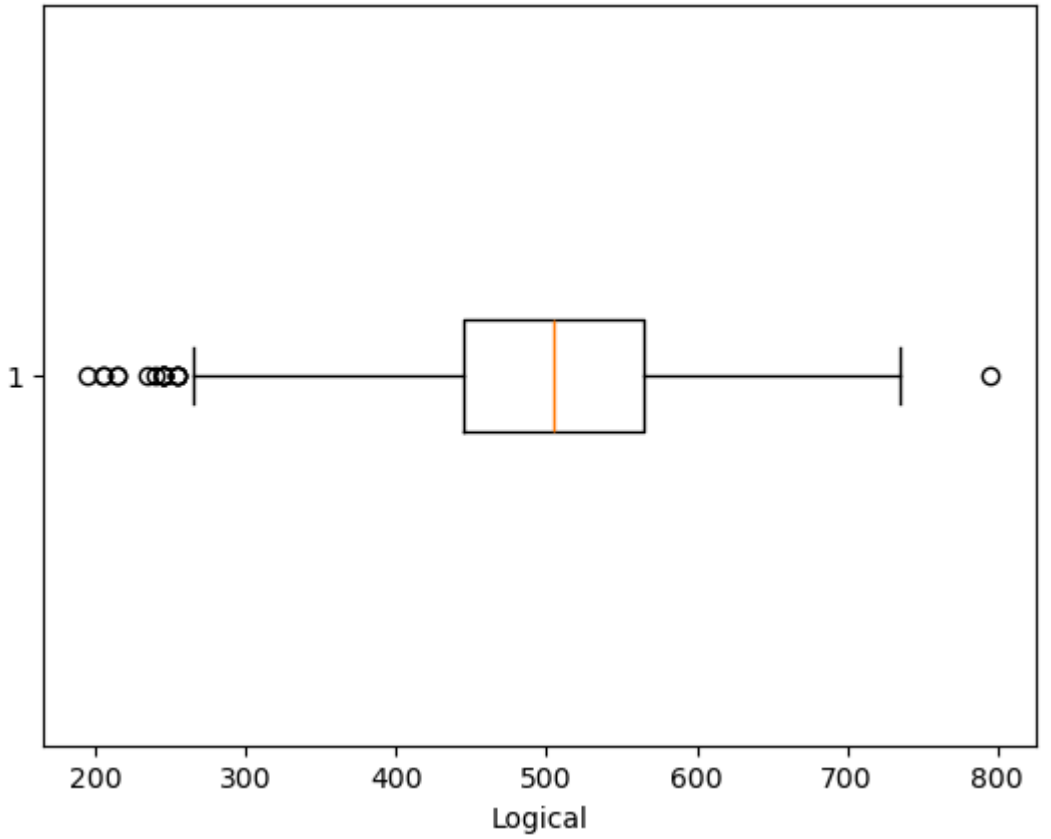

1
20
40
60
80
100
collegeGPA

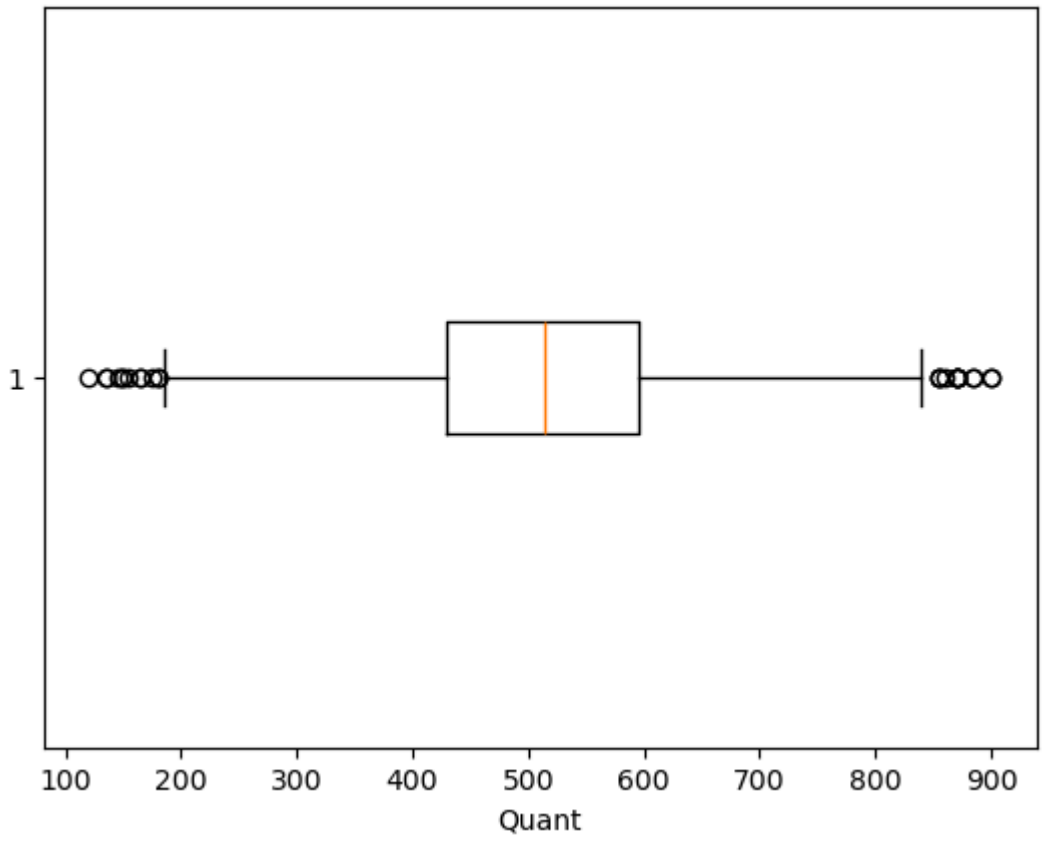

1
0
2500
5000
7500
10000
12500
15000
17500
CollegeCityID

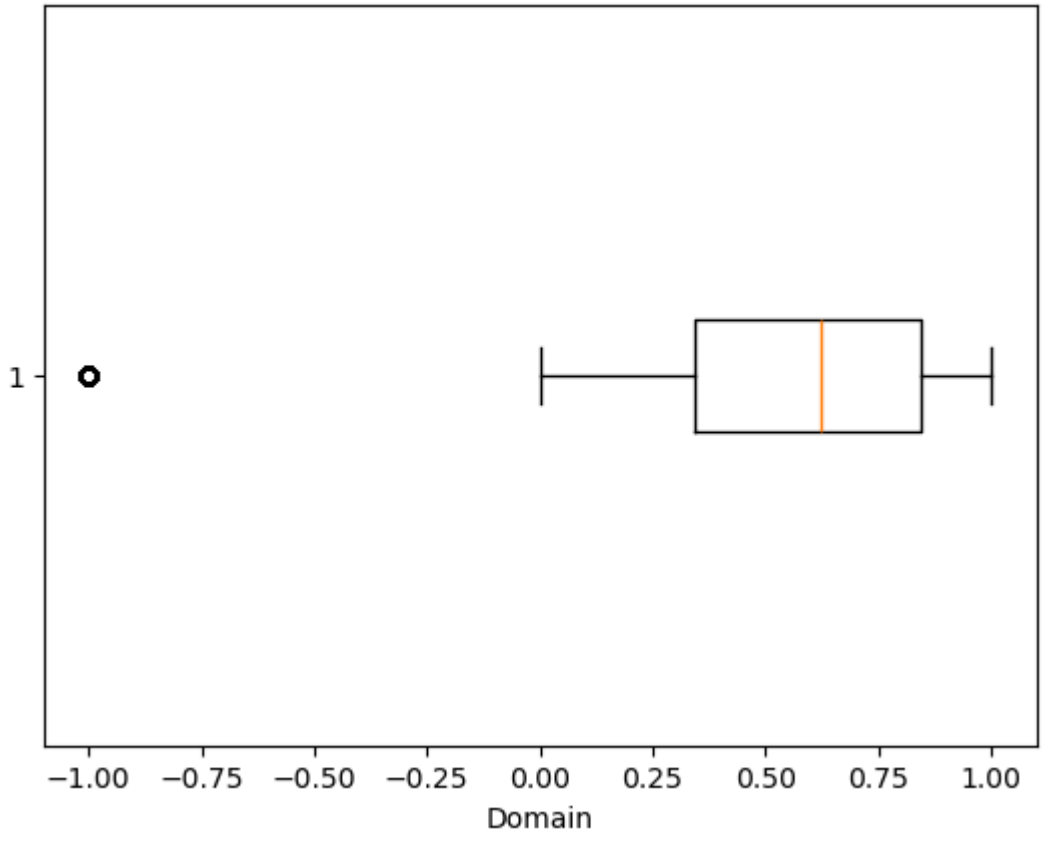

1
0.0
0.2
0.4
0.6
0.8
1.0
CollegeCityTier

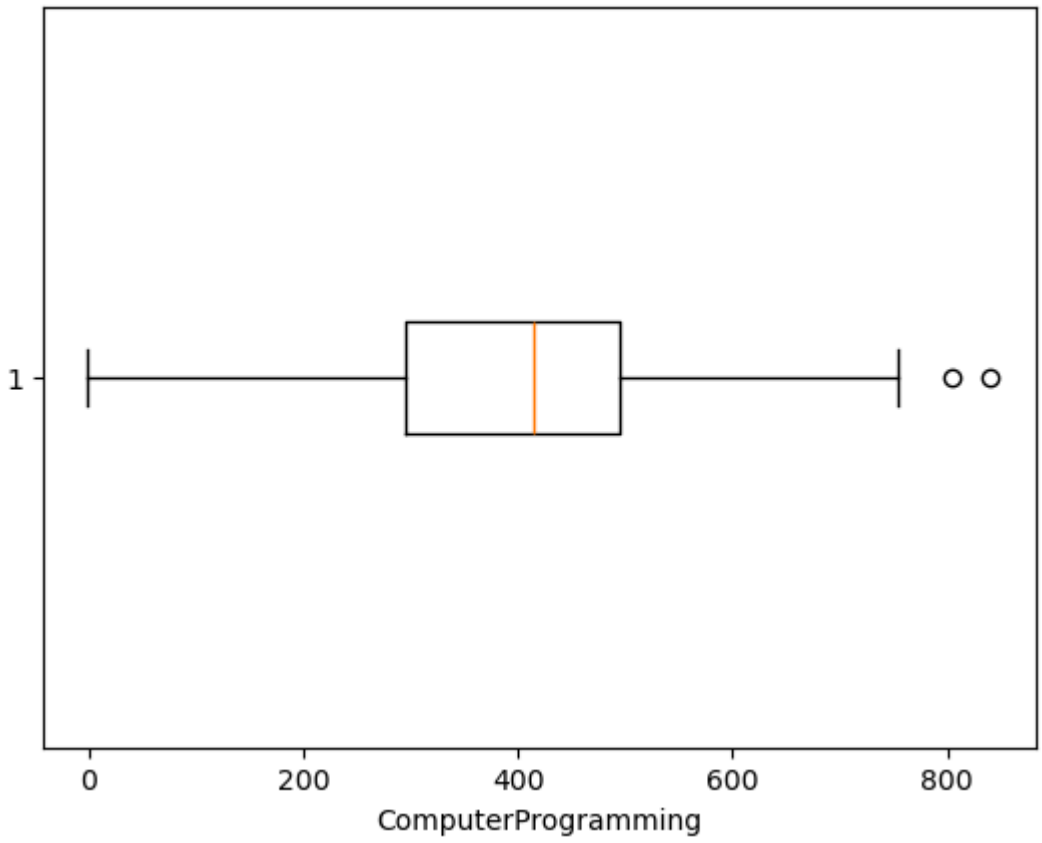

1
0
250
500
750
1000
1250
1500
1750
2000
GraduationYear

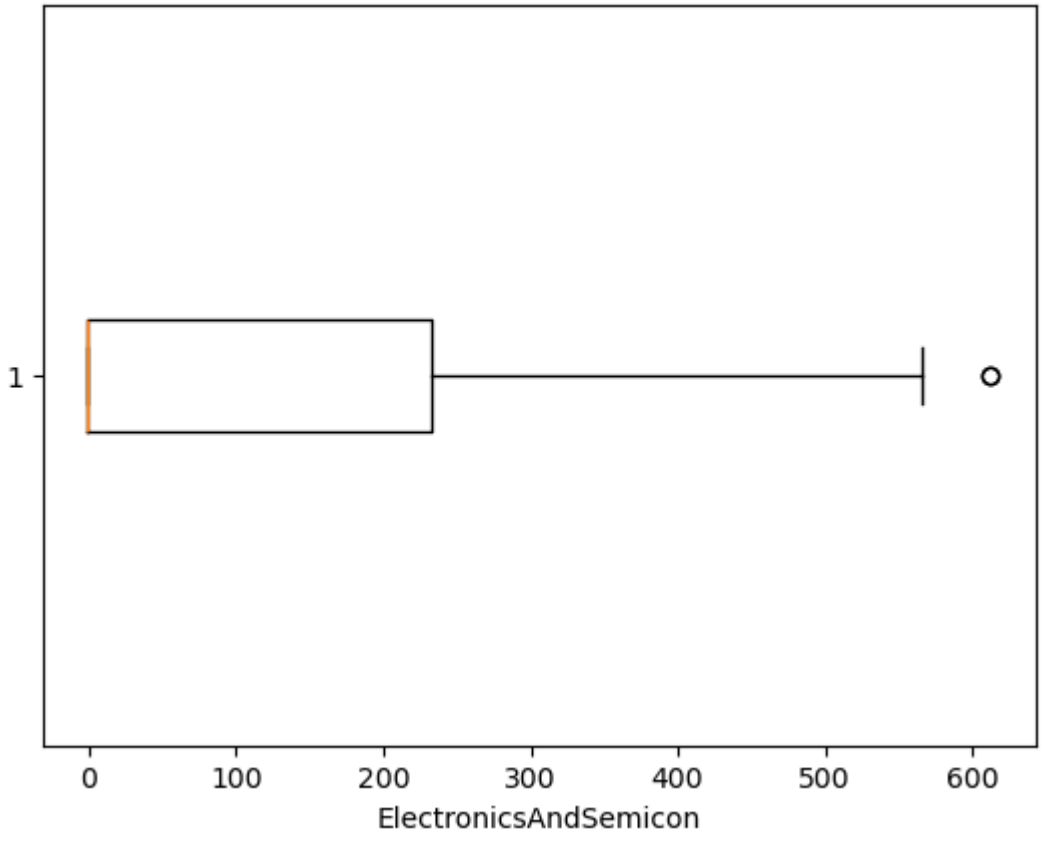

1
200
300
400
500
600
700
800
900
English

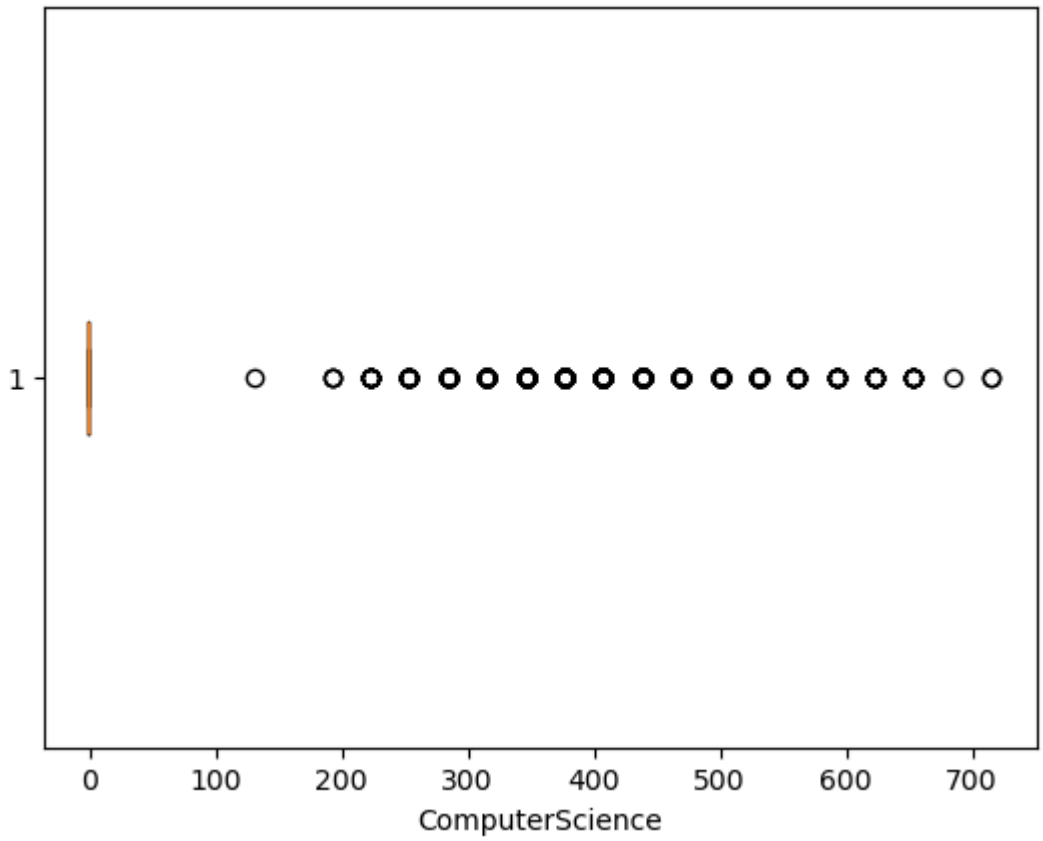

1
200
300
400
500
600
700
800
Logical

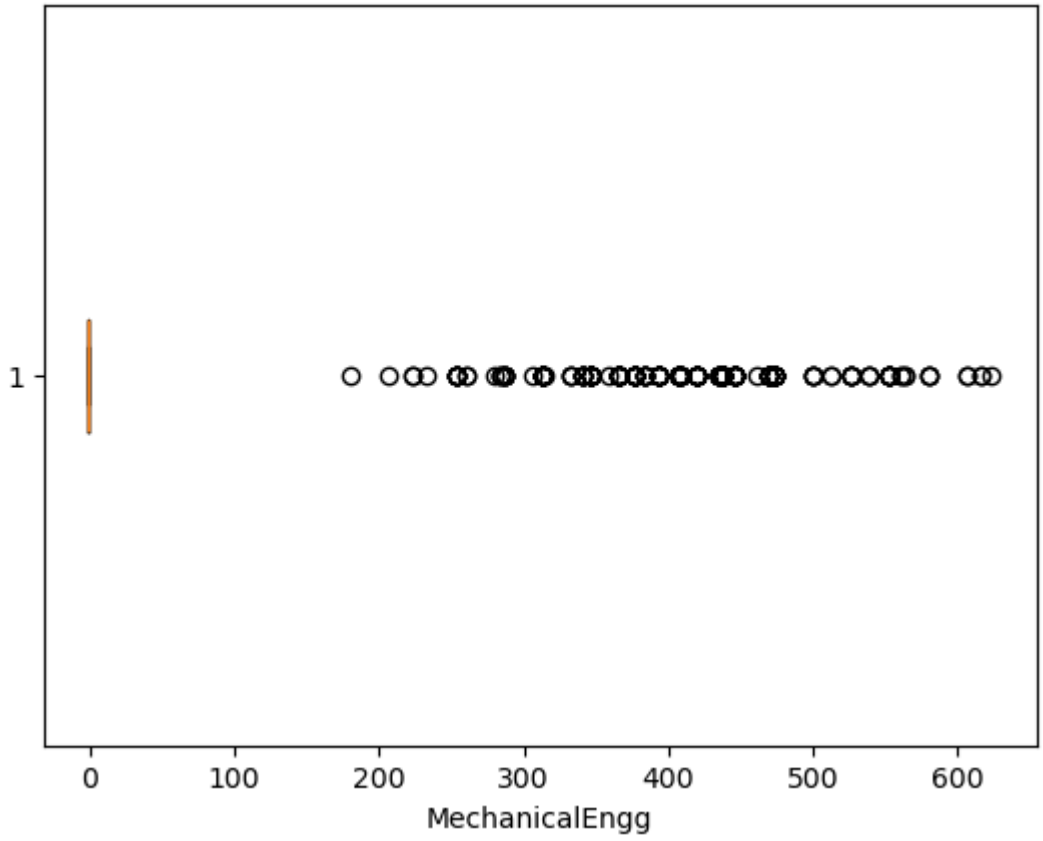

1
100
200
300
400
500
600
700
800
900
Quant

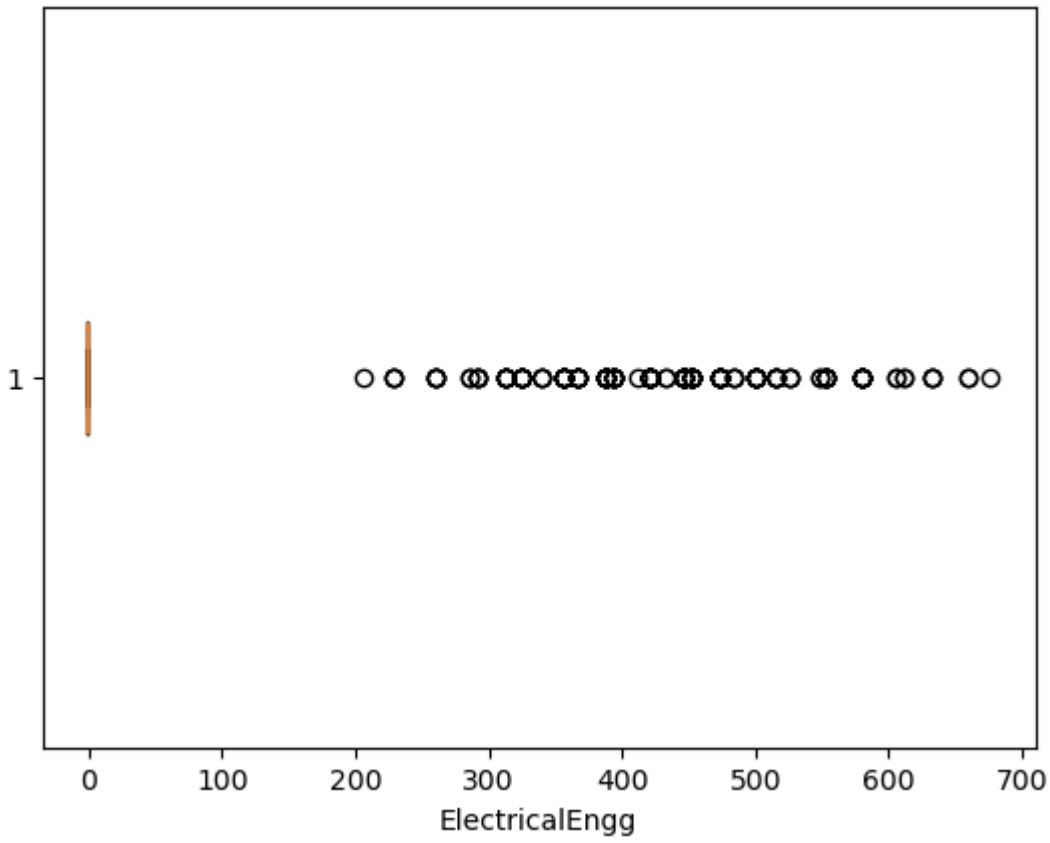

1
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Domain

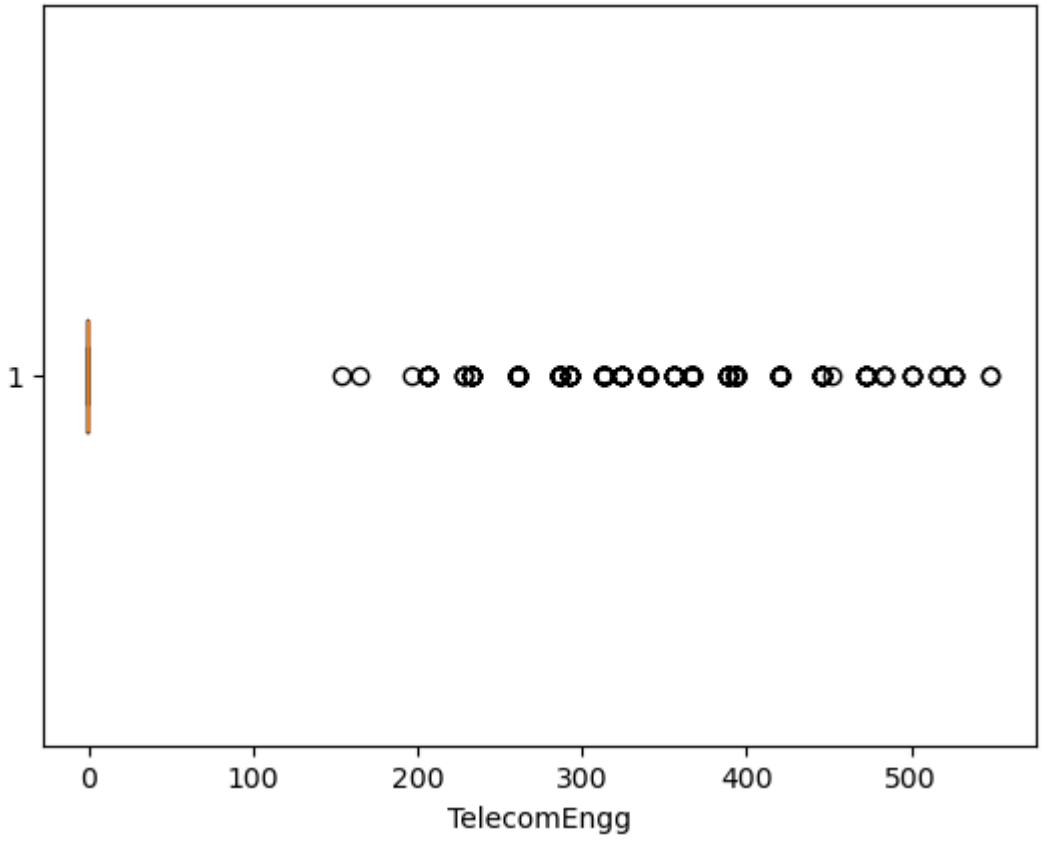

1
0
200
400
600
800
ComputerProgramming

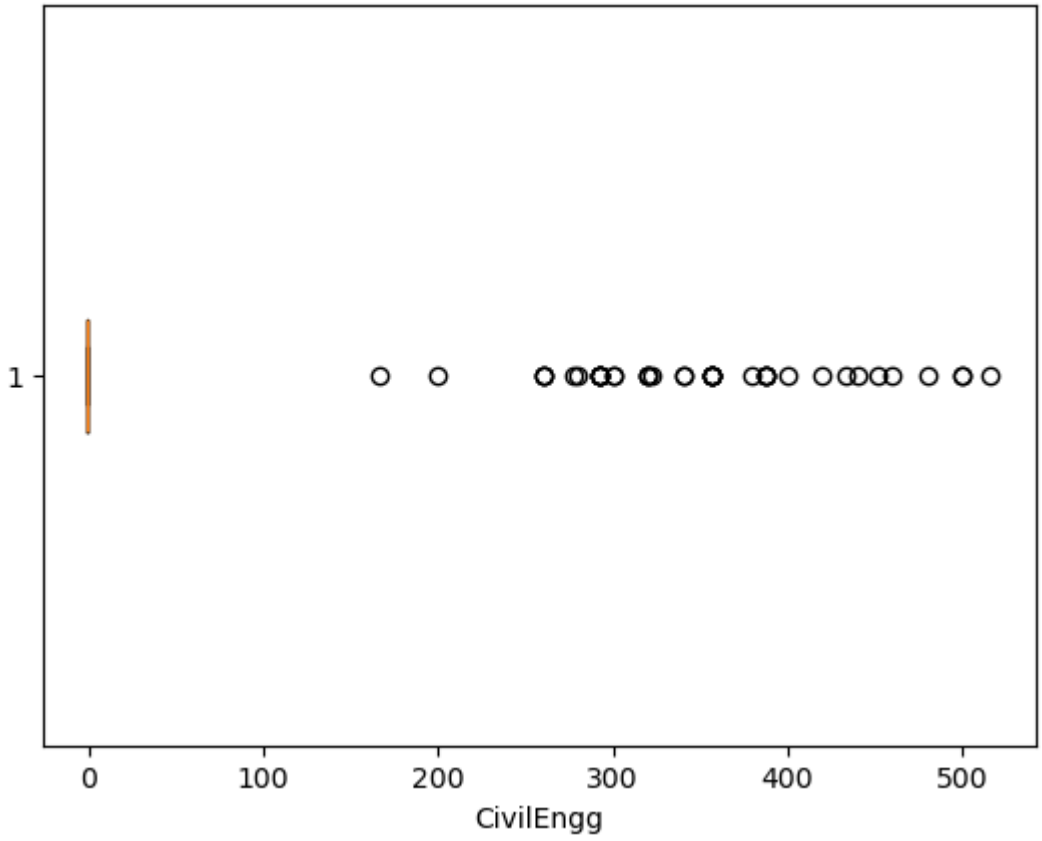

1
0
100
200
300
400
500
600
ElectronicsAndSemicon

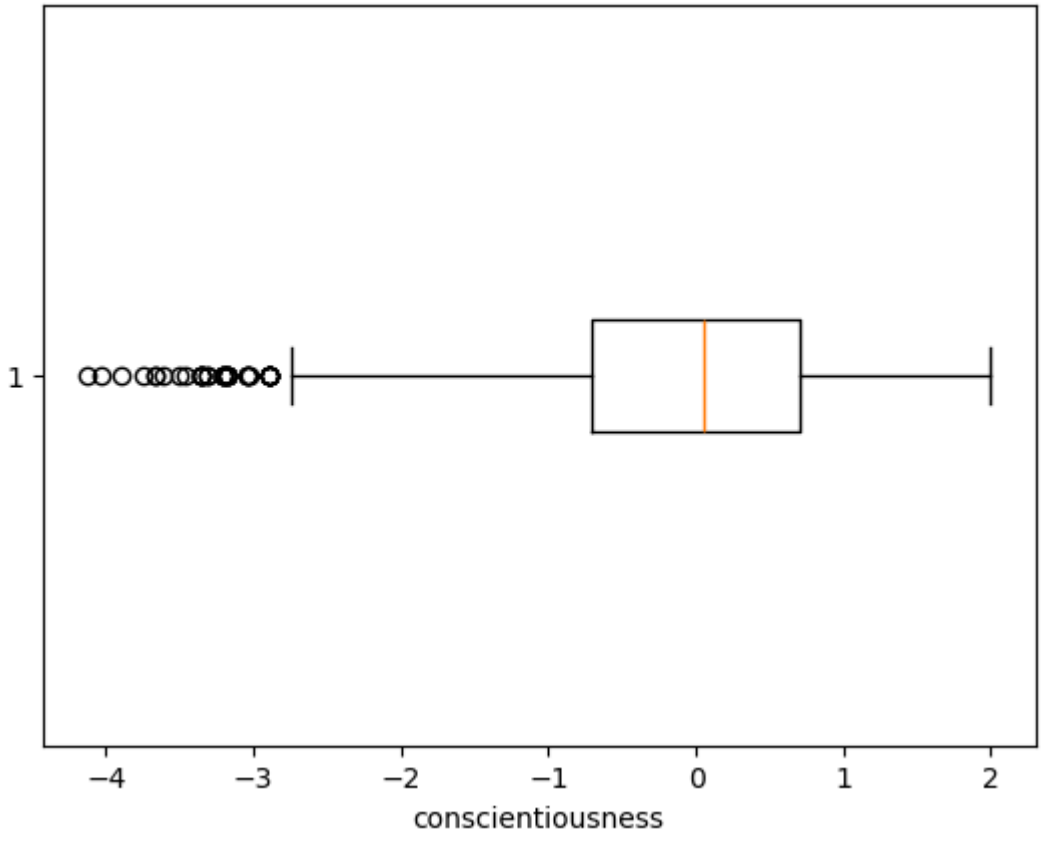

1
0
100
200
300
400
500
600
700
ComputerScience

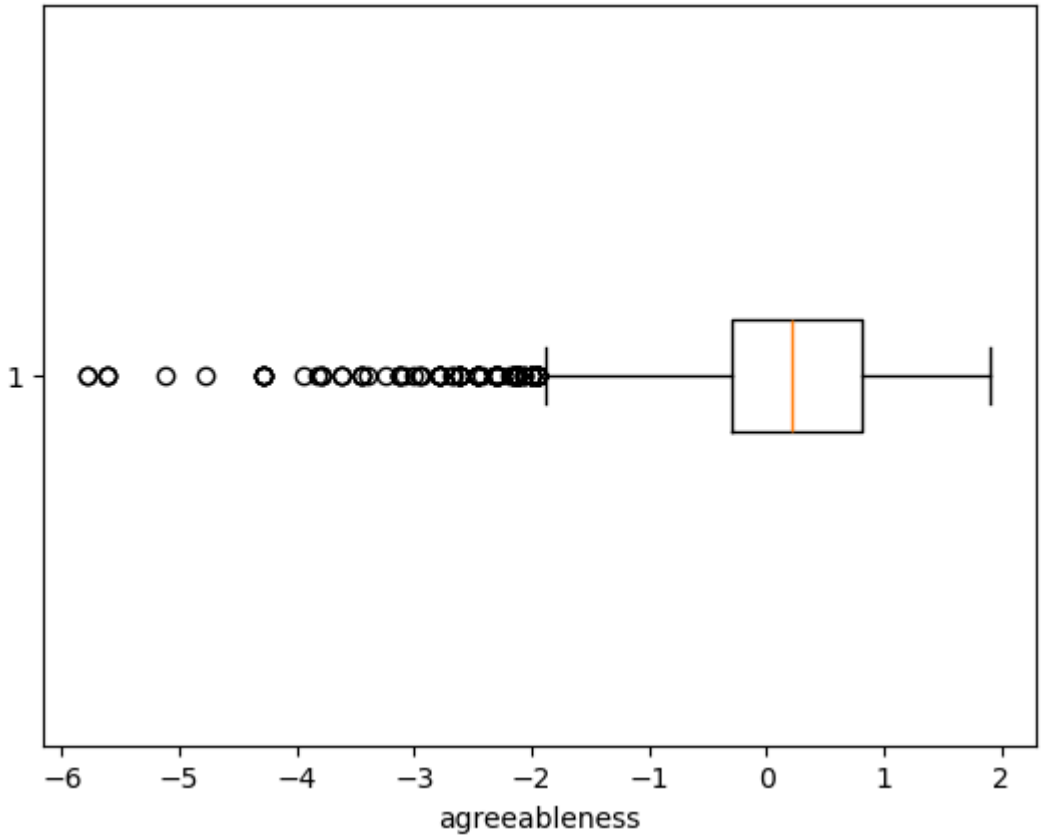

1
0
100
200
300
400
500
600
MechanicalEngg

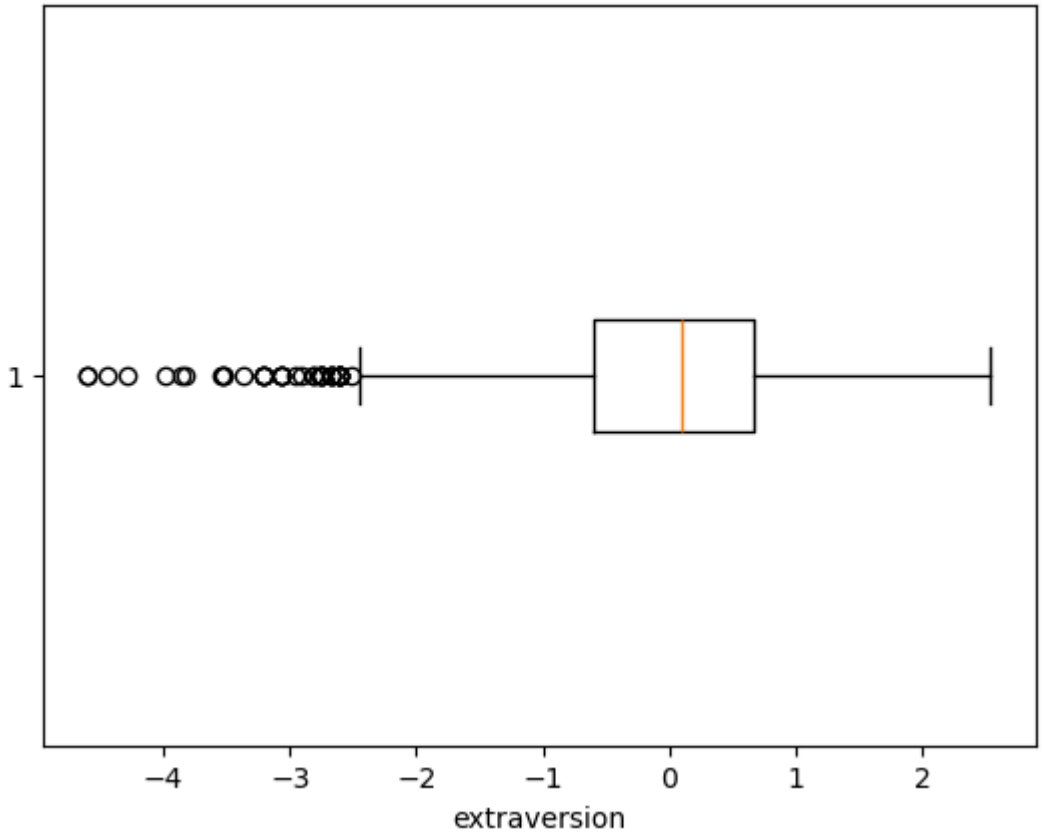

1
0
100
200
300
400
500
600
700
ElectricalEngg

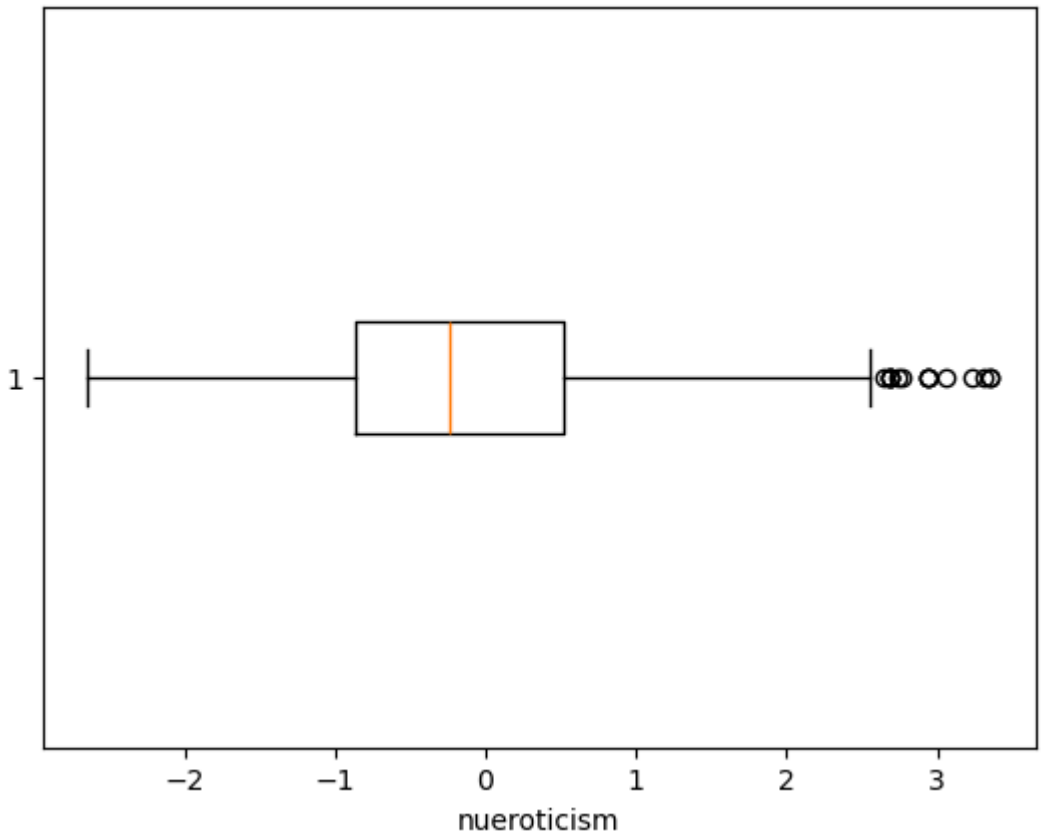

1
0
100
200
300
400
500
TelecomEngg

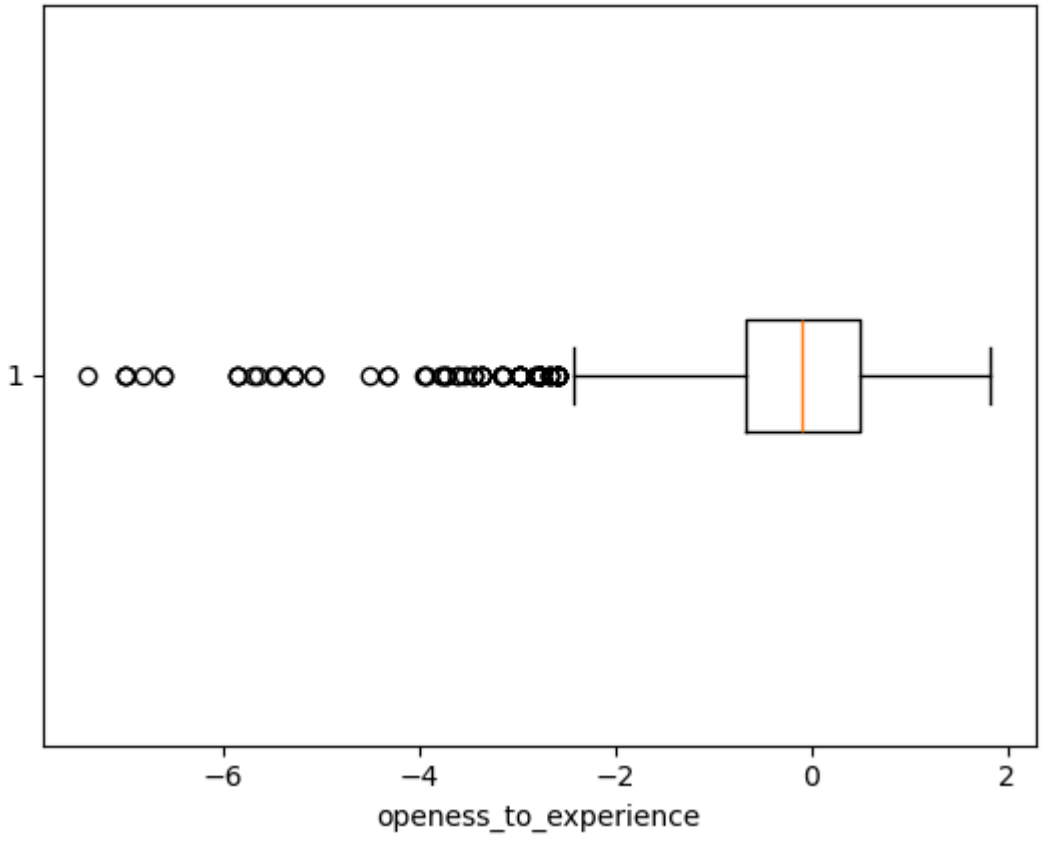

1
0
100
200
300
400
500
CivilEngg


1
−4
−3
−2
−1
0
1
2
conscientiousness


1
−6
−5
−4
−3
−2
−1
0
1
2
agreeableness


1
−4
−3
−2
−1
0
1
2
extraversion

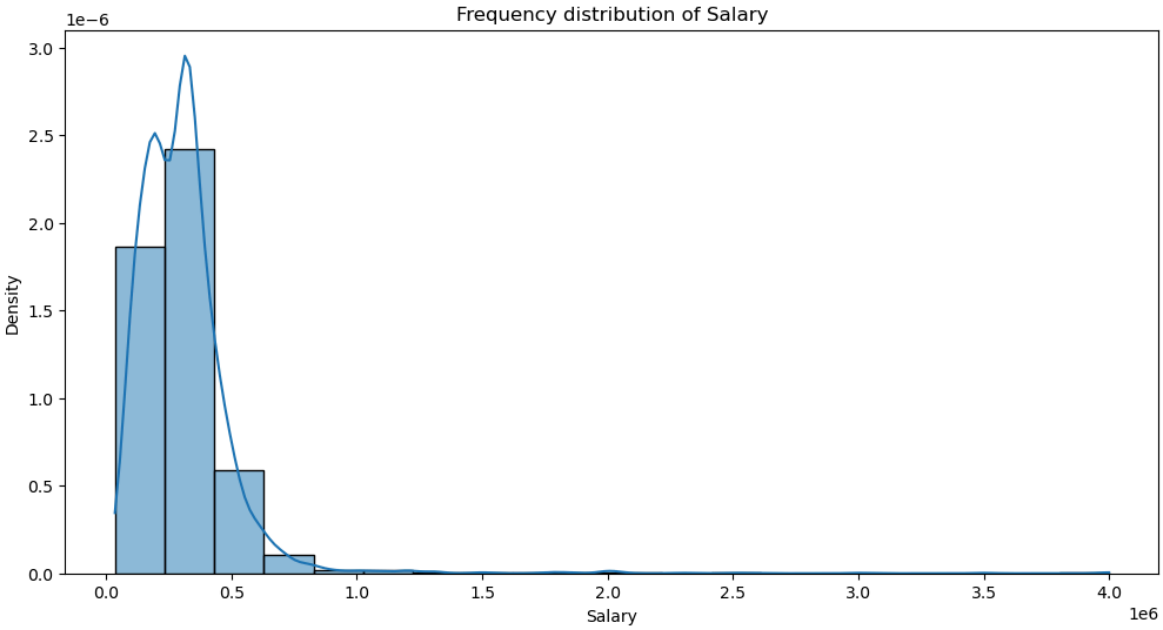




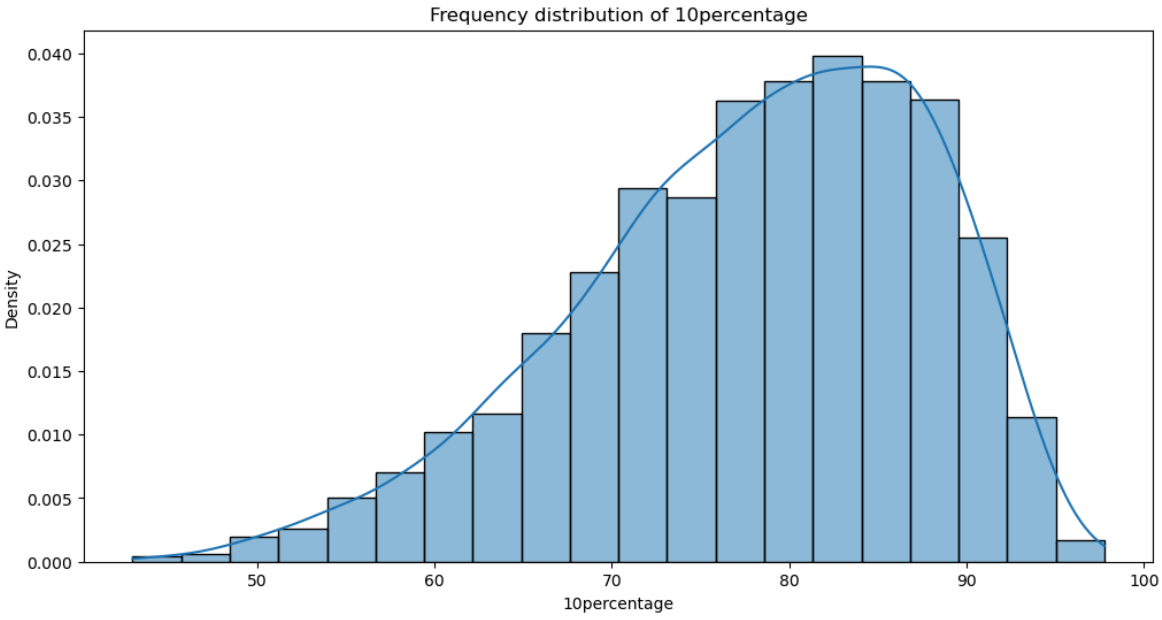
The probability and frequency distribution of each numerical column

In [24]:
```
import warnings
warnings.filterwarnings('ignore')
for i in num_cols[1:]:
    plt.figure(figsize=(12,6))
    sns.histplot(numerical_cols[i],bins=20,kde=True,stat='density')
```

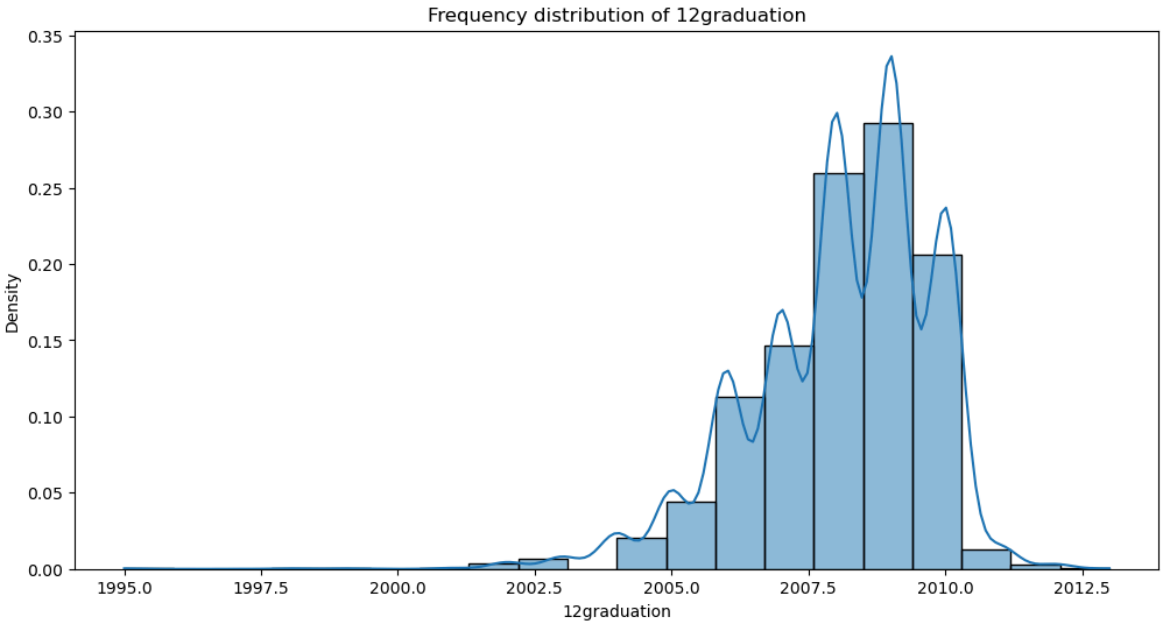
```
    plt.title(f"Frequency distribution of {i}")
    plt.xlabel(i)
    plt.ylabel('Density')
    plt.show()

    prob_dist=numerical_cols[i].value_counts(normalize=True)
    print(f"Probablity distribution of {i}: \n{prob_dist}\n")
```



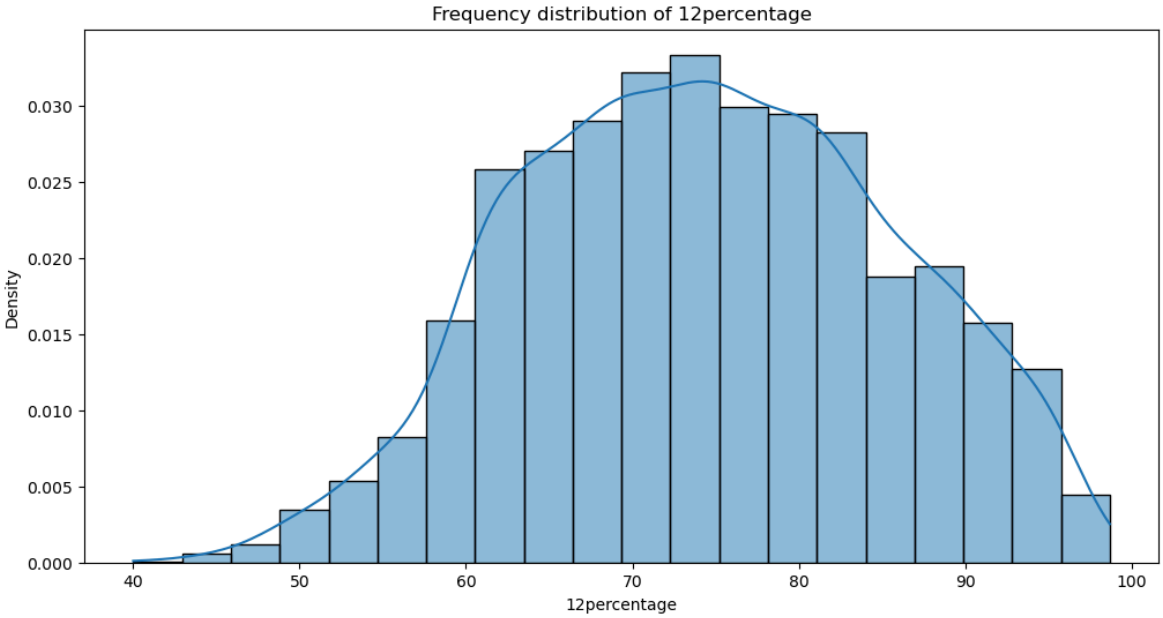

```
Probablity distribution of Salary:
Salary
300000.0     0.073287
180000.0     0.059780
200000.0     0.051276
325000.0     0.047024
120000.0     0.041271
               ...
2050000.0    0.000250
144000.0     0.000250
1320000.0    0.000250
755000.0     0.000250
925000.0     0.000250
Name: proportion, Length: 177, dtype: float64
```



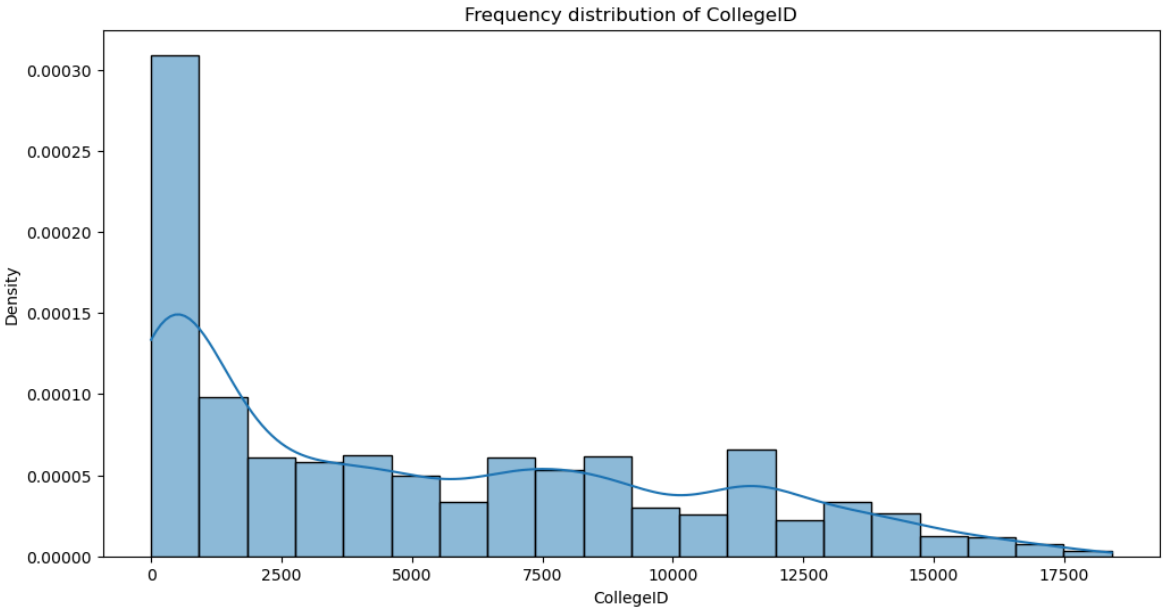
```
Probablity distribution of 10percentage:
10percentage
78.00    0.019010
82.00    0.017759
85.00    0.016758
76.00    0.016508
80.00    0.016258
           ...
82.56    0.000250
87.04    0.000250
81.14    0.000250
61.75    0.000250
78.72    0.000250
Name: proportion, Length: 851, dtype: float64
```



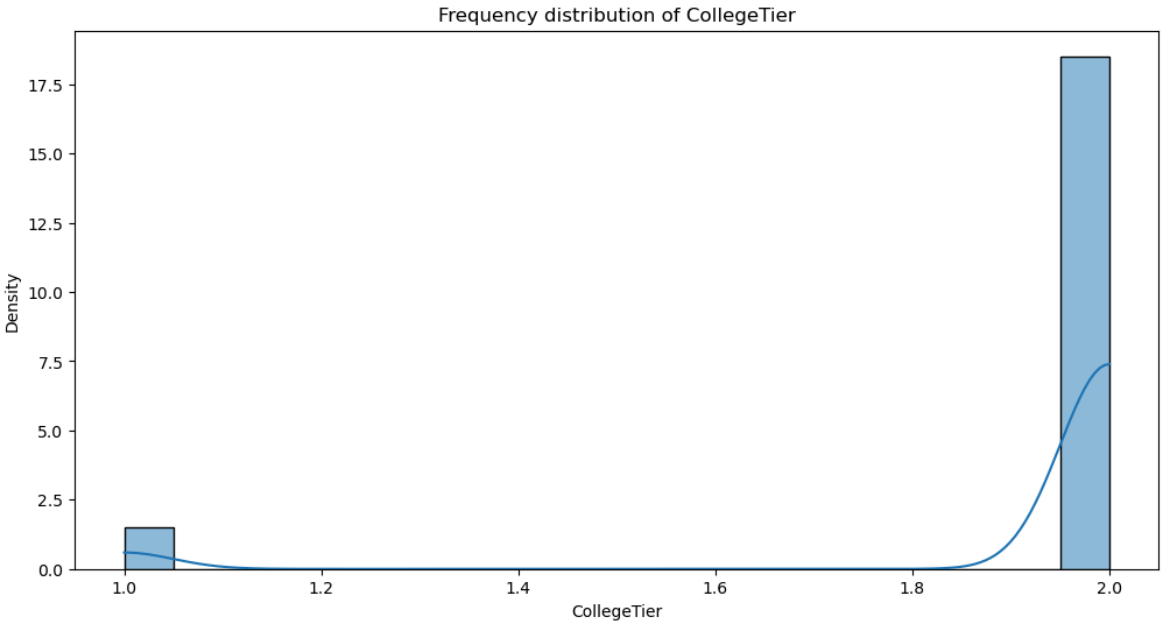

```
Probablity distribution of 12graduation:
12graduation
2009    0.263132
2008    0.233867
2010    0.185593
2007    0.132066
2006    0.101801
2005    0.040020
2004    0.018259
2011    0.011506
2003    0.006253
2002    0.003502
2012    0.002501
2001    0.000500
1995    0.000250
1998    0.000250
2013    0.000250
1999    0.000250
Name: proportion, dtype: float64
```



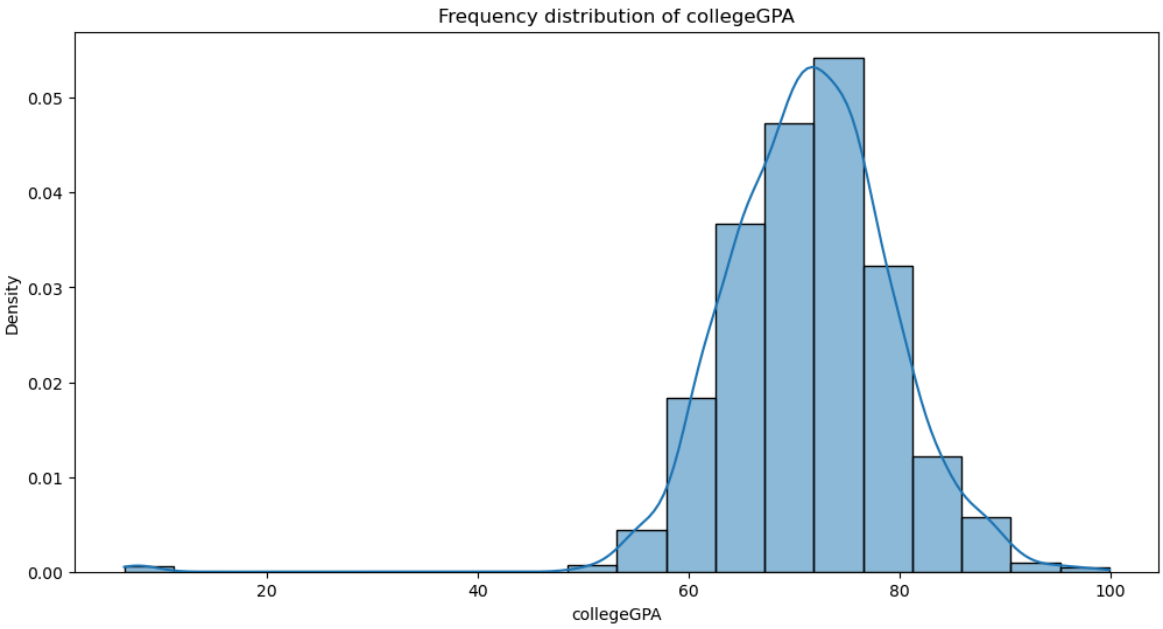

```
Probablity distribution of 12percentage:
12percentage
70.00    0.018009
72.00    0.017009
74.00    0.015758
62.00    0.014507
68.00    0.014507
           ...
58.50    0.000250
74.45    0.000250
95.41    0.000250
83.58    0.000250
82.55    0.000250
Name: proportion, Length: 801, dtype: float64
```



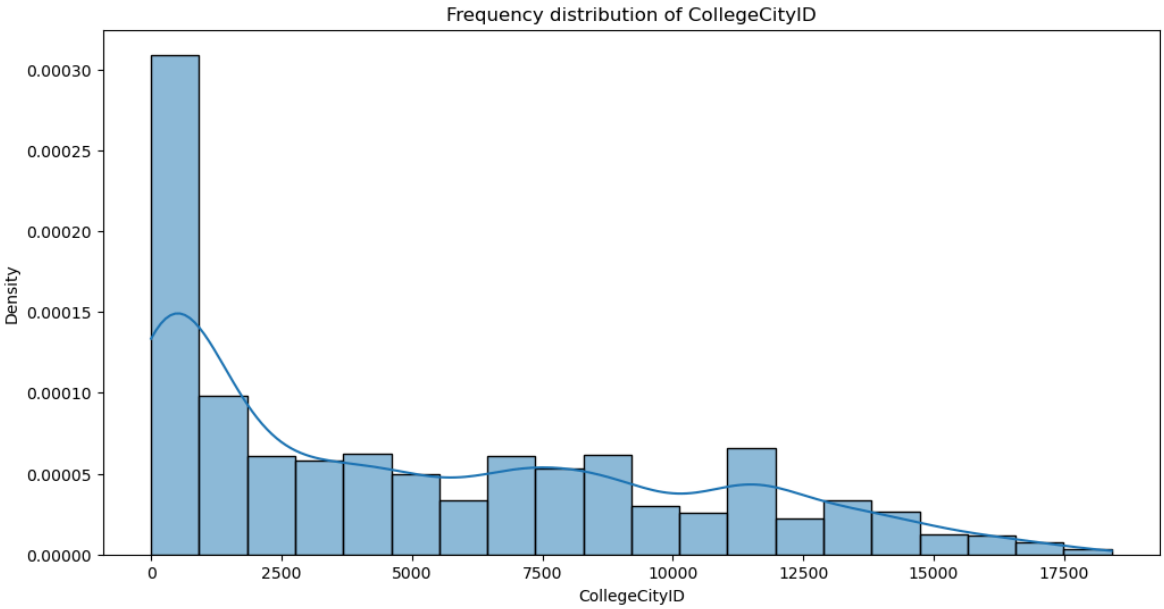
```
Probablity distribution of CollegeID:
CollegeID
272      0.023512
64       0.009505
11759    0.008754
44       0.008754
47       0.008254
           ...
128      0.000250
5068     0.000250
8637     0.000250
9361     0.000250
4883     0.000250
Name: proportion, Length: 1350, dtype: float64
```



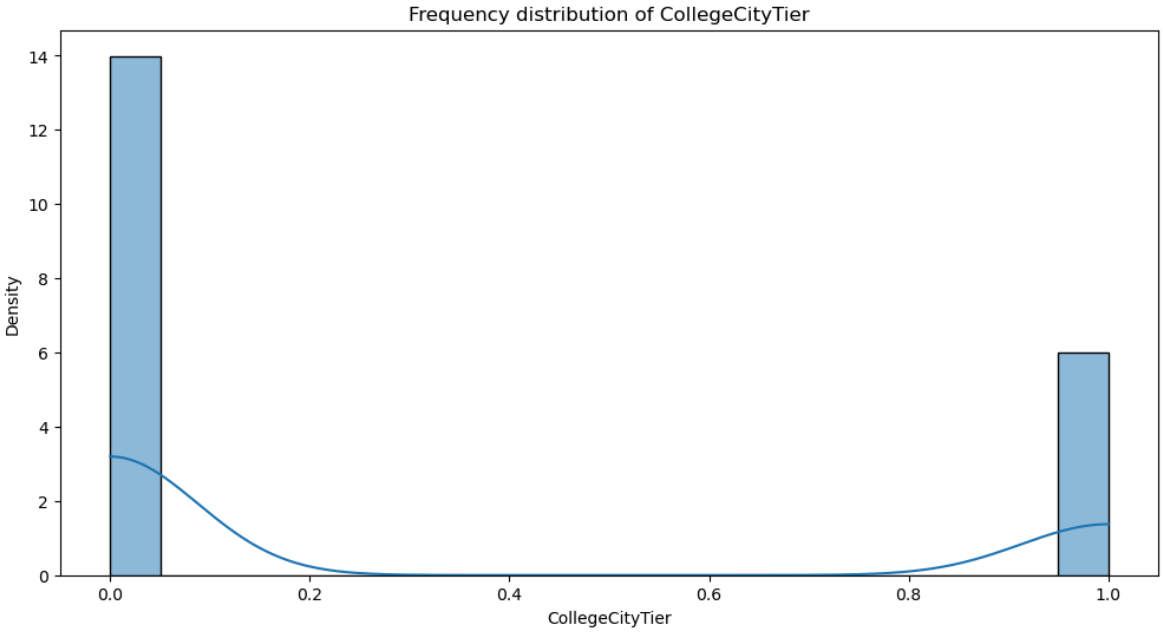

```
Probablity distribution of CollegeTier:
CollegeTier
2    0.925713
1    0.074287
Name: proportion, dtype: float64
```



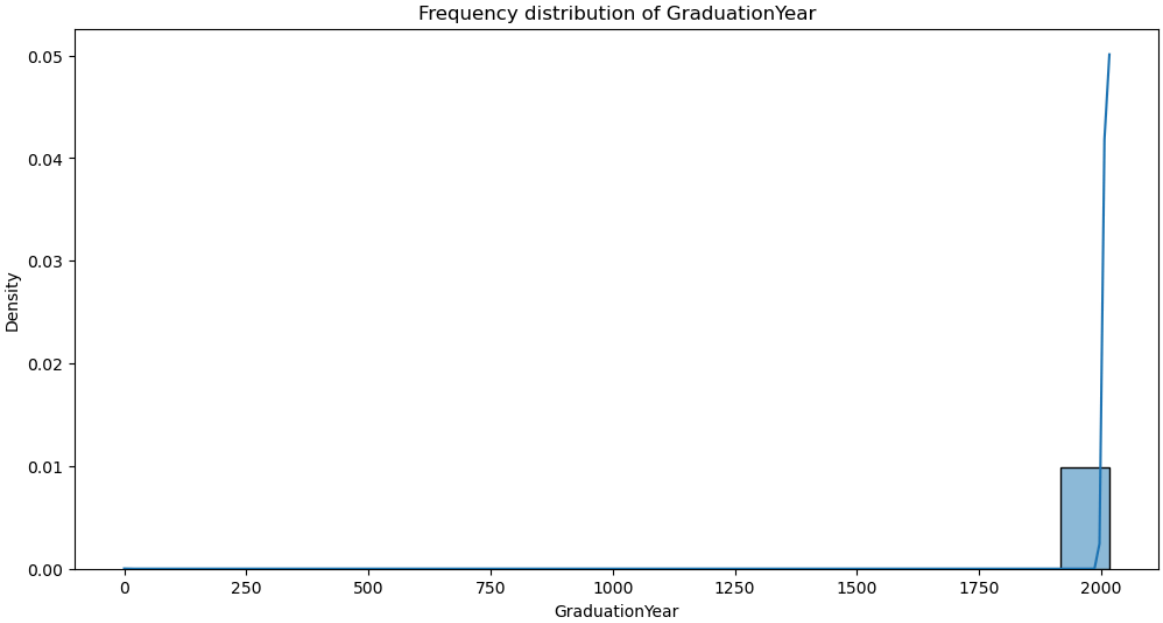
```
Probablity distribution of collegeGPA:
collegeGPA
70.00    0.028014
72.00    0.024762
75.00    0.020760
65.00    0.019760
71.00    0.018759
           ...
71.68    0.000250
73.15    0.000250
90.01    0.000250
71.36    0.000250
70.42    0.000250
Name: proportion, Length: 1282, dtype: float64
```



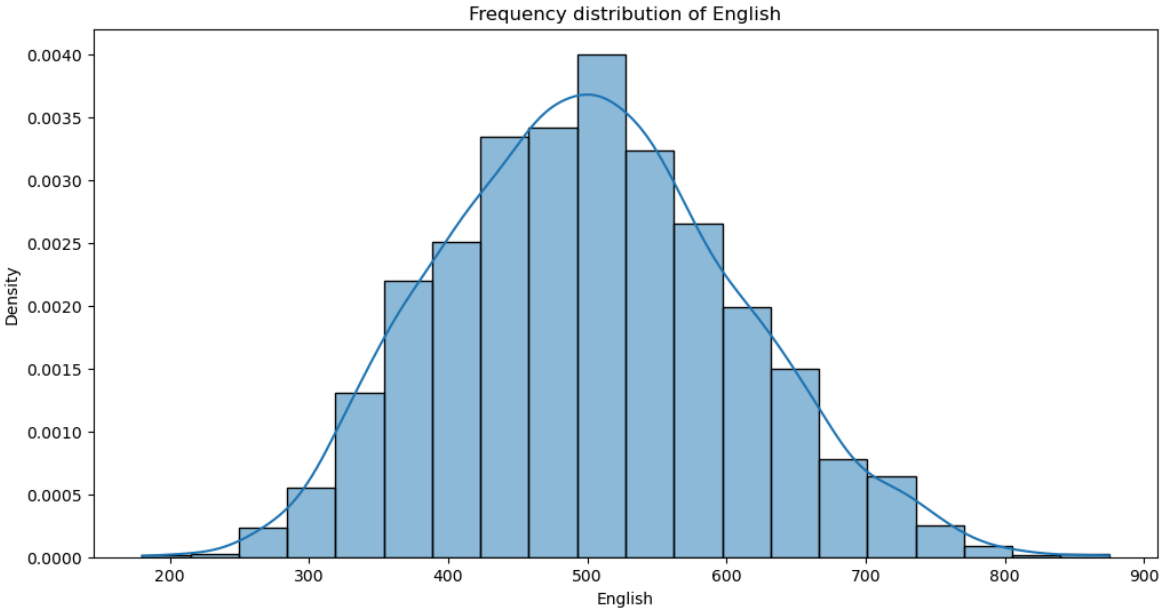

```
Probablity distribution of CollegeCityID:
CollegeCityID
272      0.023512
64       0.009505
11759    0.008754
44       0.008754
47       0.008254
           ...
128      0.000250
5068     0.000250
8637     0.000250
9361     0.000250
4883     0.000250
Name: proportion, Length: 1350, dtype: float64
```



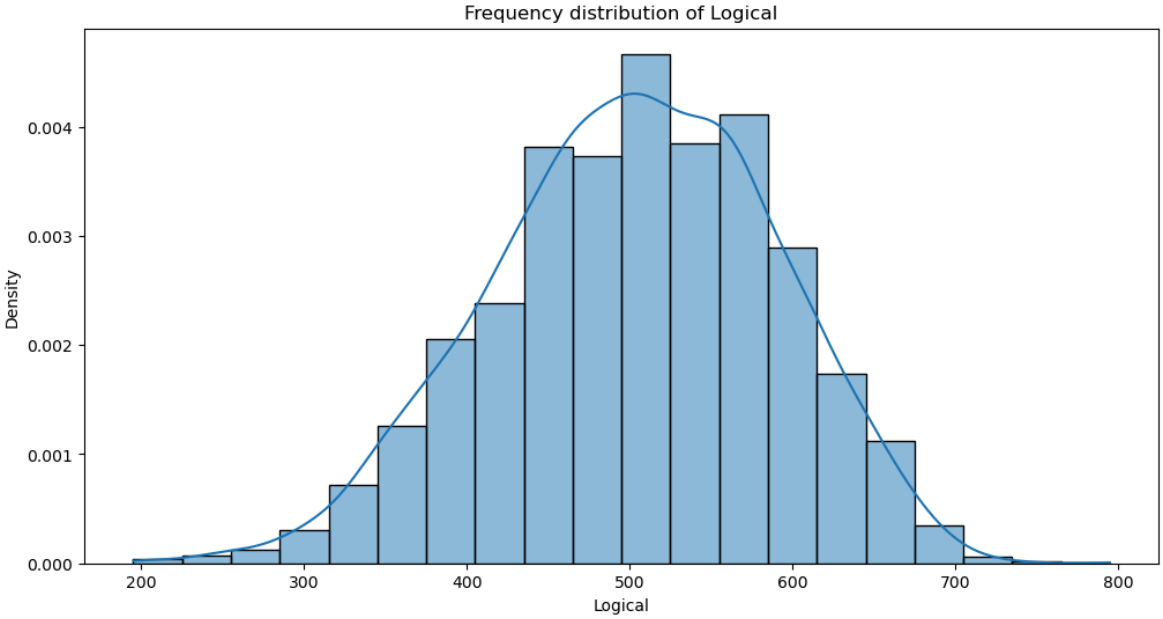

```
Probablity distribution of CollegeCityTier:
CollegeCityTier
0    0.6996
1    0.3004
Name: proportion, dtype: float64
```



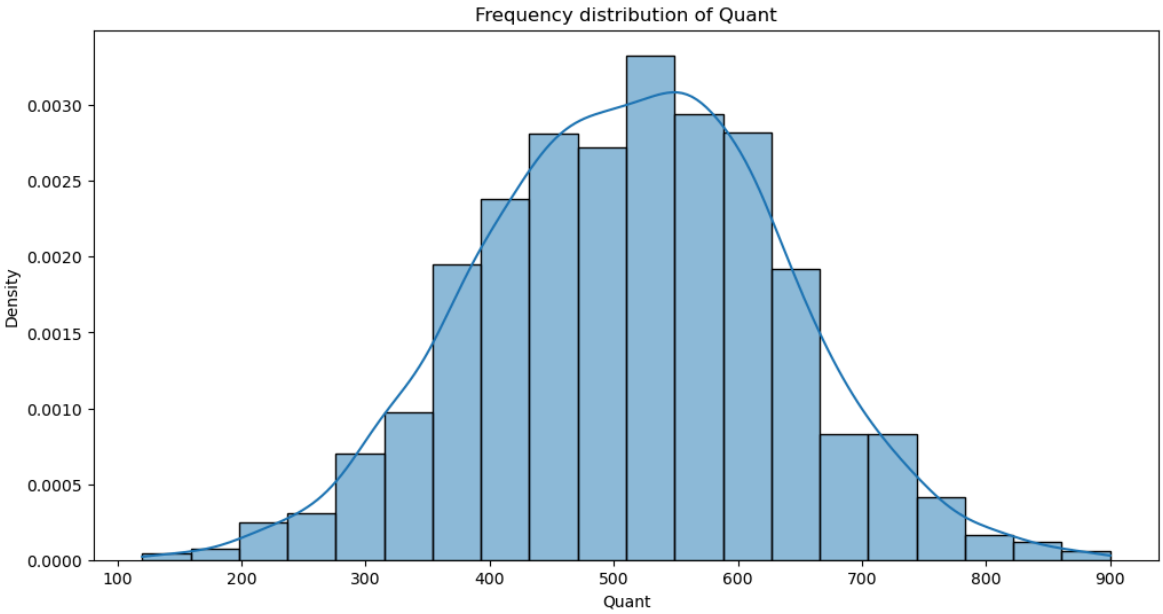

```
Probablity distribution of GraduationYear:
GraduationYear
2013    0.295398
2014    0.259130
2012    0.211856
2011    0.126813
2010    0.073037
2015    0.023512
2009    0.006003
2017    0.002001
2016    0.001751
0       0.000250
2007    0.000250
Name: proportion, dtype: float64
```



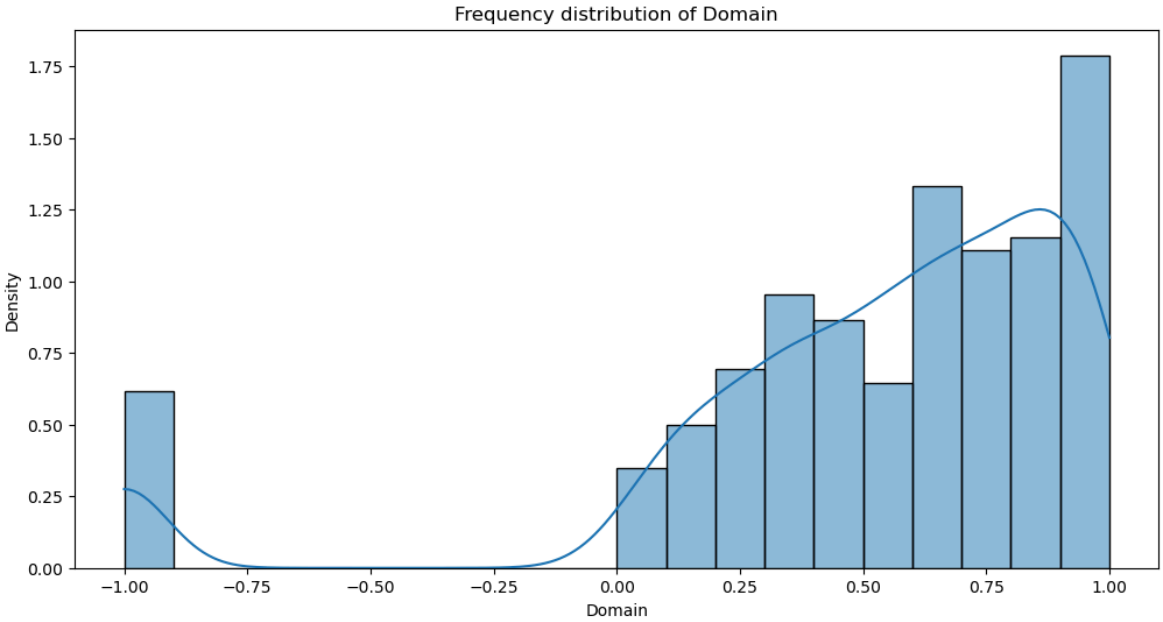

```
Probablity distribution of English:
English
475    0.040020
545    0.037769
465    0.037519
535    0.034517
405    0.027764
         ...
180    0.000250
875    0.000250
825    0.000250
870    0.000250
334    0.000250
Name: proportion, Length: 111, dtype: float64
```



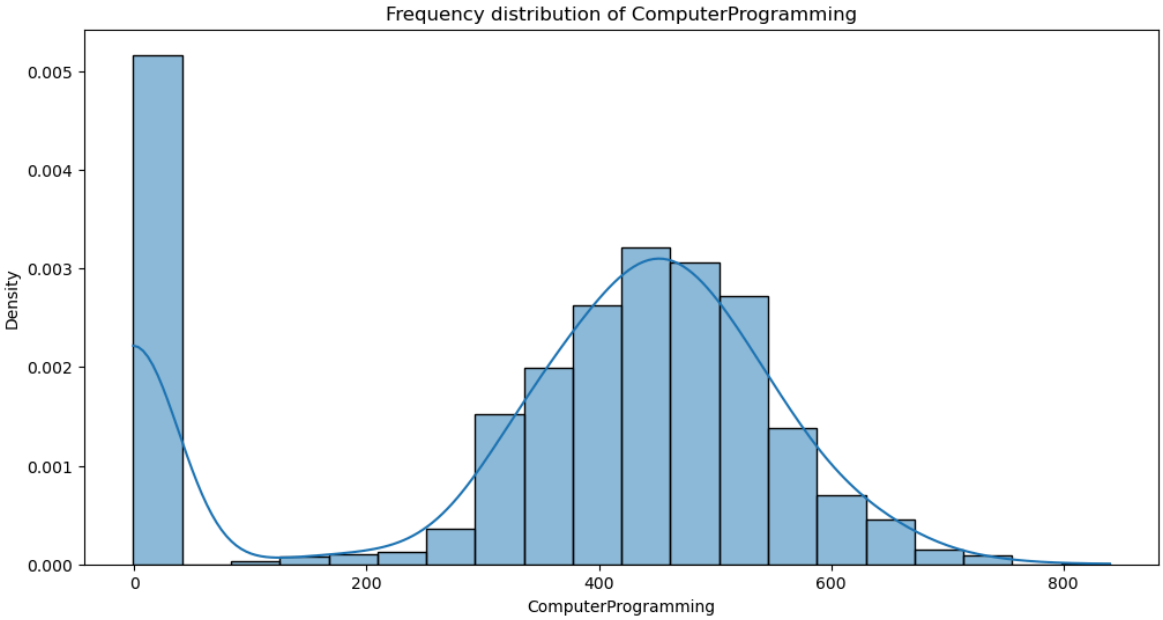
```
Probablity distribution of Logical:
Logical
495    0.039520
545    0.037769
555    0.037769
485    0.037769
505    0.029265
         ...
310    0.000250
795    0.000250
534    0.000250
454    0.000250
660    0.000250
Name: proportion, Length: 107, dtype: float64
```



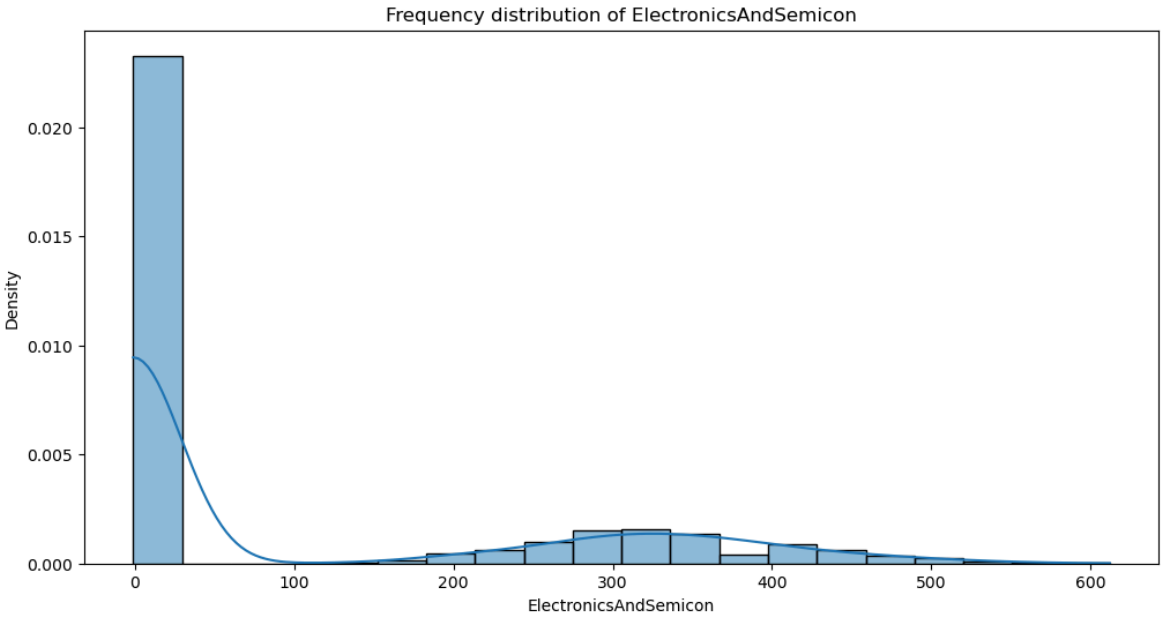

```
Probablity distribution of Quant:
Quant
605    0.035768
485    0.032516
545    0.031266
575    0.029015
515    0.024762
         ...
805    0.000250
175    0.000250
214    0.000250
860    0.000250
394    0.000250
Name: proportion, Length: 138, dtype: float64
```



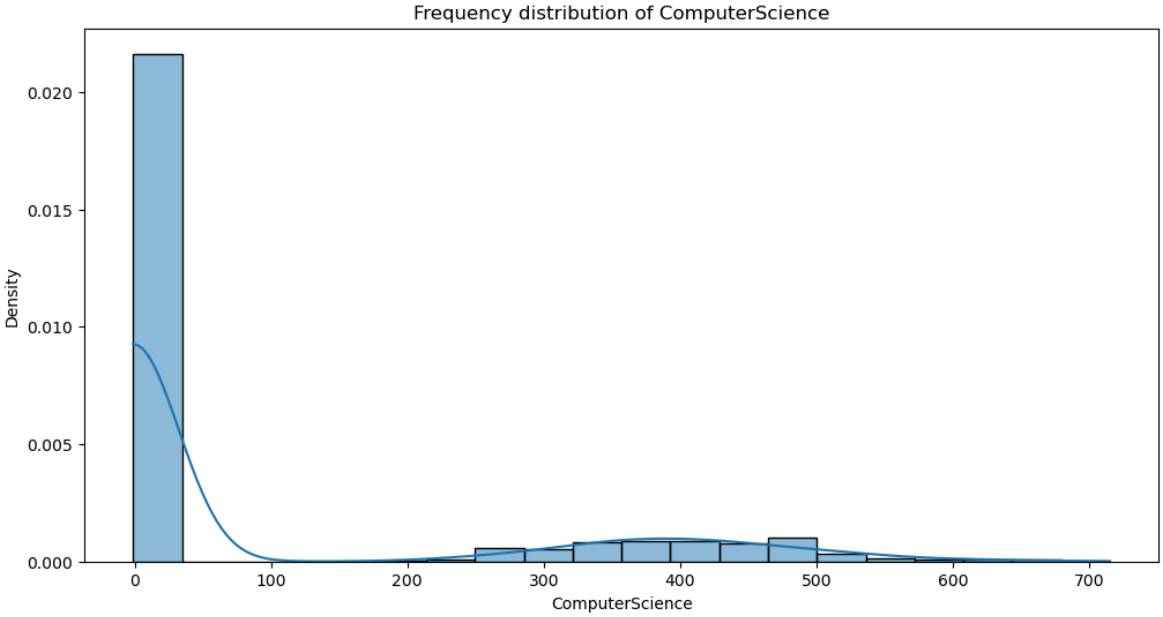

```
Probablity distribution of Domain:
Domain
-1.000000    0.061531
 0.622643    0.028264
 0.538387    0.027514
 0.486747    0.026513
 0.744758    0.025763
               ...
 0.999250    0.000250
 0.010995    0.000250
 0.639587    0.000250
 0.031150    0.000250
 0.938588    0.000250
Name: proportion, Length: 243, dtype: float64
```



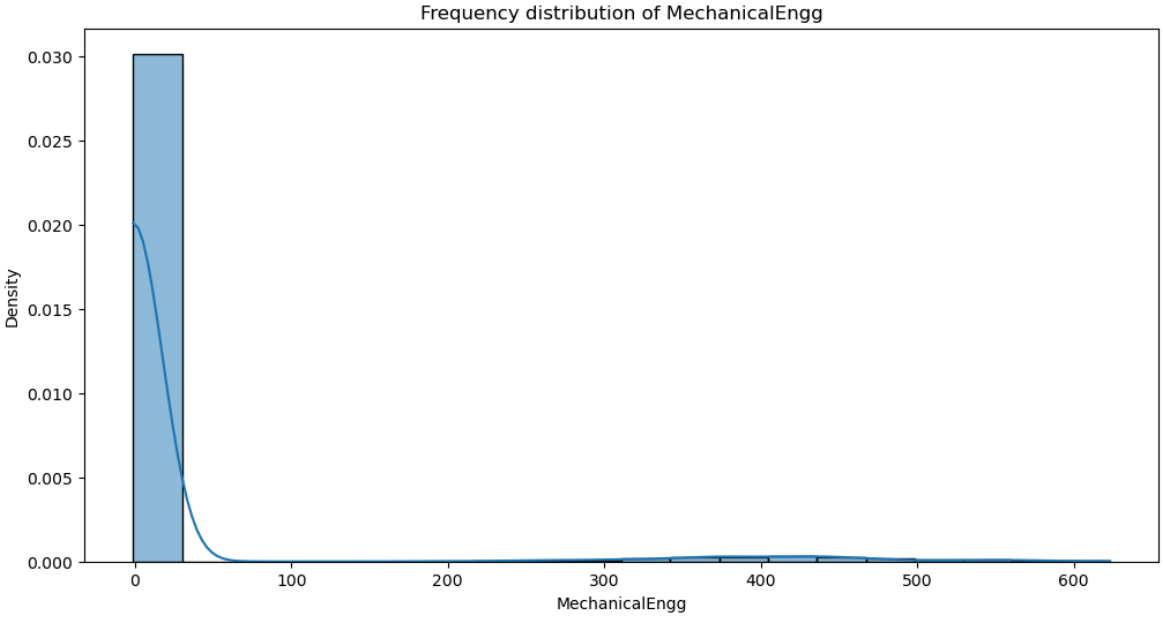
```
Probablity distribution of ComputerProgramming:
ComputerProgramming
-1      0.217109
 445    0.037769
 435    0.036018
 475    0.034767
 465    0.033517
          ...
 214    0.000250
 494    0.000250
 840    0.000250
 394    0.000250
 554    0.000250
Name: proportion, Length: 79, dtype: float64
```



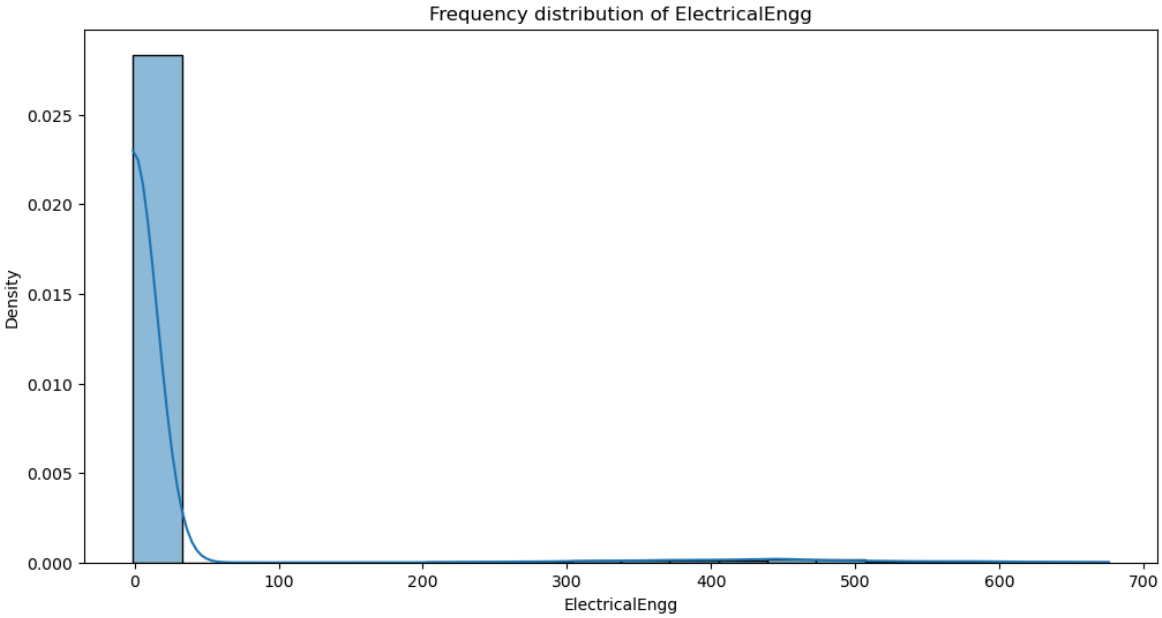
```
Probablity distribution of ElectronicsAndSemicon:
ElectronicsAndSemicon
-1      0.713857
 333    0.031016
 300    0.028514
 366    0.025763
 266    0.022011
 400    0.021011
 292    0.018509
 324    0.016508
 356    0.016508
 233    0.013257
 388    0.011756
 433    0.011506
 200    0.010755
 260    0.009005
 466    0.007754
 452    0.007254
 228    0.006003
 500    0.005753
 420    0.005503
 196    0.004002
 166    0.003002
 484    0.002501
 516    0.002251
 533    0.001501
 548    0.001251
 566    0.001001
 133    0.001001
 164    0.000750
 612    0.000500
Name: proportion, dtype: float64
```



```
Probablity distribution of ComputerScience:
ComputerScience
-1      0.774387
 407    0.032016
 376    0.030765
 346    0.029515
 438    0.027764
 469    0.020010
 315    0.019260
 500    0.016008
 284    0.012506
 530    0.011256
 253    0.007504
 561    0.005503
 223    0.003502
 592    0.003502
 623    0.002501
 653    0.002251
 192    0.000750
 715    0.000500
 684    0.000250
 130    0.000250
Name: proportion, dtype: float64
```



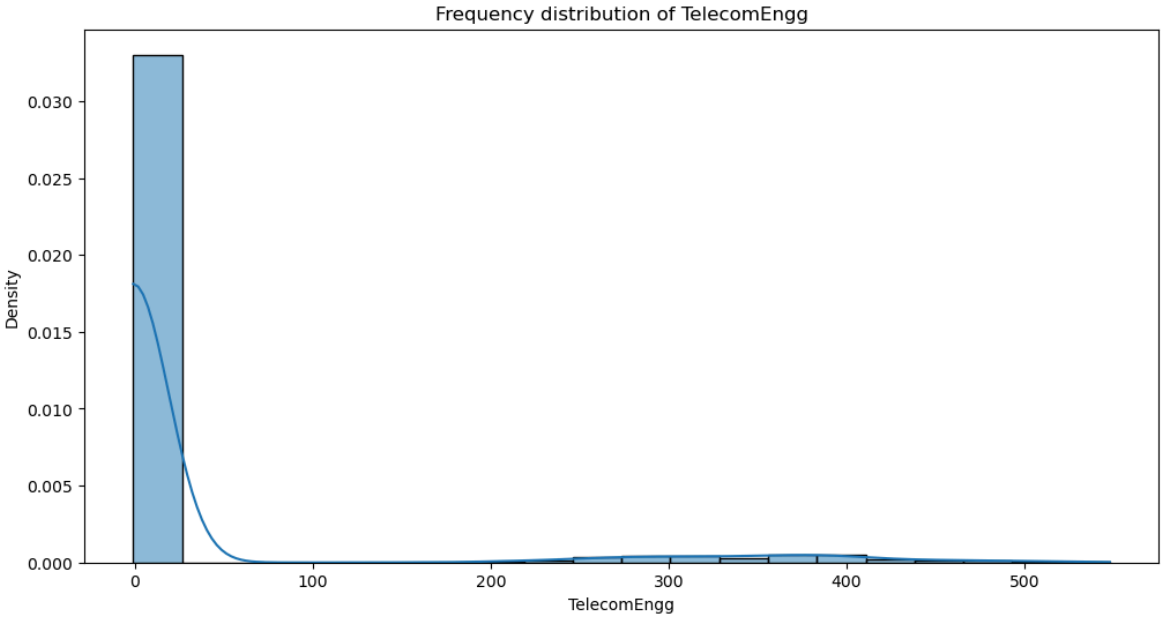
Probablity distribution of MechanicalEngg:
MechanicalEngg
-1   0.941221
 366   0.005003
 446   0.004002
 438   0.003752
 420   0.003502
 376   0.003252
 313   0.003252
 393   0.003252
 407   0.003002
 346   0.002751
 469   0.002501
 473   0.002501
 553   0.002001
 435   0.001751
 383   0.001501
 340   0.001501
 526   0.001251
 409   0.001251
 286   0.001251
 500   0.001001
 253   0.001001
 284   0.000750
 332   0.000750
 538   0.000750
 254   0.000750
 580   0.000750
 616   0.000500
 564   0.000500
 606   0.000500
 223   0.000500
 512   0.000500
 561   0.000500
 260   0.000500
 358   0.000250
 280   0.000250
 315   0.000250
 233   0.000250
 306   0.000250
 461   0.000250
 180   0.000250
 206   0.000250
 623   0.000250
Name: proportion, dtype: float64



Probablity distribution of ElectricalEngg:

ElectricalEngg	
-1	0.959730
420	0.004002
446	0.003502
388	0.002501
473	0.002501
452	0.002501
356	0.002251
500	0.002001
580	0.002001
366	0.001751
324	0.001751
393	0.001751
553	0.001501
313	0.001501
516	0.001251
260	0.001001
292	0.001001
340	0.000750
228	0.000750
526	0.000750
484	0.000750
633	0.000750
548	0.000500
433	0.000500
606	0.000500
612	0.000500
660	0.000500
286	0.000500
676	0.000250
411	0.000250
206	0.000250

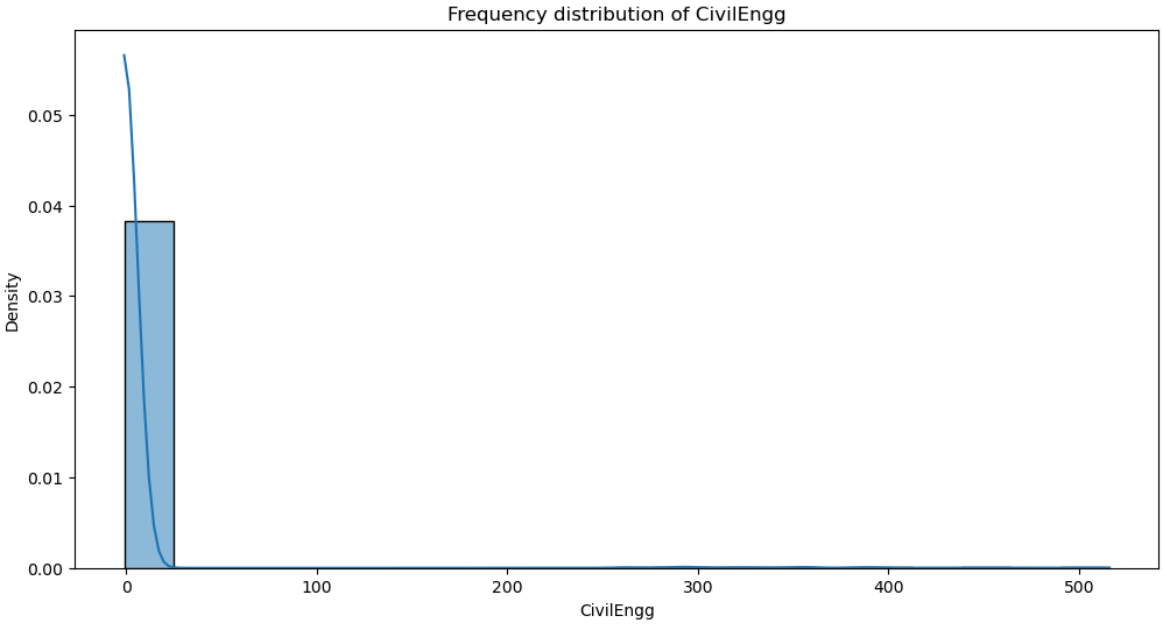
Name: proportion, dtype: float64



Probablity distribution of TelecomEngg:

TelecomEngg	
-1	0.906453
393	0.011256
366	0.010755
260	0.008754
313	0.008504
340	0.008004
286	0.007754
420	0.006503
446	0.004002
388	0.003502
233	0.003502
473	0.003252
292	0.003252
356	0.003002
324	0.002751
206	0.002001
500	0.001251
526	0.001251
516	0.001001
484	0.001001
228	0.000750
548	0.000500
153	0.000250
196	0.000250
164	0.000250
452	0.000250

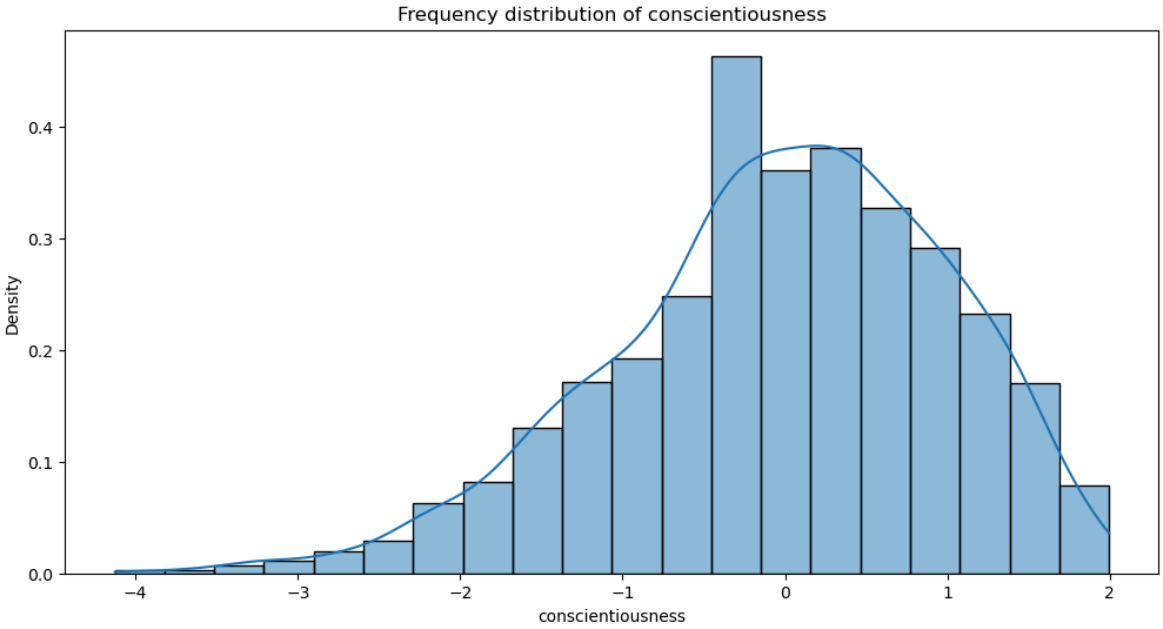
Name: proportion, dtype: float64



Probablity distribution of CivilEngg:

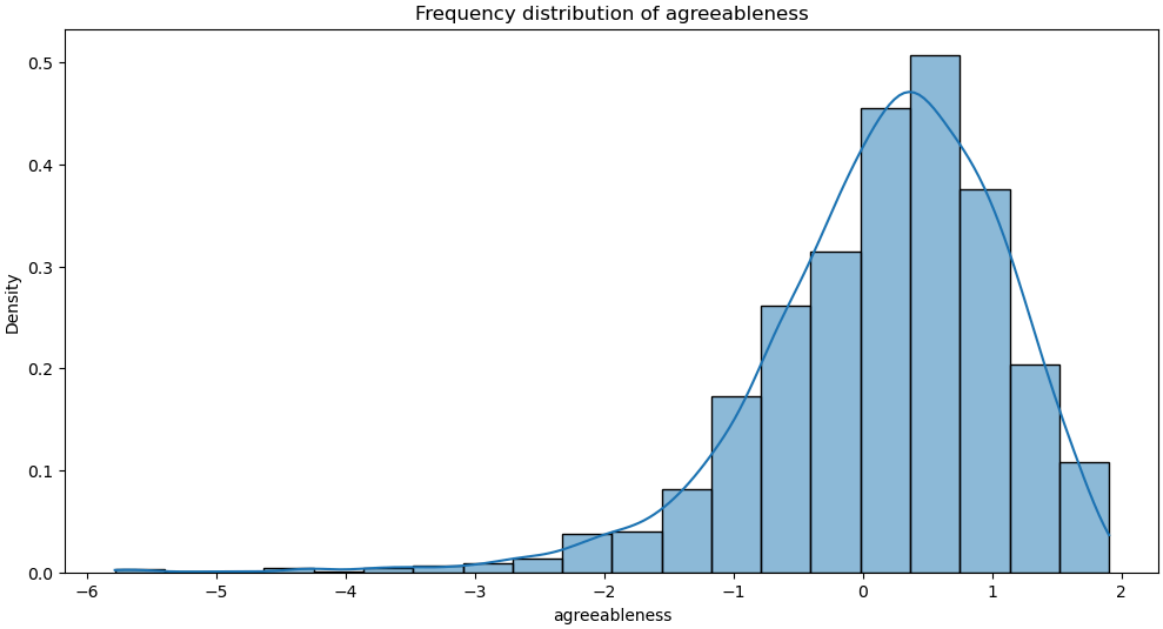
CivilEngg	
-1	0.989495
356	0.001501
292	0.001501
388	0.001001
260	0.000750
320	0.000750
500	0.000500
300	0.000500
340	0.000500
516	0.000250
460	0.000250
420	0.000250
280	0.000250
433	0.000250
380	0.000250
452	0.000250
277	0.000250
166	0.000250
322	0.000250
200	0.000250
440	0.000250
400	0.000250
480	0.000250

Name: proportion, dtype: float64

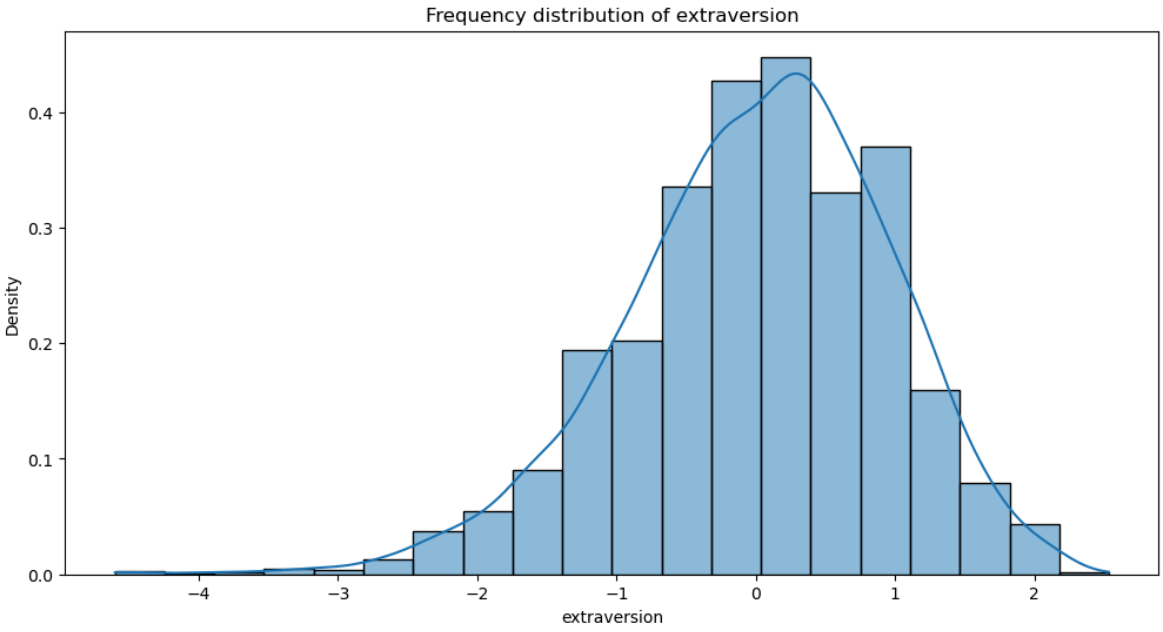


Probablity distribution of conscientiousness:

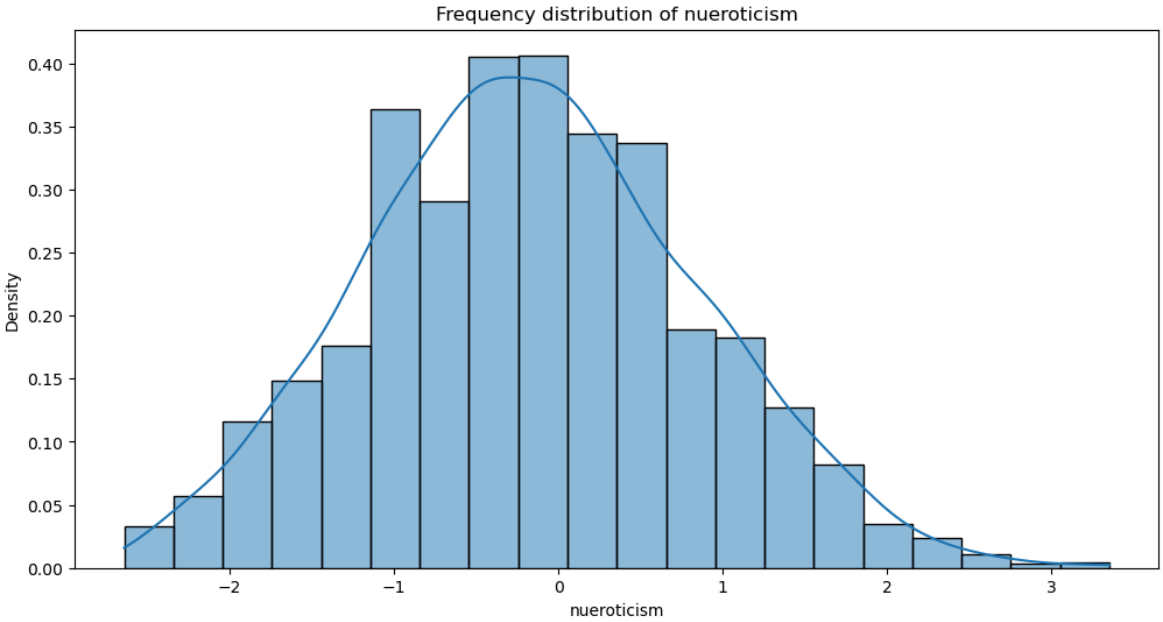
```
conscientiousness
 0.2718    0.036268
 0.1282    0.033517
-0.1590    0.033267
 0.4155    0.032766
-0.0154    0.032266
              ...
-3.4624    0.000250
-1.2950    0.000250
-0.9653    0.000250
-0.4854    0.000250
 0.8986    0.000250
Name: proportion, Length: 141, dtype: float64
```



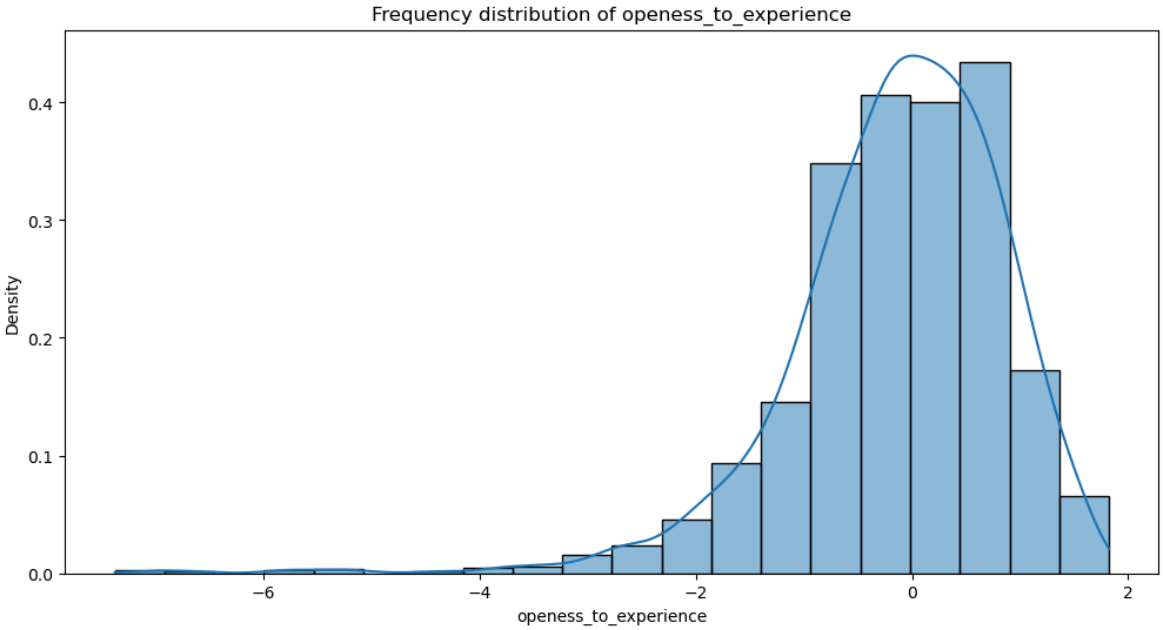
```
Probablity distribution of agreeableness:
agreeableness
 0.3789    0.048274
 0.2124    0.045023
 0.5454    0.043772
 0.0459    0.041521
 0.8784    0.039520
             ...
-3.1264    0.000250
-3.0094    0.000250
-3.9501    0.000250
-1.7223    0.000250
-0.8320    0.000250
Name: proportion, Length: 149, dtype: float64
```




```
Probablity distribution of extraversion:
extraversion
 0.4711    0.044772
 0.3174    0.044522
 0.1637    0.038769
 0.7785    0.036518
-0.1437    0.033767
             ...
-3.5370    0.000250
-0.4226    0.000250
 1.5791    0.000250
-0.1408    0.000250
-1.2056    0.000250
Name: proportion, Length: 154, dtype: float64
```




```
Probablity distribution of nueroticism:
nueroticism
-0.48790    0.031516
-0.74150    0.029515
 0.01920    0.028014
-0.61470    0.027264
-0.36120    0.026513
               ...
 1.06113    0.000250
-0.74960    0.000250
 2.76500    0.000250
 1.82493    0.000250
 2.03060    0.000250
Name: proportion, Length: 217, dtype: float64
```



```
Probablity distribution of openess_to_experience:
openess_to_experience
 0.6721    0.046773
-0.0943    0.045523
 0.0973    0.045523
 0.4805    0.044272
 0.2889    0.043522
             ...
-6.8009    0.000250
 0.1187    0.000250
-5.6860    0.000250
-1.1291    0.000250
-0.4229    0.000250
Name: proportion, Length: 142, dtype: float64
```

Frequency distribution of each categorical Variable/Column

In [25]:
```
frequency_dis={i: df[i].value_counts() for i in cat_cols}

for i,freq in frequency_dis.items():
    print(f"Frequncy distribution for {i}:\n{freq}\n")
```

```
Frequncy distribution for Unnamed: 0:
Unnamed: 0
train    3998
Name: count, dtype: int64

Frequncy distribution for DOJ:
DOJ
7/1/14 0:00     199
6/1/14 0:00     180
8/1/14 0:00     178
9/1/14 0:00     142
1/1/14 0:00     142
               ...
11/1/15 0:00      1
11/1/09 0:00      1
8/1/04 0:00       1
9/1/09 0:00       1
2/1/07 0:00       1
Name: count, Length: 81, dtype: int64

Frequncy distribution for DOL:
DOL
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: count, Length: 67, dtype: int64

Frequncy distribution for Designation:
Designation
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: count, Length: 419, dtype: int64

Frequncy distribution for JobCity:
JobCity
Bangalore           627
-1                  461
Noida               368
Hyderabad           335
Pune                290
                   ...
Tirunelvelli          1
Ernakulam             1
```

```
Nanded                1
Dharmapuri            1
Asifabadbanglore      1
Name: count, Length: 339, dtype: int64

Frequncy distribution for Gender:
Gender
m    3041
f     957
Name: count, dtype: int64

Frequncy distribution for DOB:
DOB
1/1/91 0:00      11
7/15/91 0:00     10
7/5/91 0:00       8
12/13/91 0:00     8
6/3/91 0:00       8
                 ..
12/30/92 0:00     1
10/20/86 0:00     1
11/17/89 0:00     1
9/30/92 0:00      1
4/15/87 0:00      1
Name: count, Length: 1872, dtype: int64

Frequncy distribution for 10board:
10board
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: count, Length: 275, dtype: int64

Frequncy distribution for 12board:
12board
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: count, Length: 340, dtype: int64

Frequncy distribution for Degree:
Degree
B.Tech/B.E.      3700
MCA               243
```

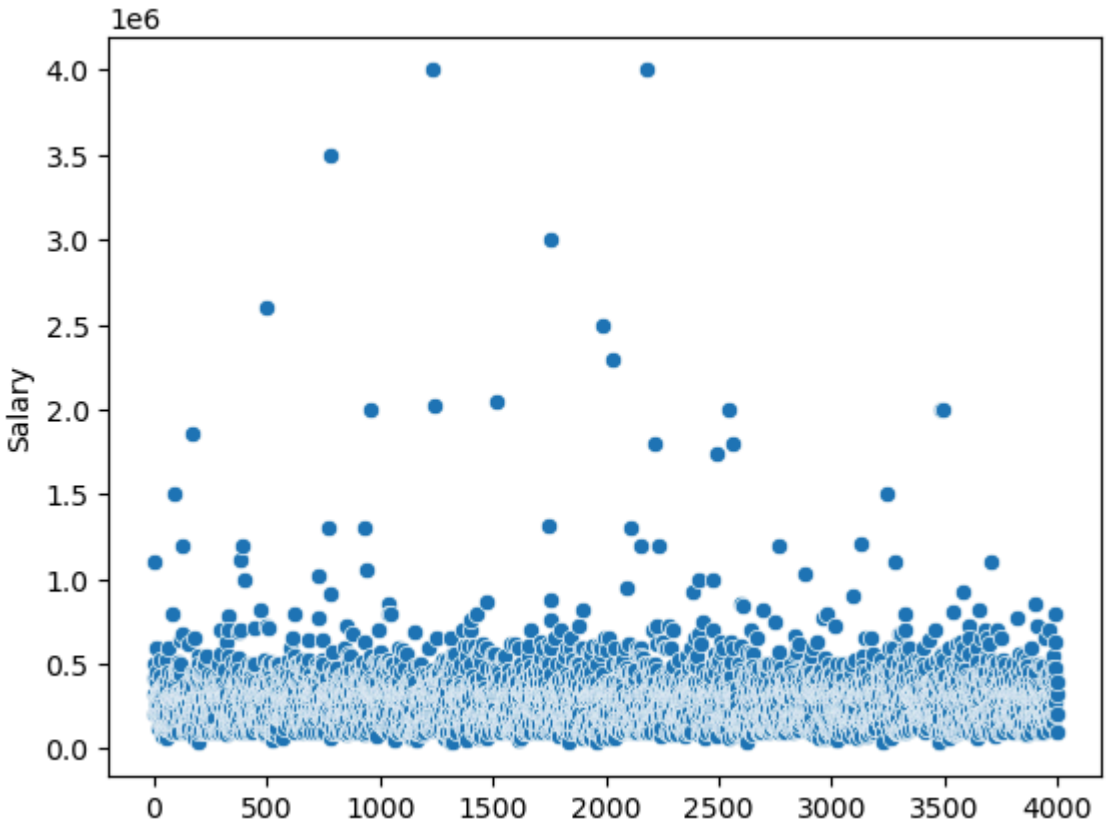
```
M.Tech./M.E.       53
M.Sc. (Tech.)       2
Name: count, dtype: int64

Frequncy distribution for Specialization:
Specialization
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: count, dtype: int64

Frequncy distribution for CollegeState:
CollegeState
Uttar Pradesh        915
Karnataka            370
Tamil Nadu           367
Telangana            319
```

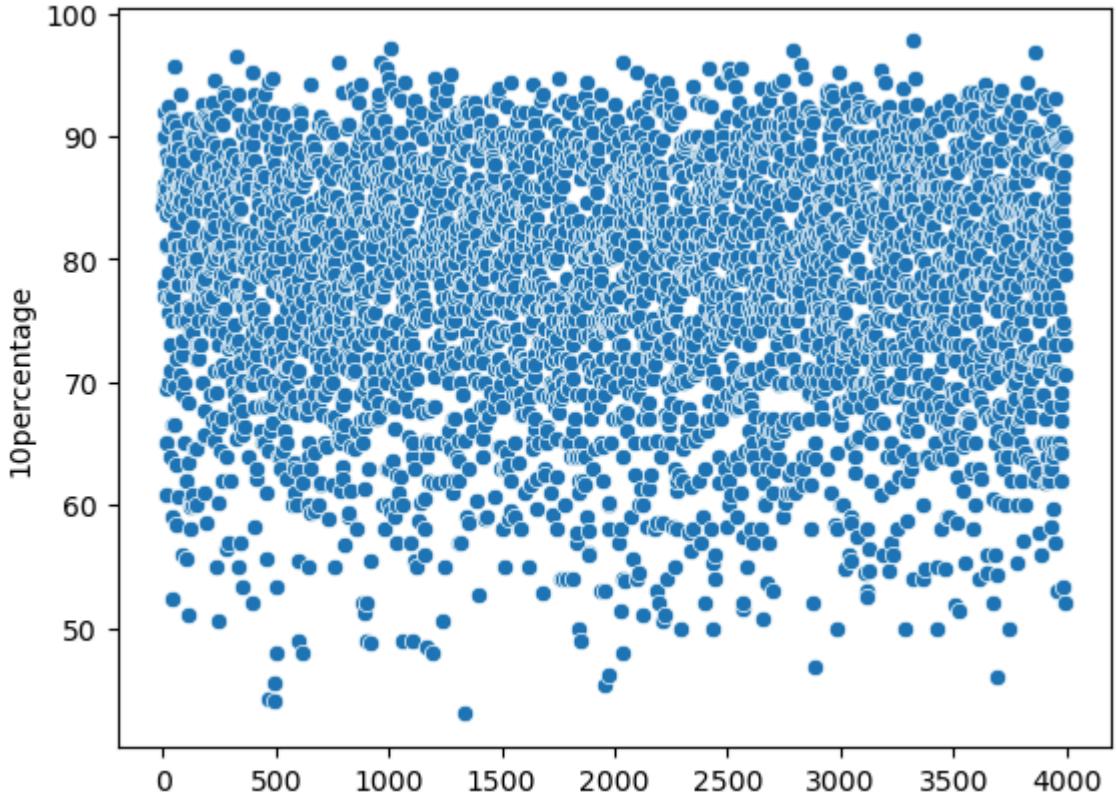
```
Maharashtra          262
Andhra Pradesh       225
West Bengal          196
Punjab               193
Madhya Pradesh       189
Haryana              180
Rajasthan            174
Orissa               172
Delhi                162
Uttarakhand          113
Kerala                33
Jharkhand             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: count, dtype: int64
```

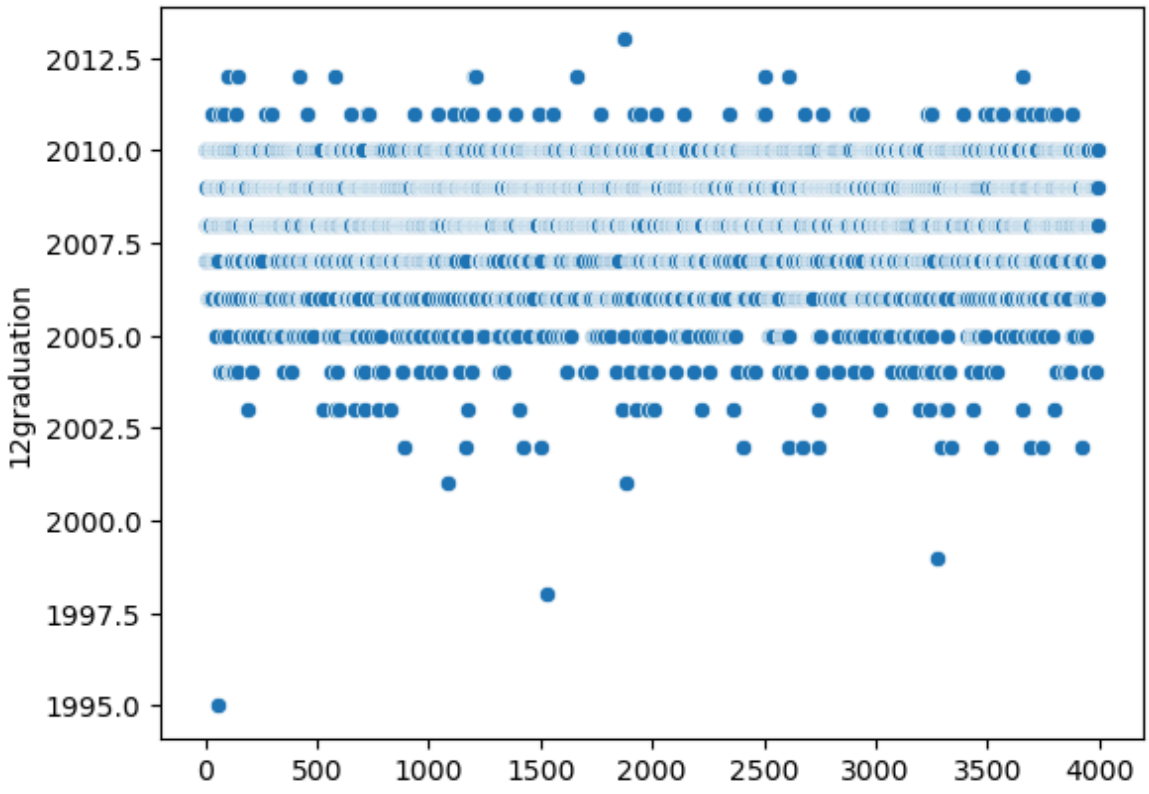
Bi-Variate Analysis

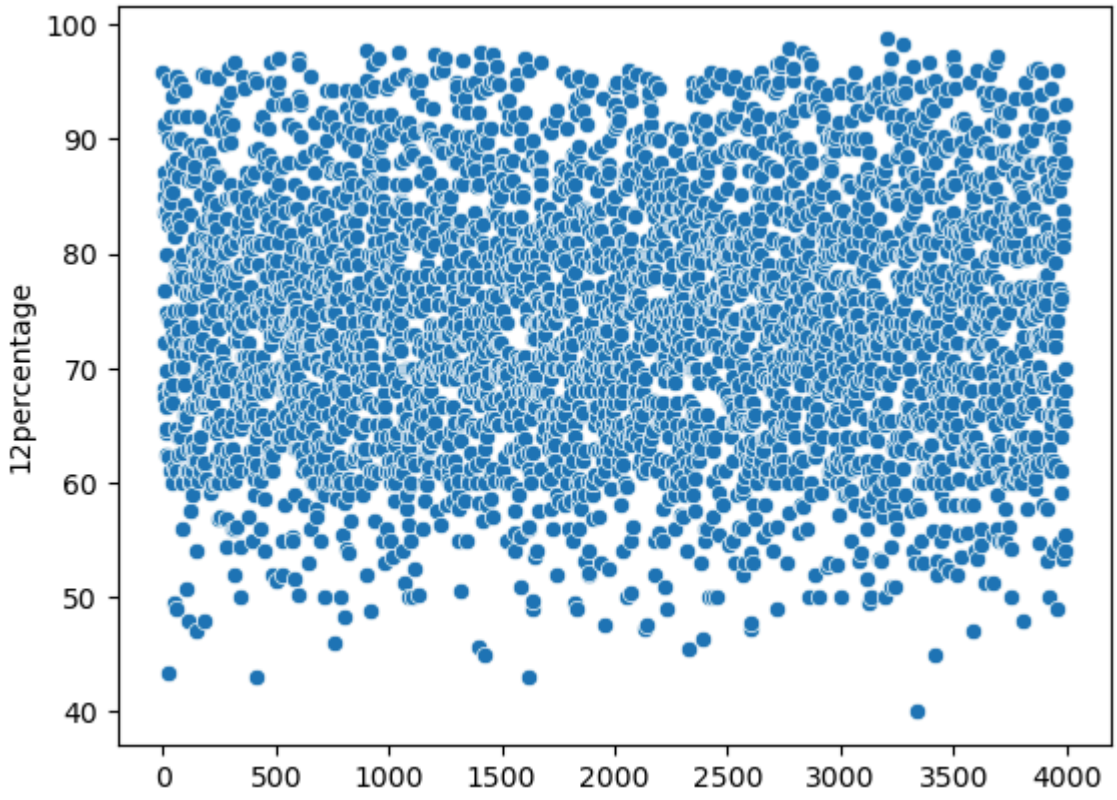
Relationships between numerical columns using Scatter plots

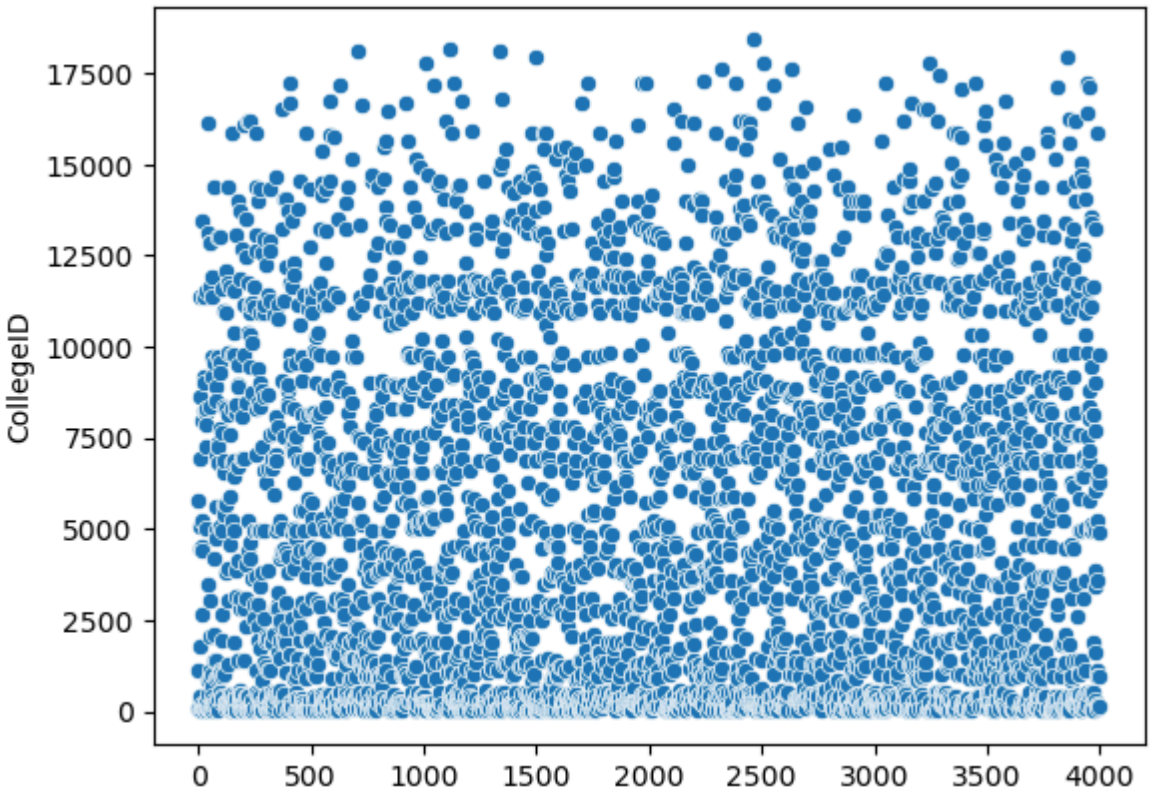
In [26]:
```
for i in num_cols[1:]:
    sns.scatterplot(df[i])
    plt.show()
```

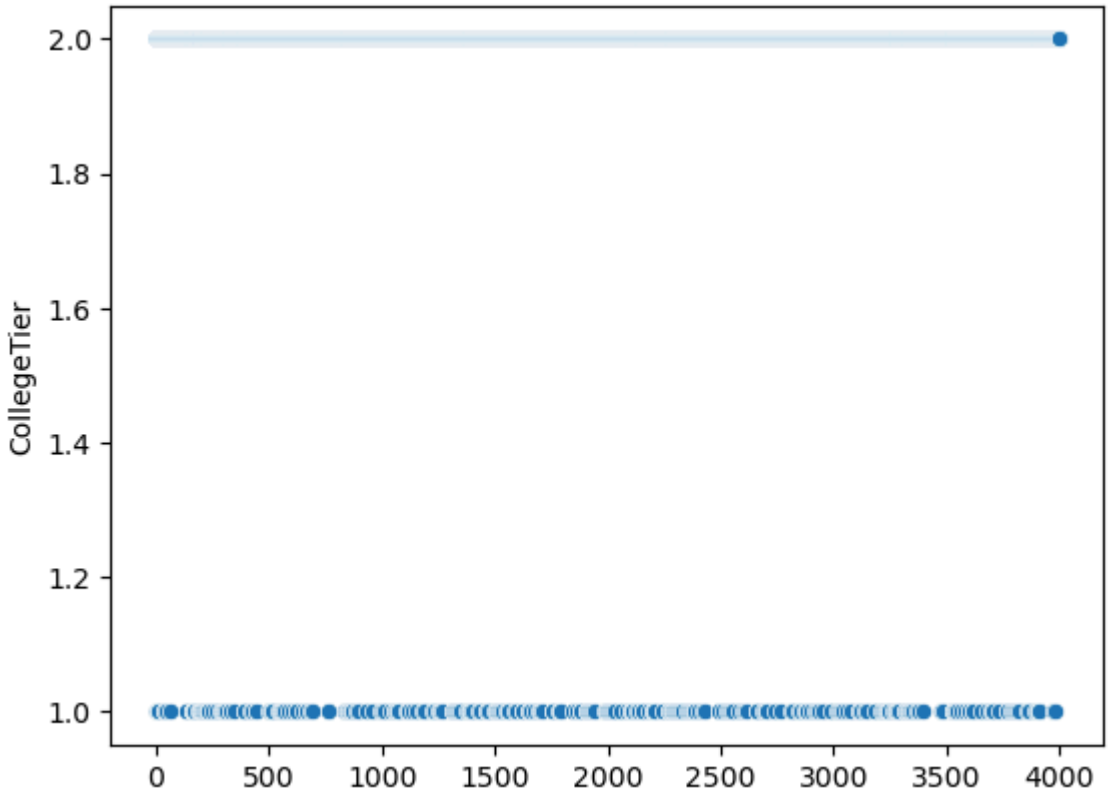


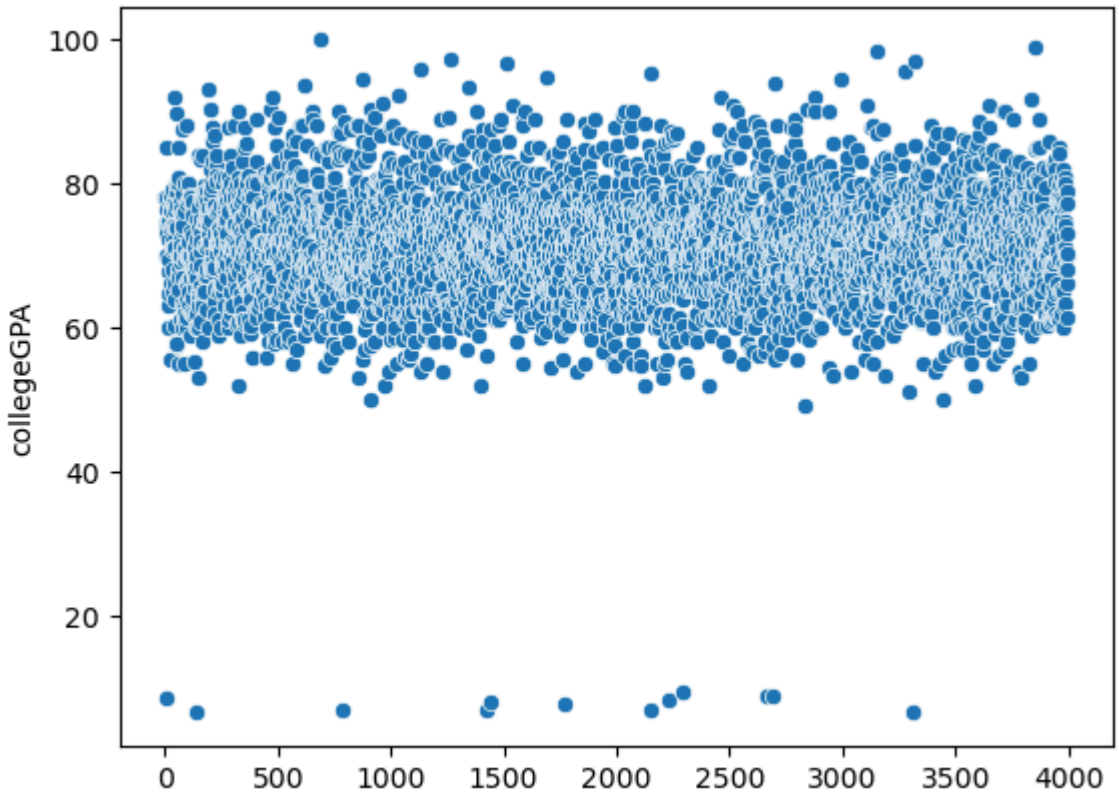

10percentage
100
90
80
70
60
50
0
500
1000
1500
2000
2500
3000
3500
4000

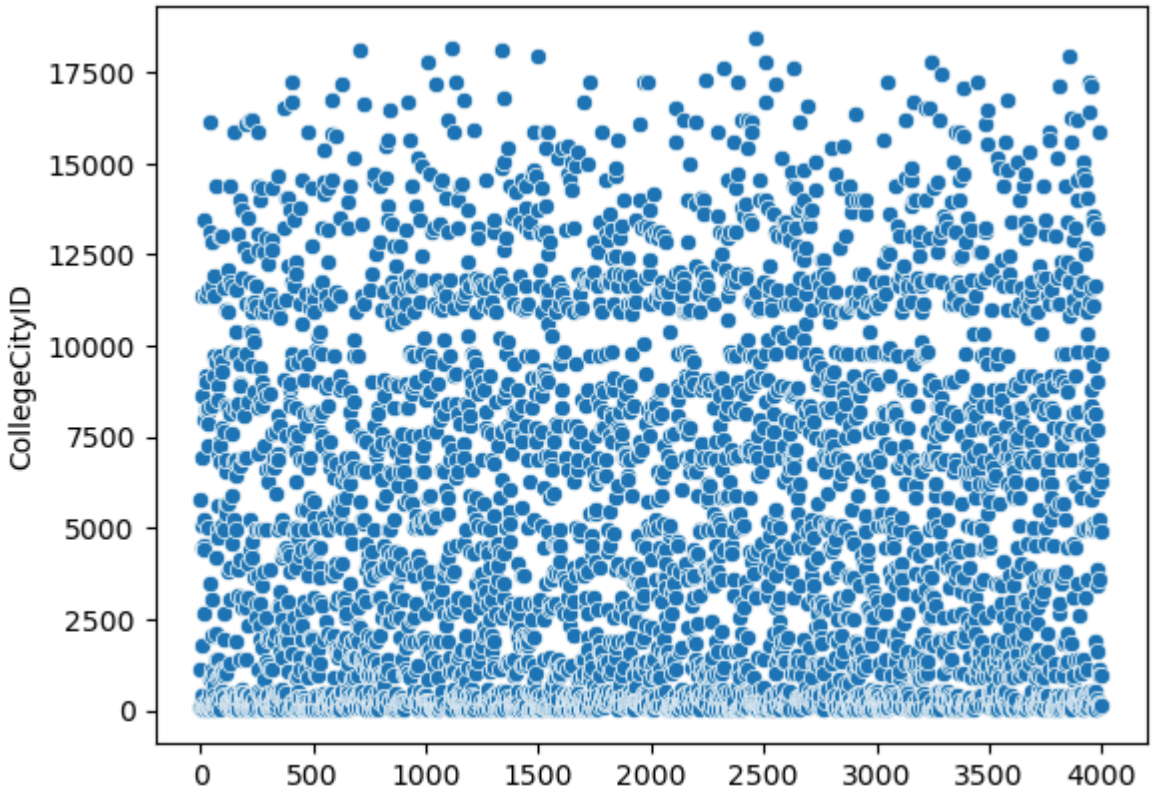

12graduation
2012.5
2010.0
2007.5
2005.0
2002.5
2000.0
1997.5
1995.0
0
500
1000
1500
2000
2500
3000
3500
4000

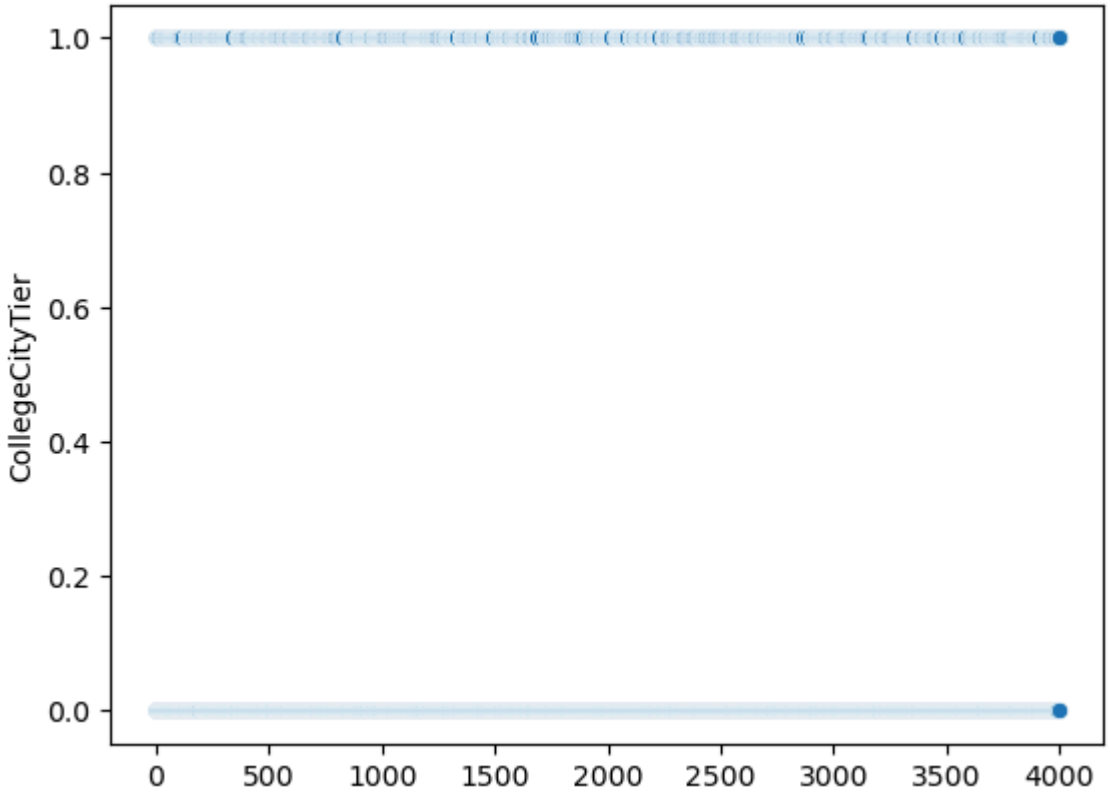

12percentage
100
90
80
70
60
50
40
0
500
1000
1500
2000
2500
3000
3500
4000

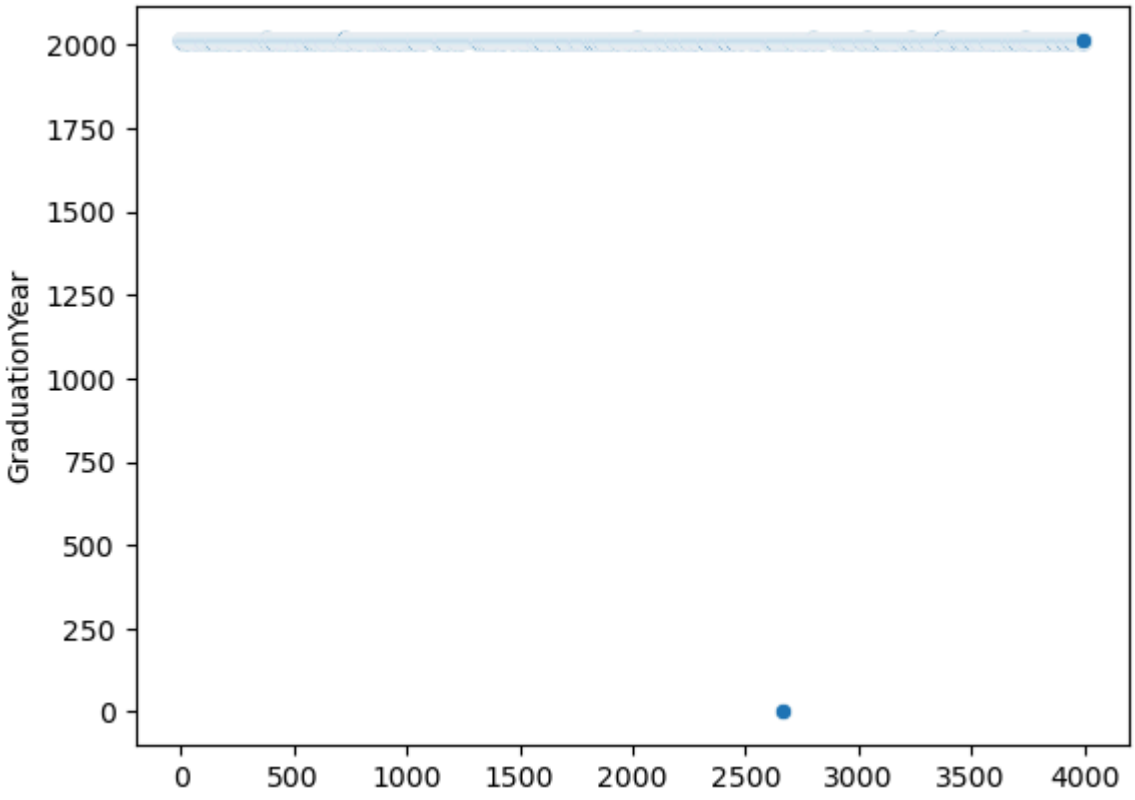

CollegeID
17500
15000
12500
10000
7500
5000
2500
0
0
500
1000
1500
2000
2500
3000
3500
4000

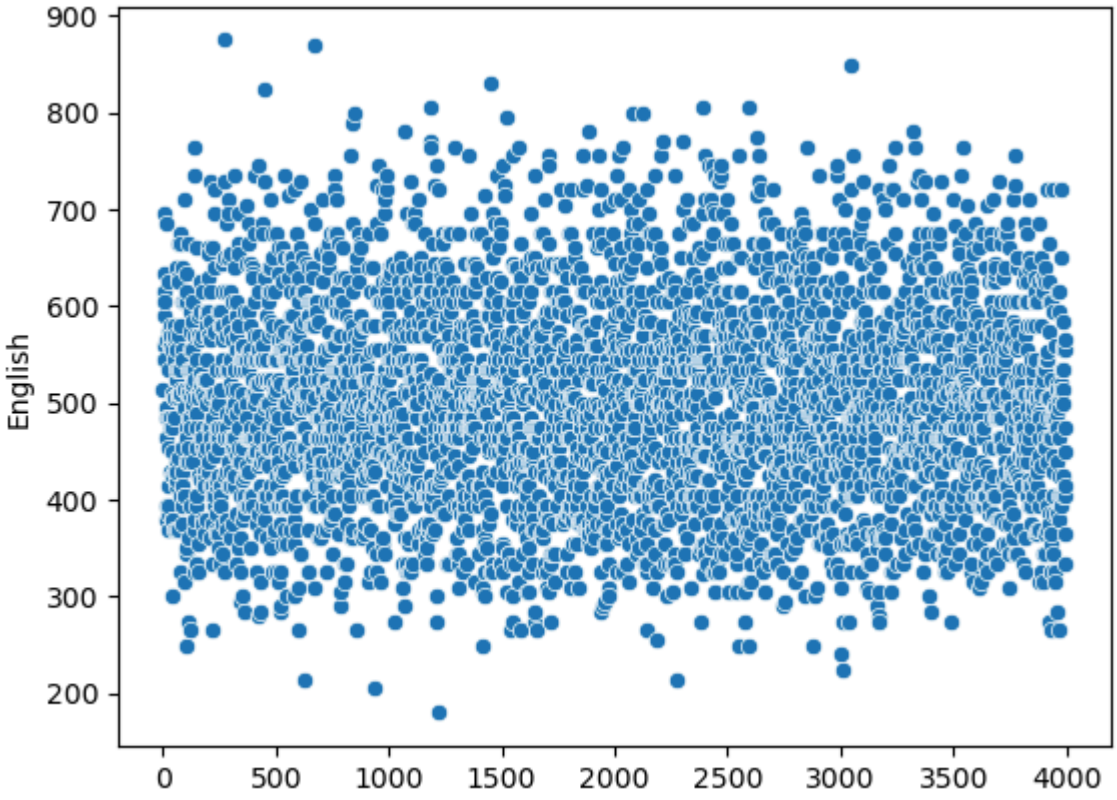

CollegeTier
2.0
1.8
1.6
1.4
1.2
1.0
0
500
1000
1500
2000
2500
3000
3500
4000

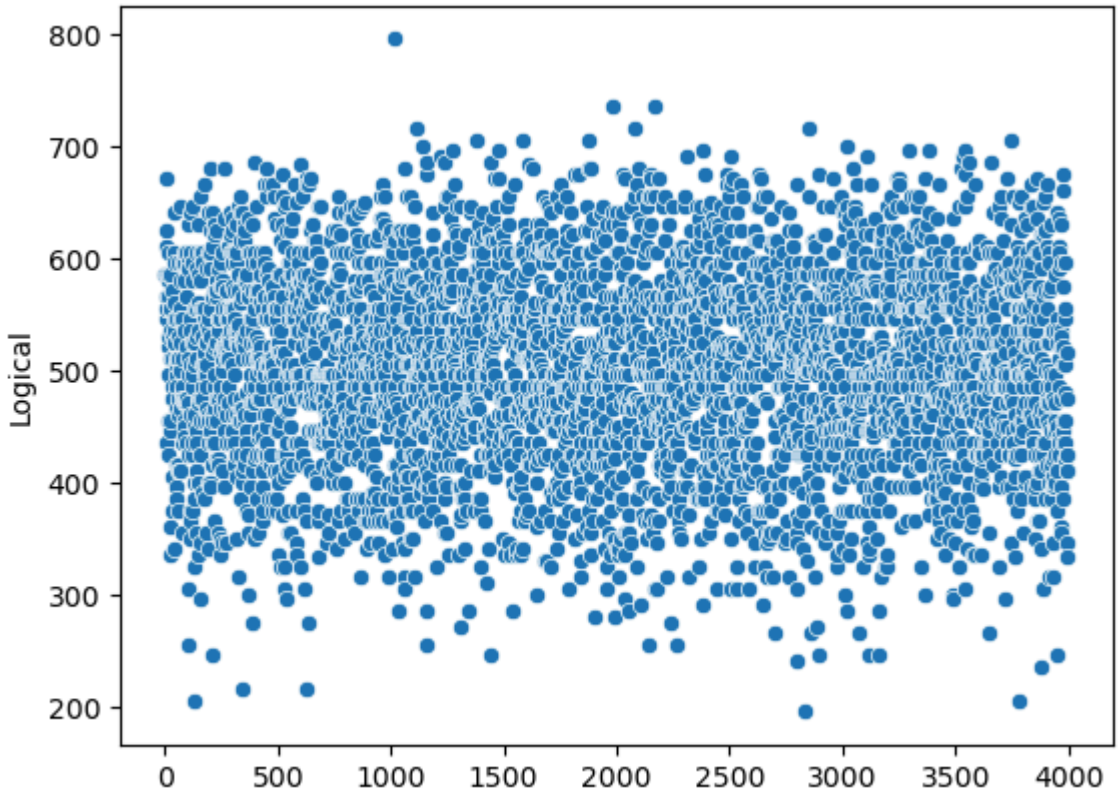

collegeGPA
100
80
60
40
20
0
500
1000
1500
2000
2500
3000
3500
4000

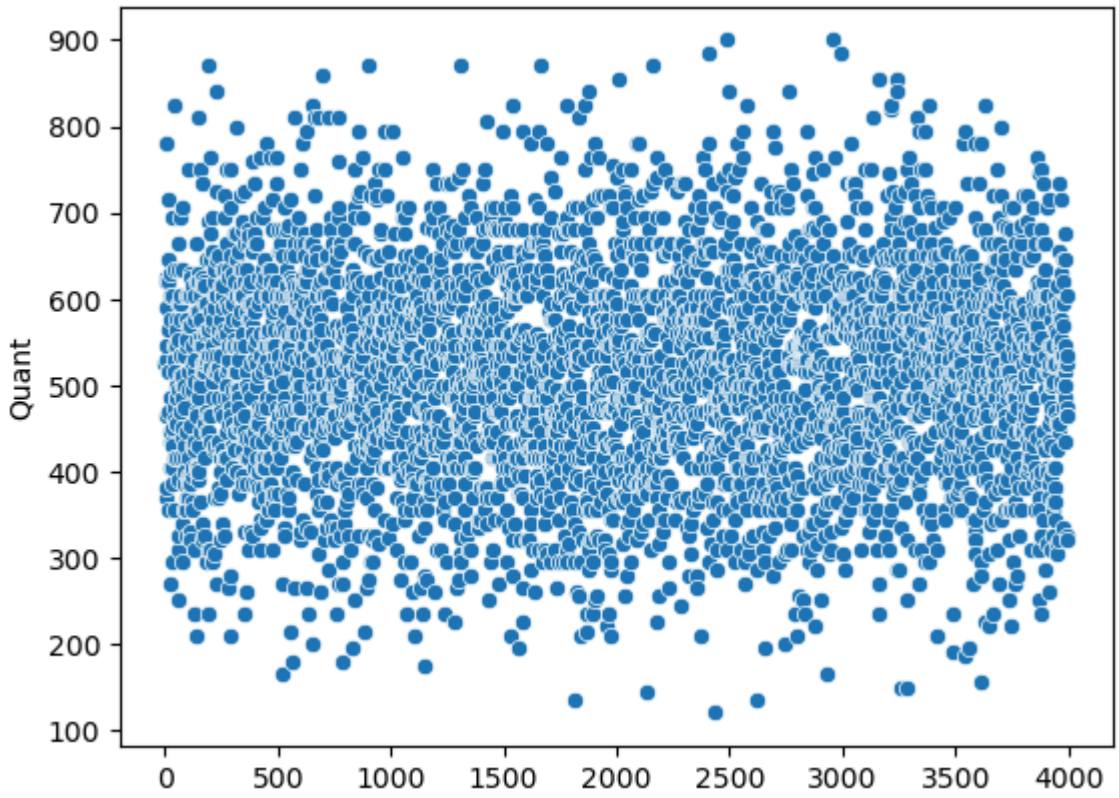

CollegeCityID
17500
15000
12500
10000
7500
5000
2500
0
0
500
1000
1500
2000
2500
3000
3500
4000

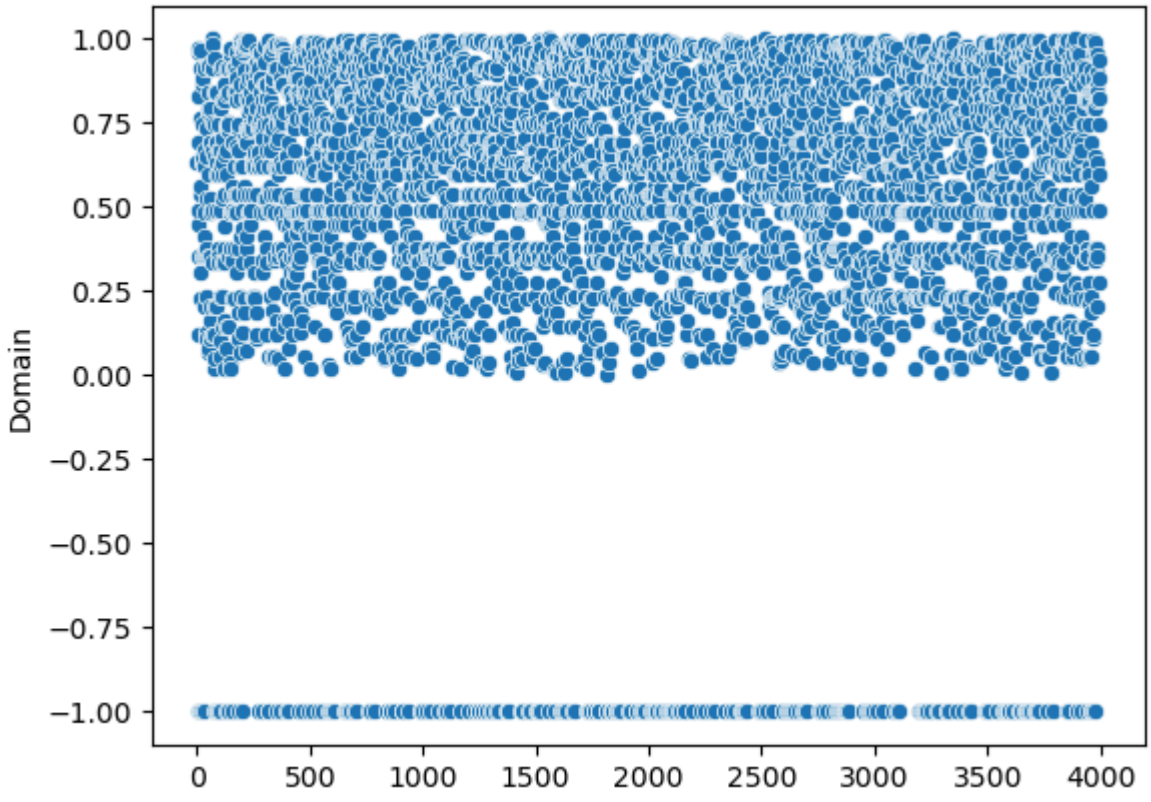

CollegeCityTier
1.0
0.8
0.6
0.4
0.2
0.0
0
500
1000
1500
2000
2500
3000
3500
4000

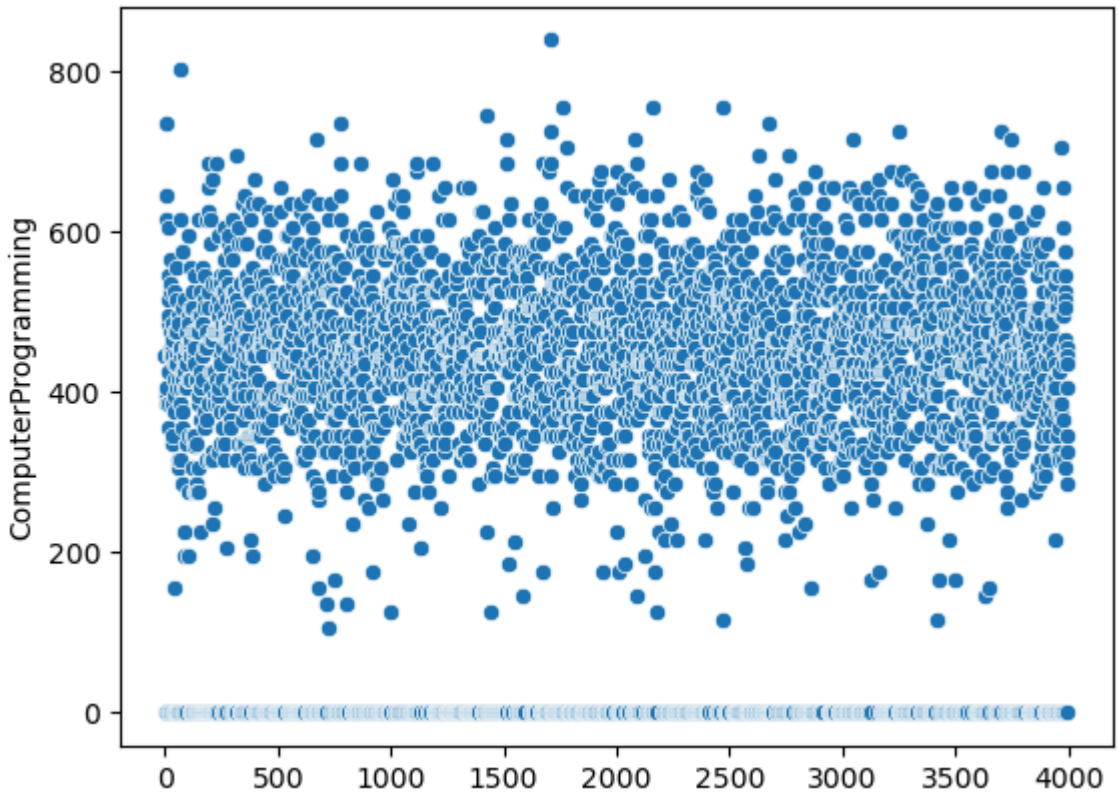

GraduationYear
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500
3000
3500
4000

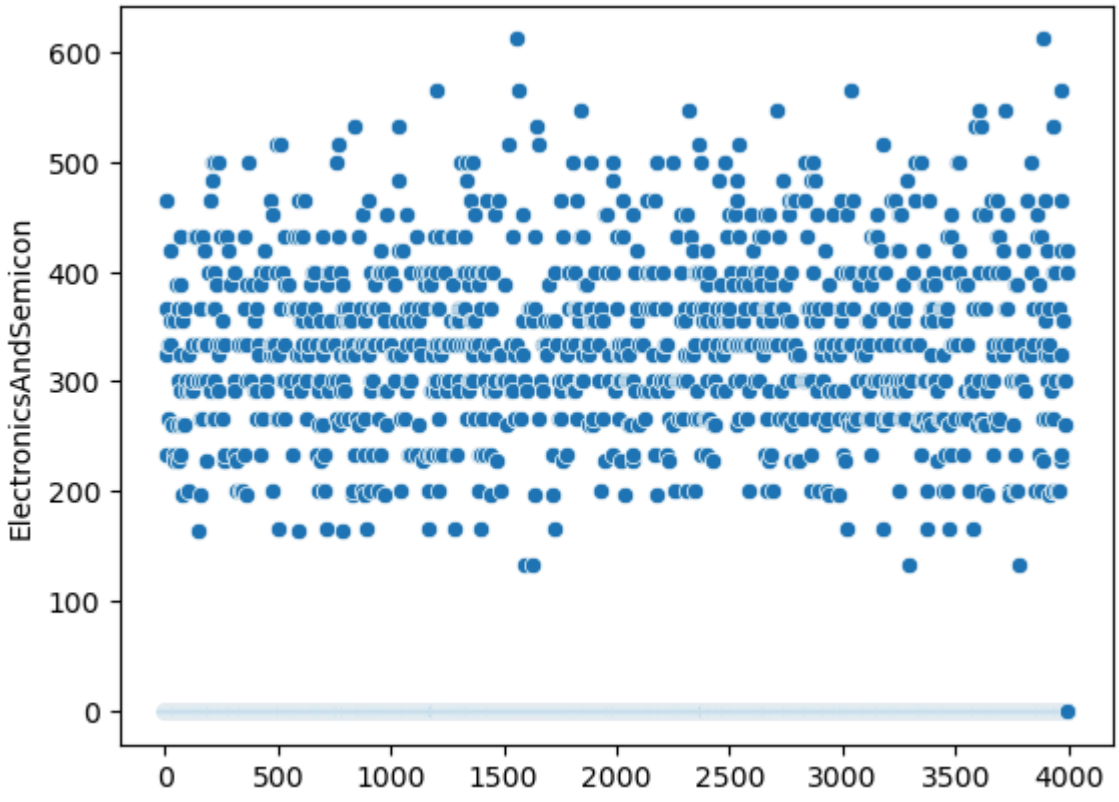

English
900
800
700
600
500
400
300
200
0
500
1000
1500
2000
2500
3000
3500
4000

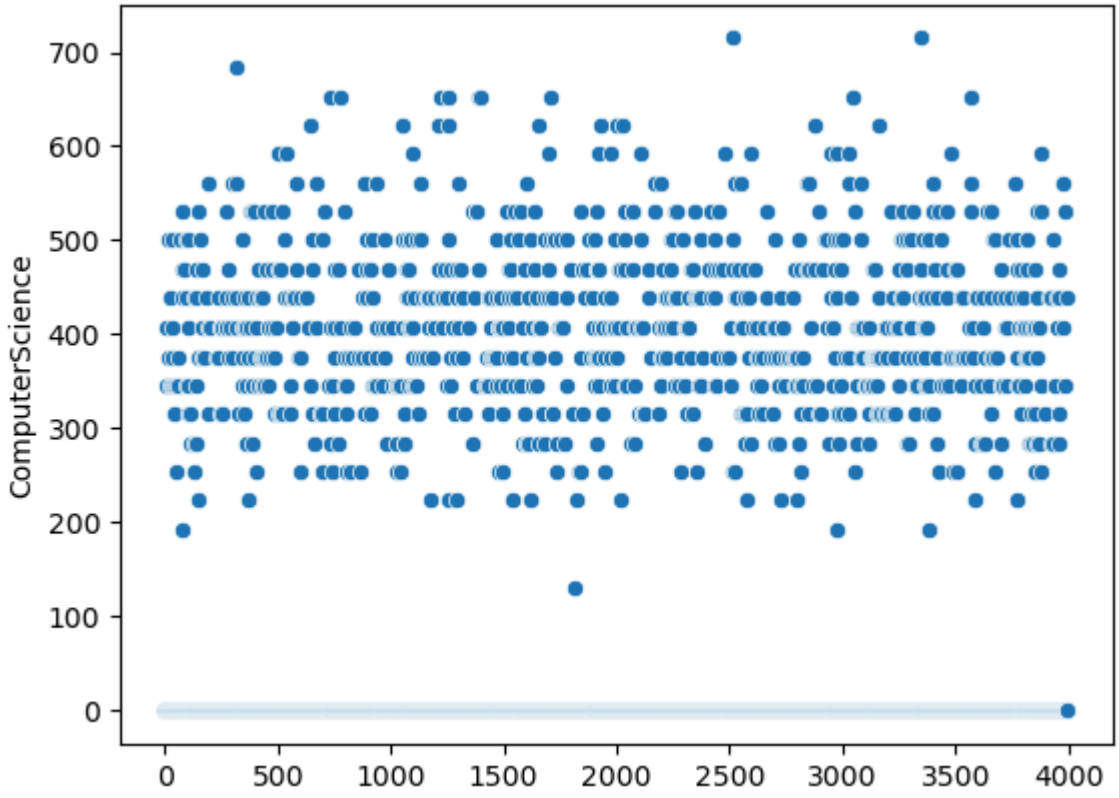

Logical
800
700
600
500
400
300
200
0
500
1000
1500
2000
2500
3000
3500
4000

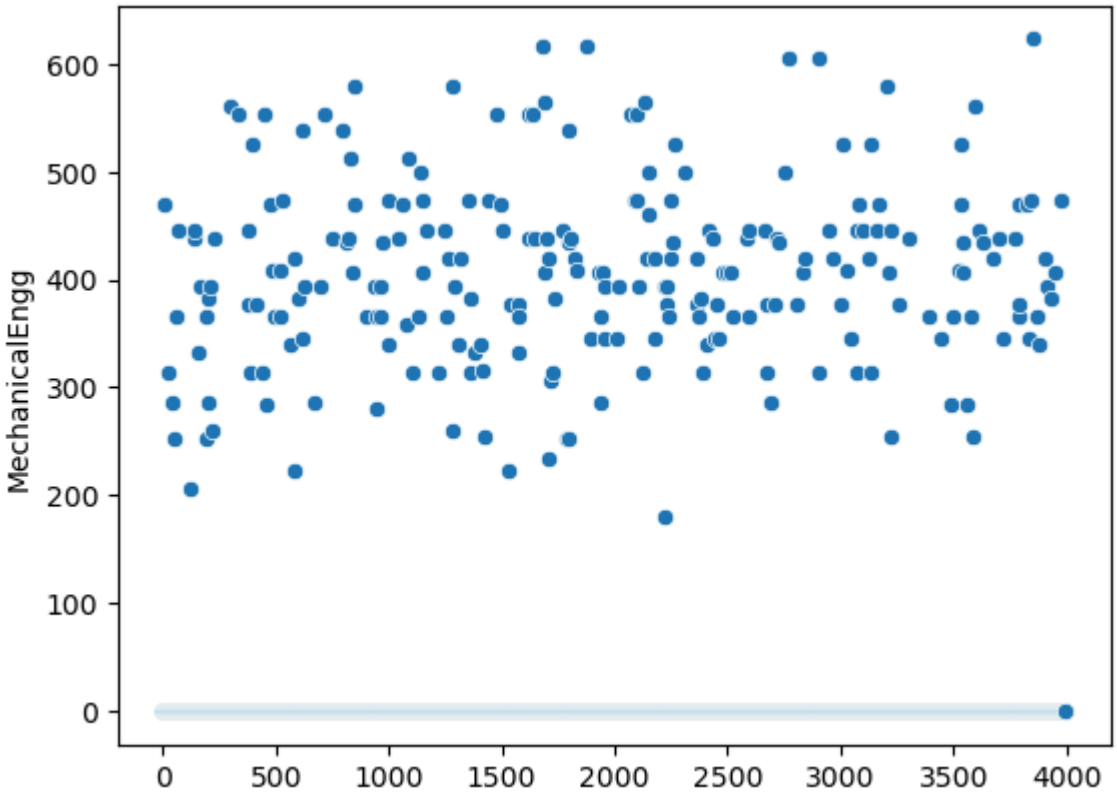

Quant
900
800
700
600
500
400
300
200
100
0
500
1000
1500
2000
2500
3000
3500
4000

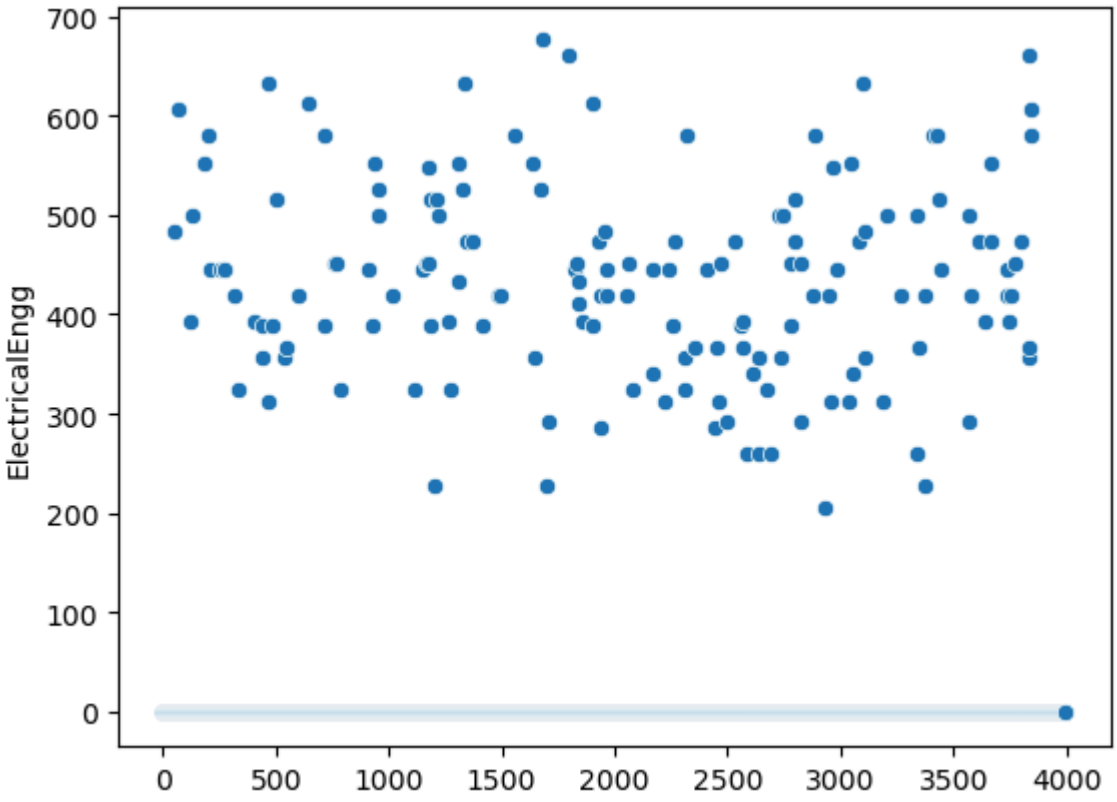

Domain
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
500
1000
1500
2000
2500
3000
3500
4000

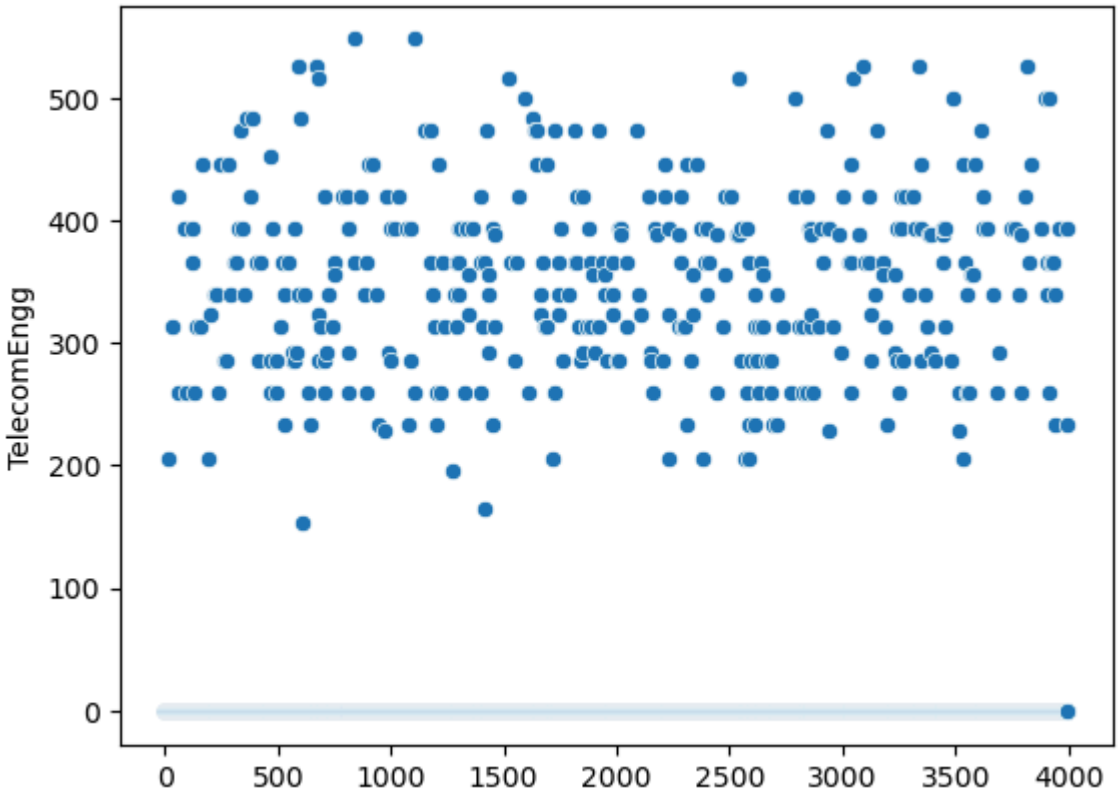

ComputerProgramming
800
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500
4000

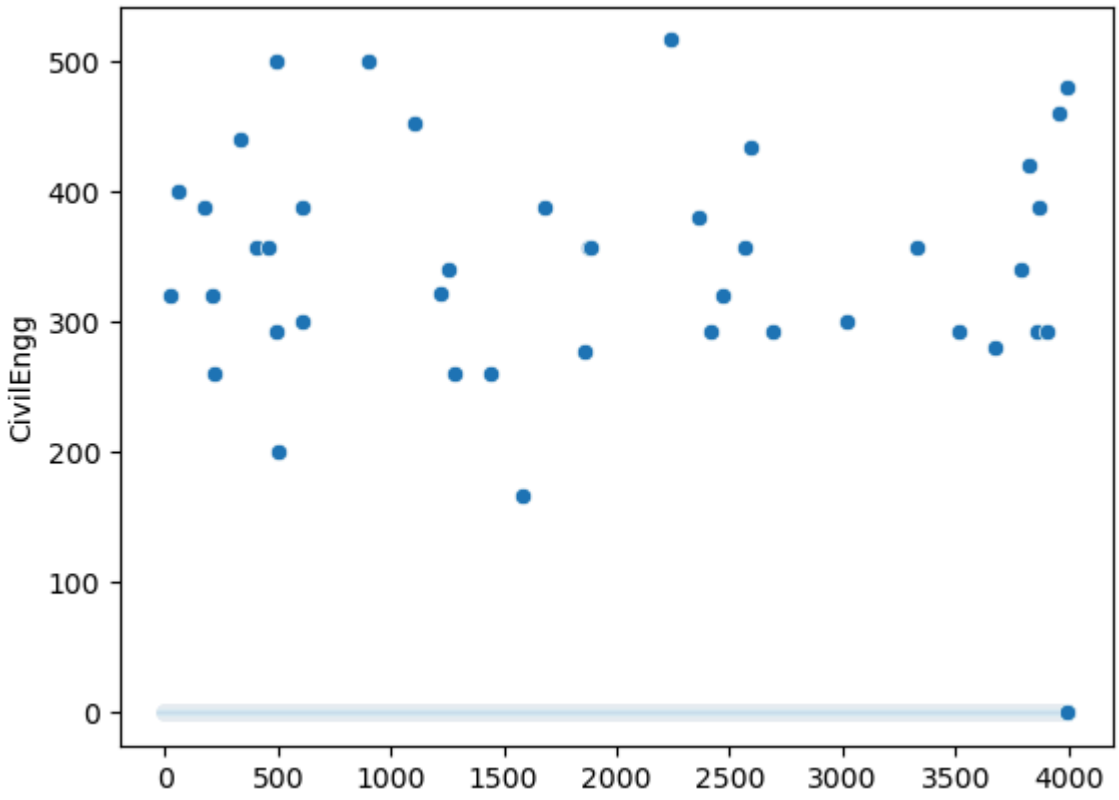

ElectronicsAndSemicon
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
4000

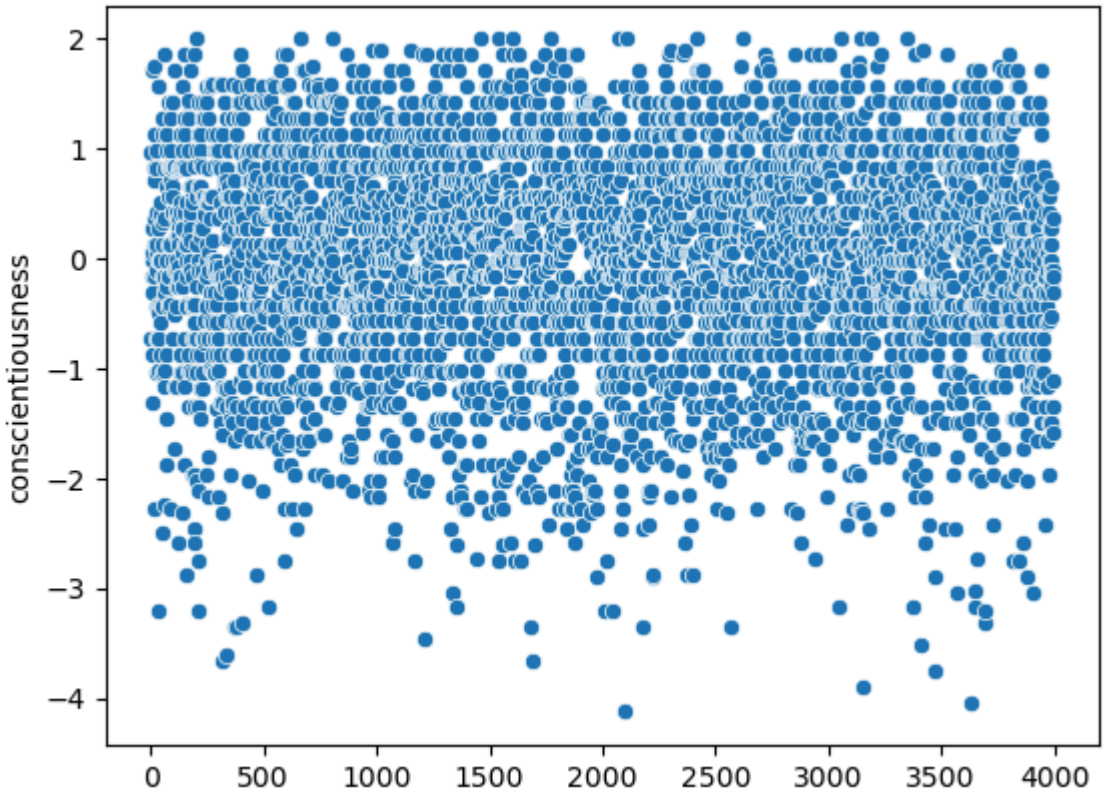

ComputerScience
700
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
4000

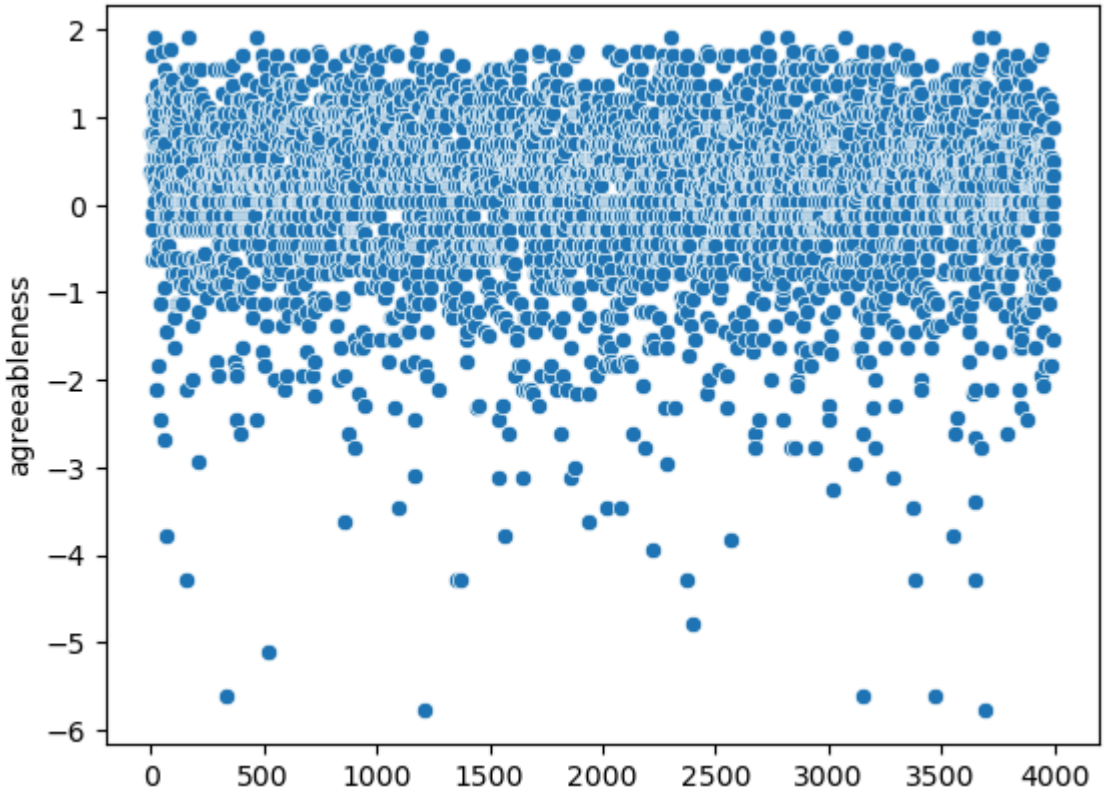

MechanicalEngg
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
4000

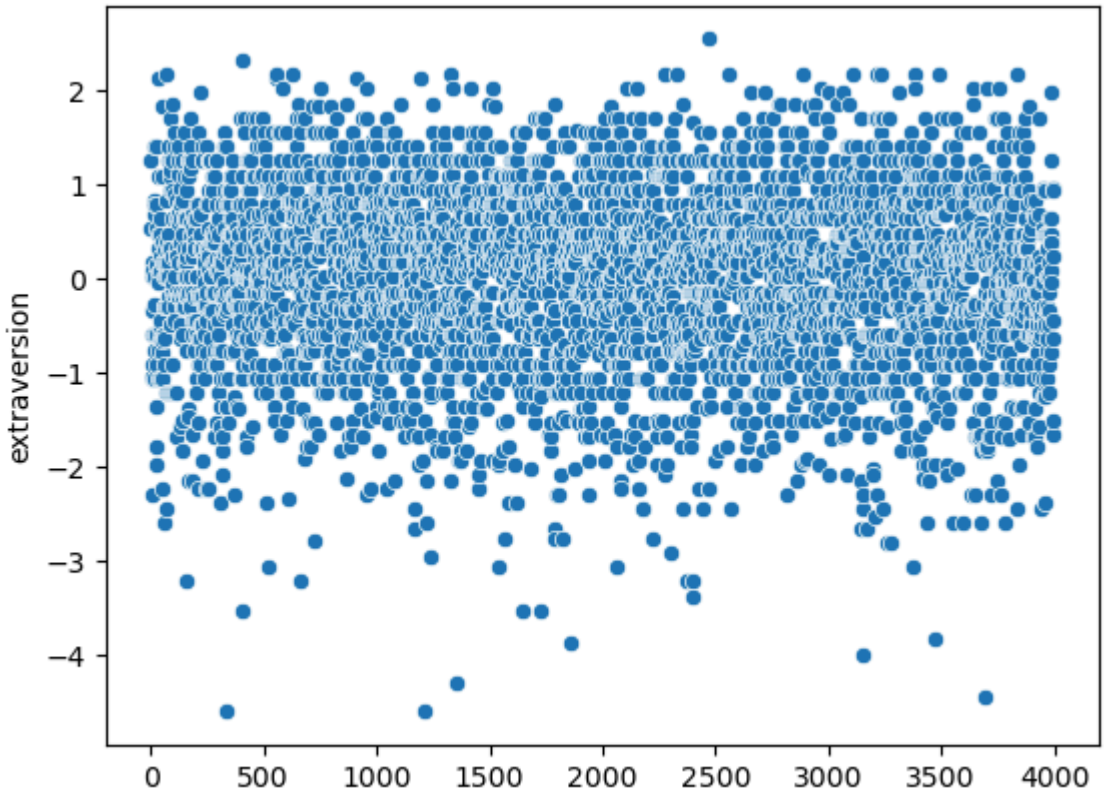

ElectricalEngg
700
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500
4000

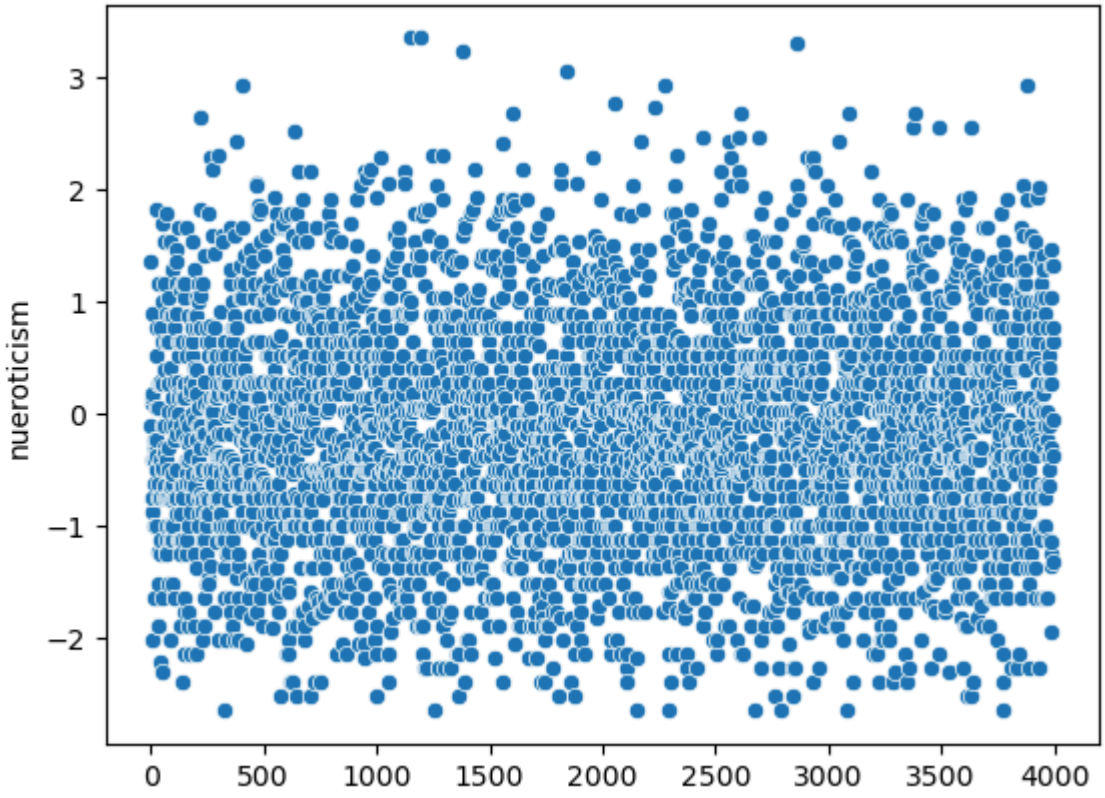

TelecomEngg
0
100
200
300
400
500
0
500
1000
1500
2000
2500
3000
3500
4000

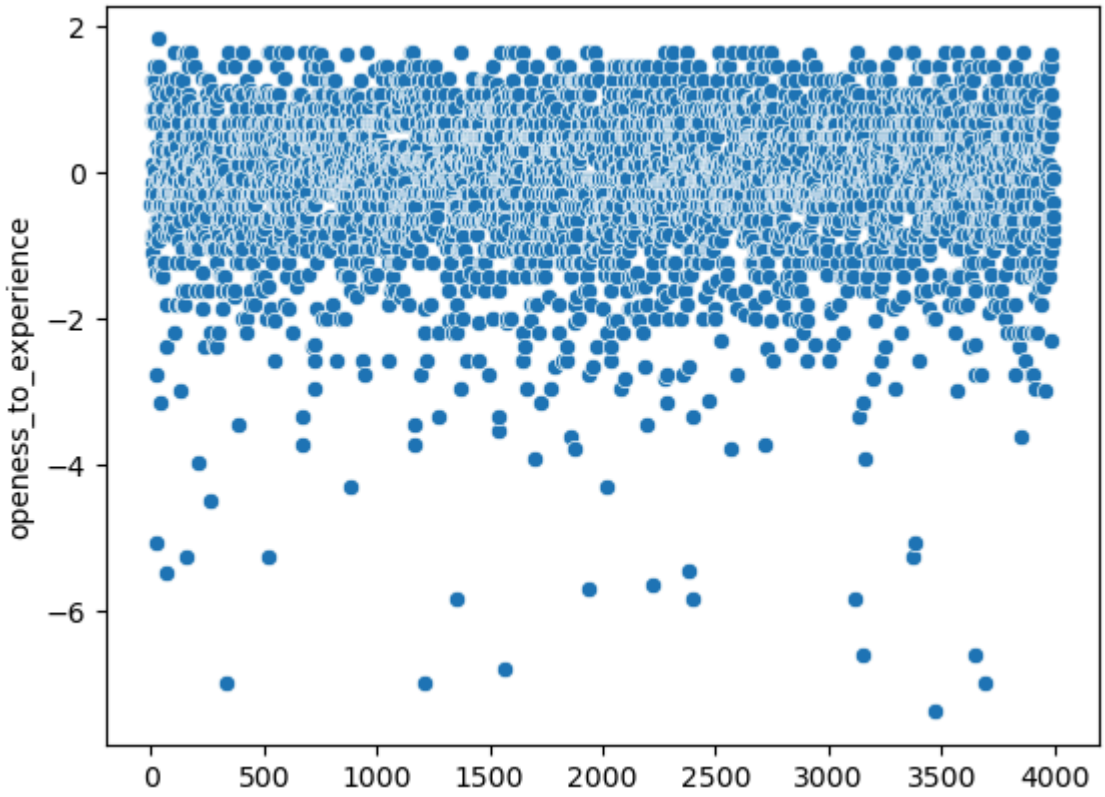

CivilEngg
0
100
200
300
400
500
0
500
1000
1500
2000
2500
3000
3500
4000


conscientiousness
2
1
0
−1
−2
−3
−4
0
500
1000
1500
2000
2500
3000
3500
4000


agreeableness
2
1
0
−1
−2
−3
−4
−5
−6
0
500
1000
1500
2000
2500
3000
3500
4000


extraversion
2
1
0
−1
−2
−3
−4
0
500
1000
1500
2000
2500
3000
3500
4000


nueroticism
3
2
1
0
−1
−2
0
500
1000
1500
2000
2500
3000
3500
4000



In [38]:
```
df['Degree'].unique()
```

Out[38]:
```
array(['B.Tech/B.E.', 'MCA', 'M.Tech./M.E.', 'M.Sc. (Tech.)'],
      dtype=object)
```

In [39]:
```
df['Degree'].nunique()
```

Out[39]:
```
4
```

In [41]:
```
df['JobCity'].value_counts()
```

Out[41]:
```
JobCity
Bangalore           627
-1                  461
Noida               368
Hyderabad           335
Pune                290
                   ...
Tirunelvelli          1
Ernakulam             1
Nanded                1
Dharmapuri            1
Asifabadbanglore      1
Name: count, Length: 339, dtype: int64
```

In [43]:
```
df['Gender'].value_counts()
```

Out[43]:
```
Gender
m    3041
f     957
Name: count, dtype: int64
```

In [44]:
```
for i in df['Gender'].unique():
    con=df['Gender']==i
```

```
    count=len(df[con])
    print(f"The number of employees according to gender {i} is: {count}")
```

```
The number of employees according to gender f is: 957
The number of employees according to gender m is: 3041
```

In [45]:
```
df['10board'].value_counts()
```

Out[45]:
```
10board
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: count, Length: 275, dtype: int64
```

there a relationship between gender and specialization?

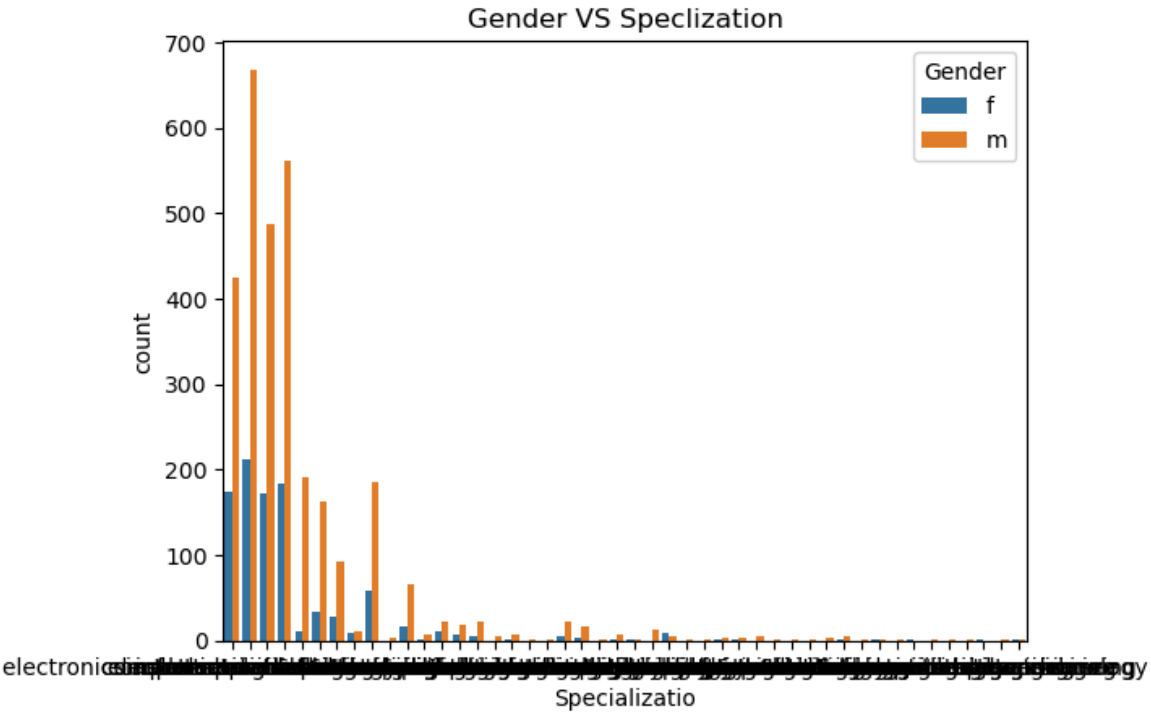
In [49]:
```
cat_cols
```

Out[49]:
```
Index(['Unnamed: 0', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10board', '12board', 'Degree', 'Specialization', 'CollegeState'],
      dtype='object')
```

In [50]:
```
from scipy.stats import chi2_contingency

contingency_table=pd.crosstab(df['Gender'],df['Specialization'])
print('contingency_table:')
print(contingency_table)

chi2,p,dof,expected=chi2_contingency(contingency_table)
print("\nChi-squared Test:")
print(f"Chi-squared statistics :{chi2}")
print(f"P-value:{p}")

sns.countplot(df,x='Specialization',hue='Gender')
plt.title("Gender VS Speclization")
plt.xlabel("Specializatio")
plt.ylabel("count")
plt.legend(title='Gender')

if p<0.05:
    print("\nThere is a siginificant relationship between gender and Specializat
else:
    print("\nThere is no siginificant relationship between gender and Specializa
```

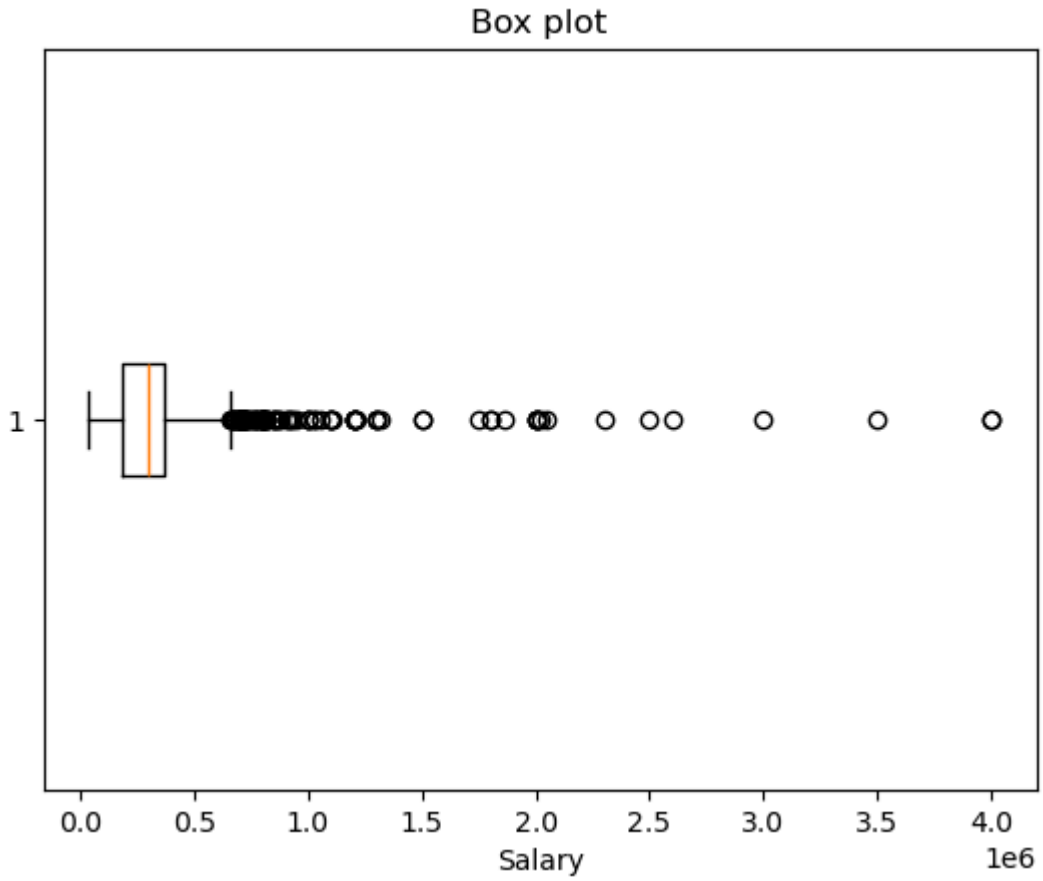
```
contingency_table:
Specialization  aeronautical engineering  \
Gender
f                                       1
m                                       2

Specialization  applied electronics and instrumentation  \
Gender
f                                                    2
m                                                    7

Specialization  automobile/automotive engineering  biomedical engineering  \
Gender
f                                                 0                      2
m                                                 5                      0

Specialization  biotechnology  ceramic engineering  chemical engineering  \
Gender
f                            9                    0                     1
m                            6                    1                     8

Specialization  civil engineering  computer and communication engineering  \
Gender
f                               6                                        0
m                              23                                        1

Specialization  computer application  ...  internal combustion engine  \
Gender                                 ...
f                                 59  ...                           0
m                                185  ...                           1

Specialization  mechanical & production engineering  \
Gender
f                                                  0
m                                                  1

Specialization  mechanical and automation  mechanical engineering  \
Gender
f                                       0                      10
m                                       5                     191

Specialization  mechatronics  metallurgical engineering  other  \
Gender
f                           1                          0      0
m                           3                          2     13

Specialization  polymer technology  power systems and automation  \
Gender
f                                 0                             0
m                                 1                             1

Specialization  telecommunication engineering
Gender
f                                           1
m                                           5

[2 rows x 46 columns]

Chi-squared Test:
Chi-squared statistics :104.46891913608455
```

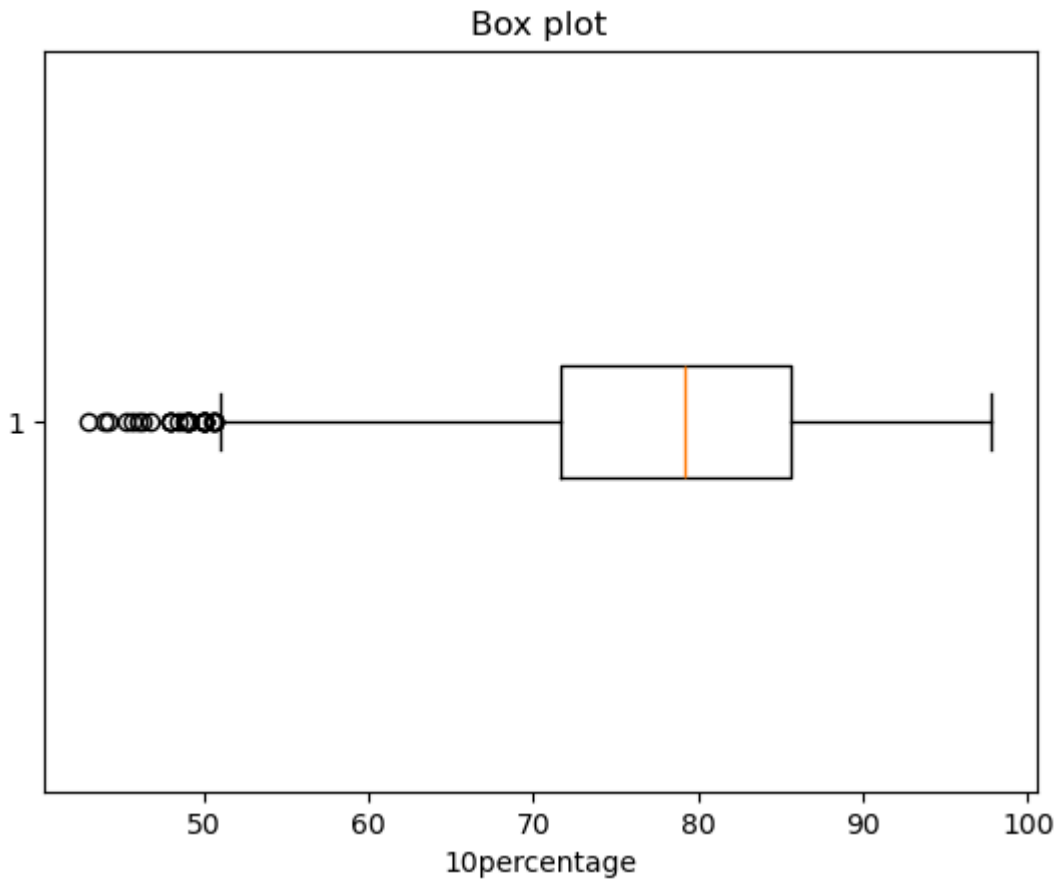
```
P-value:1.2453868176976918e-06

There is a siginificant relationship between gender and Specialization
```

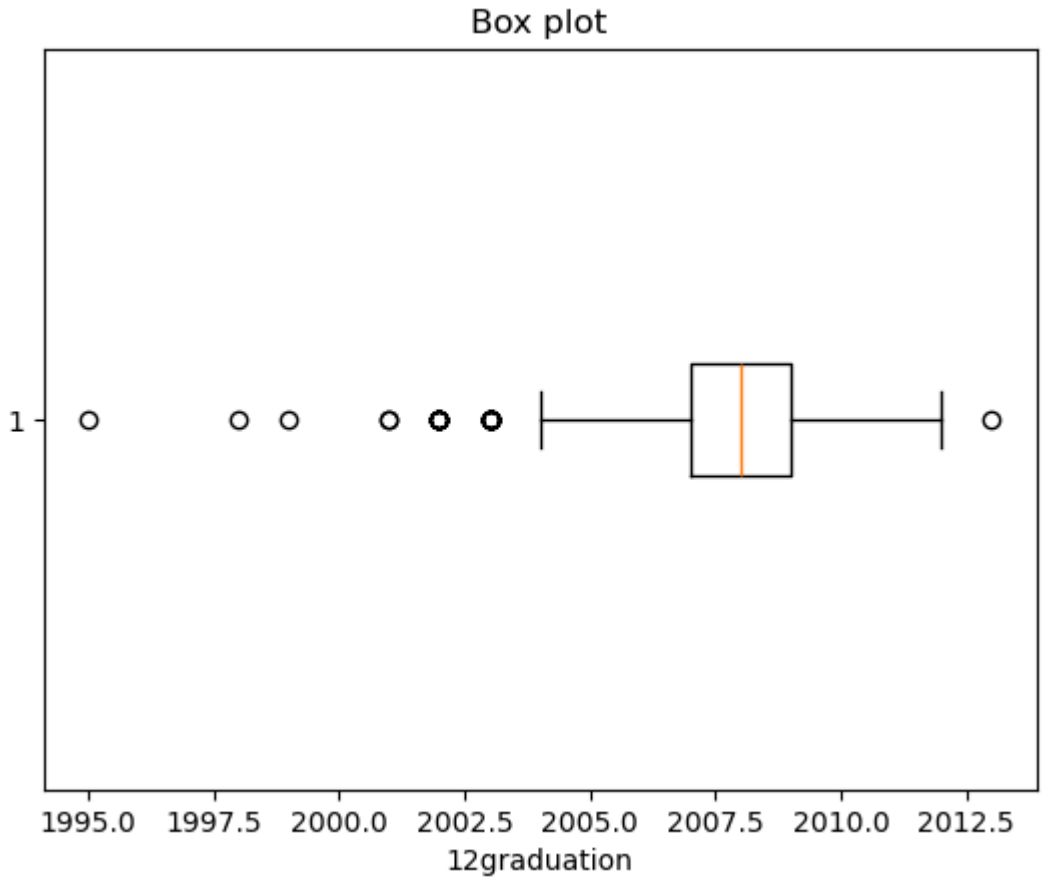


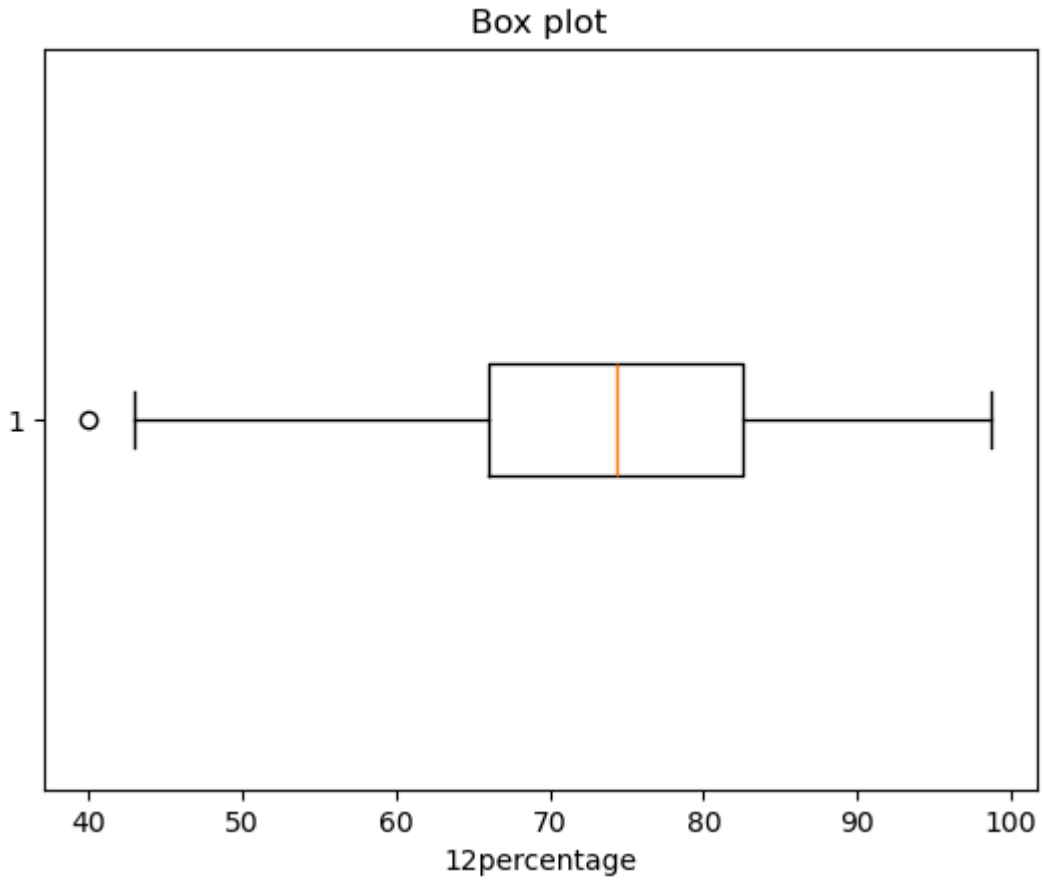
outlier analysis

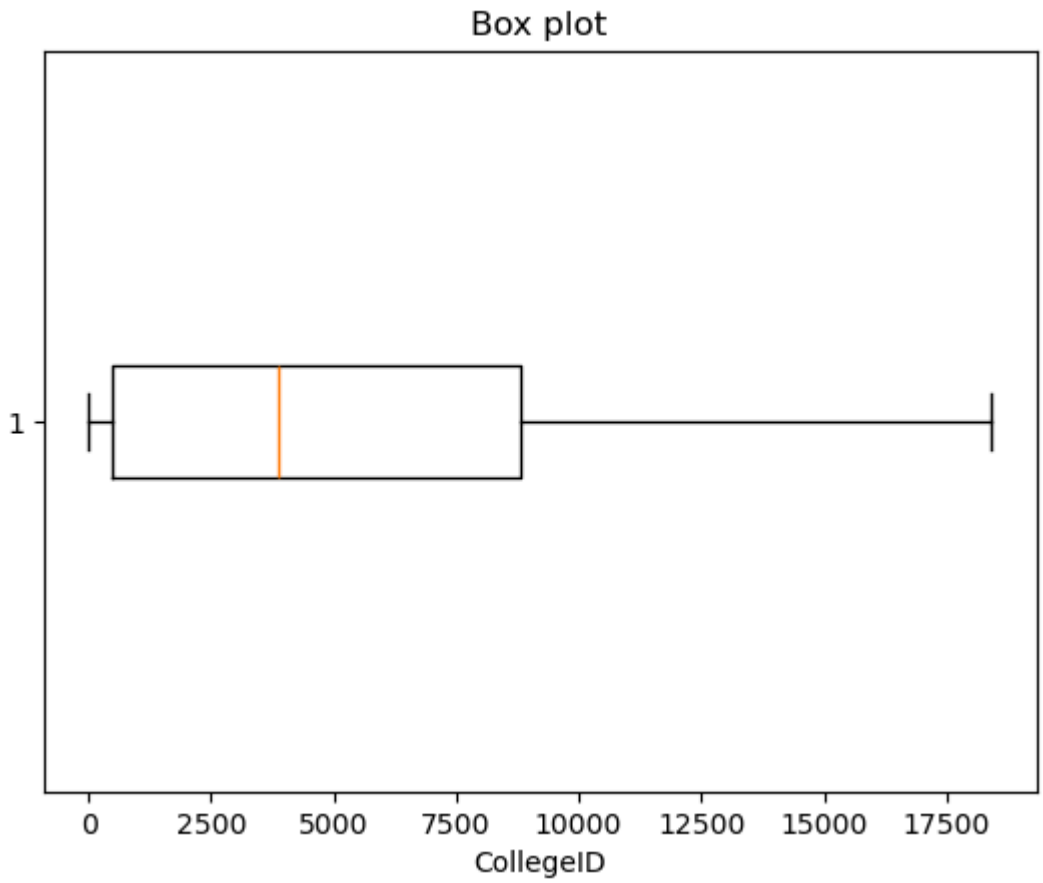
In [57]:
```
for i in num_cols[1:]:
    data=df[i]
    plt.boxplot(data,vert=False)
    plt.title("Box plot")
    plt.xlabel(i)
    plt.show()
```

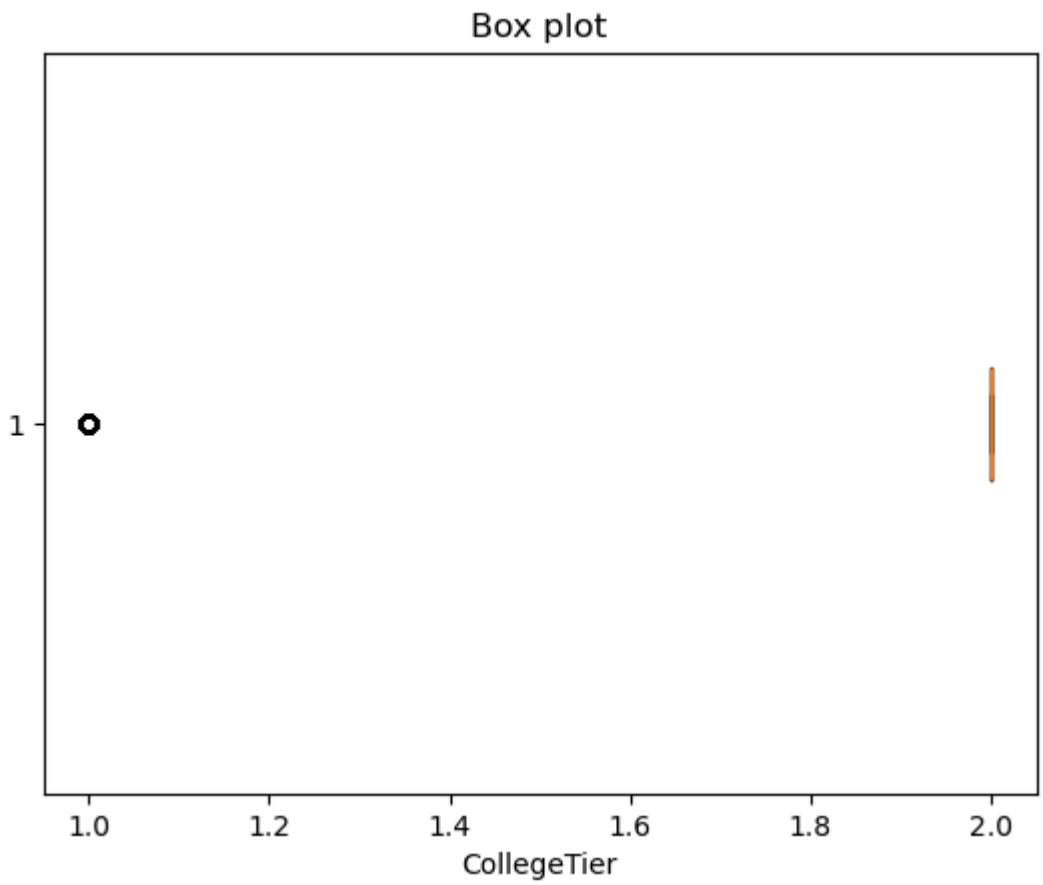

Box plot
1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Salary
1e6

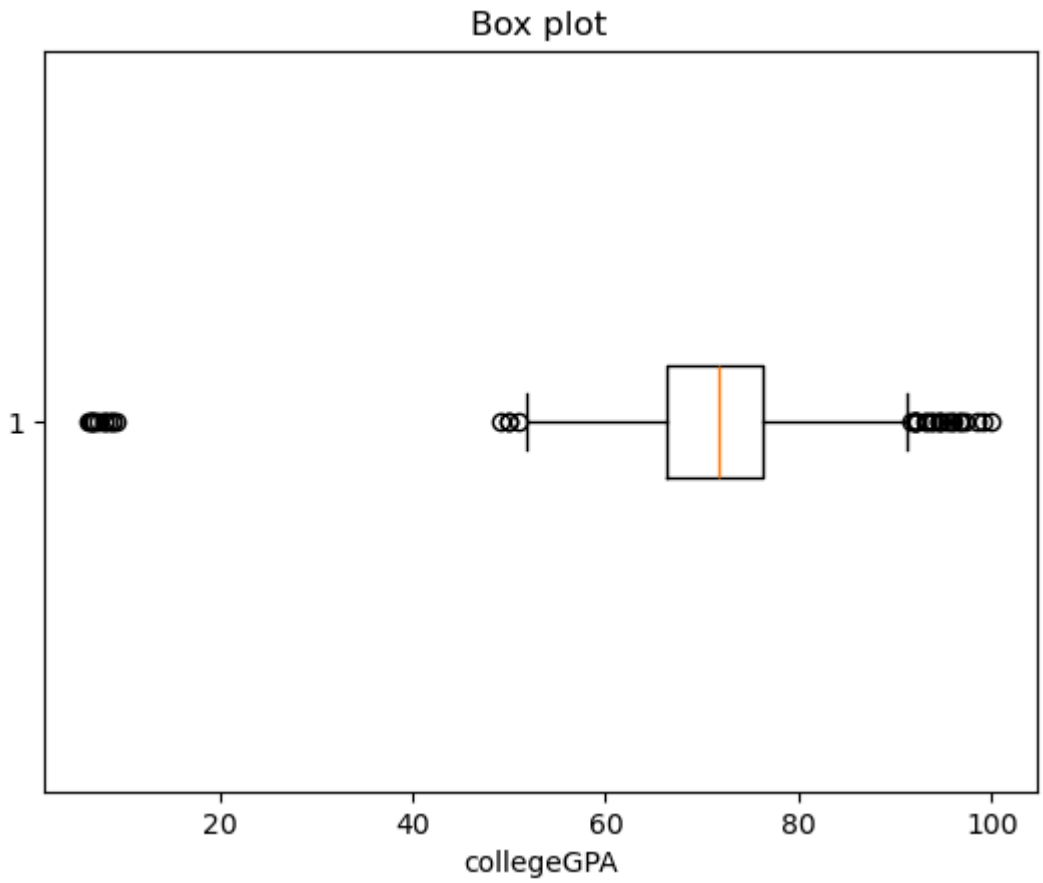

Box plot
1
50
60
70
80
90
100
10percentage

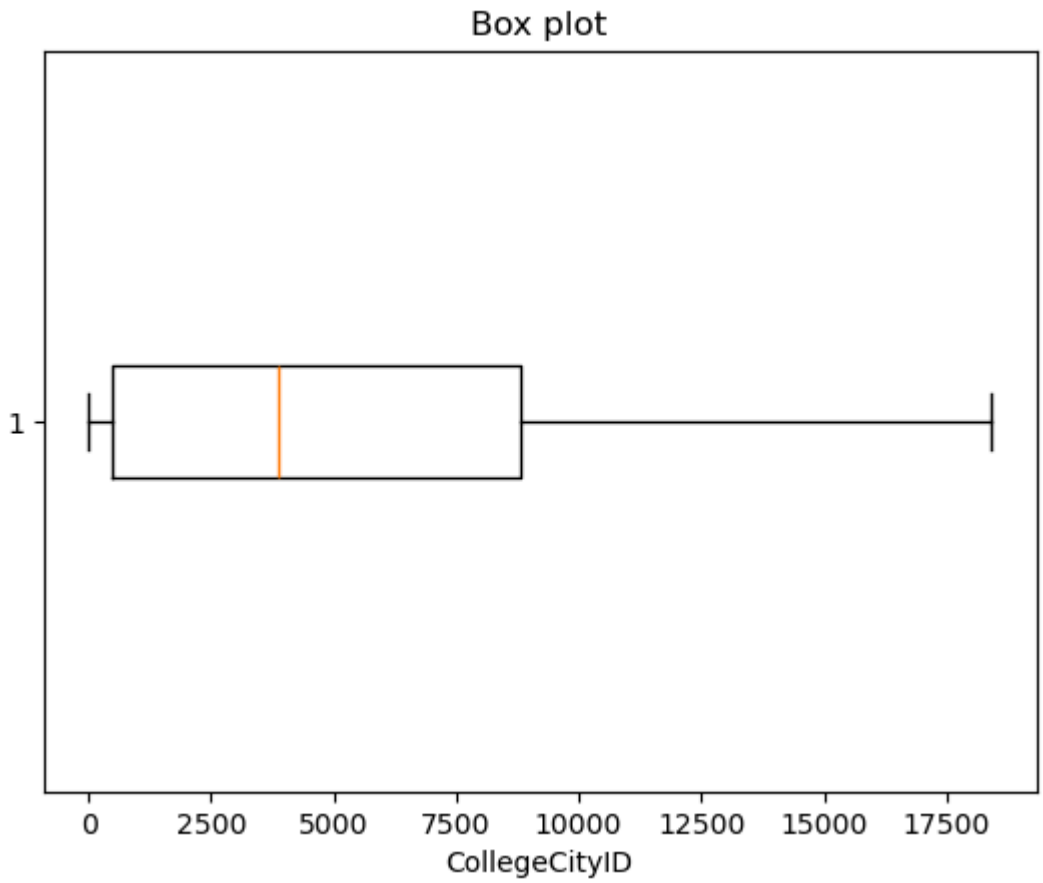

Box plot
1
1995.0
1997.5
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
12graduation

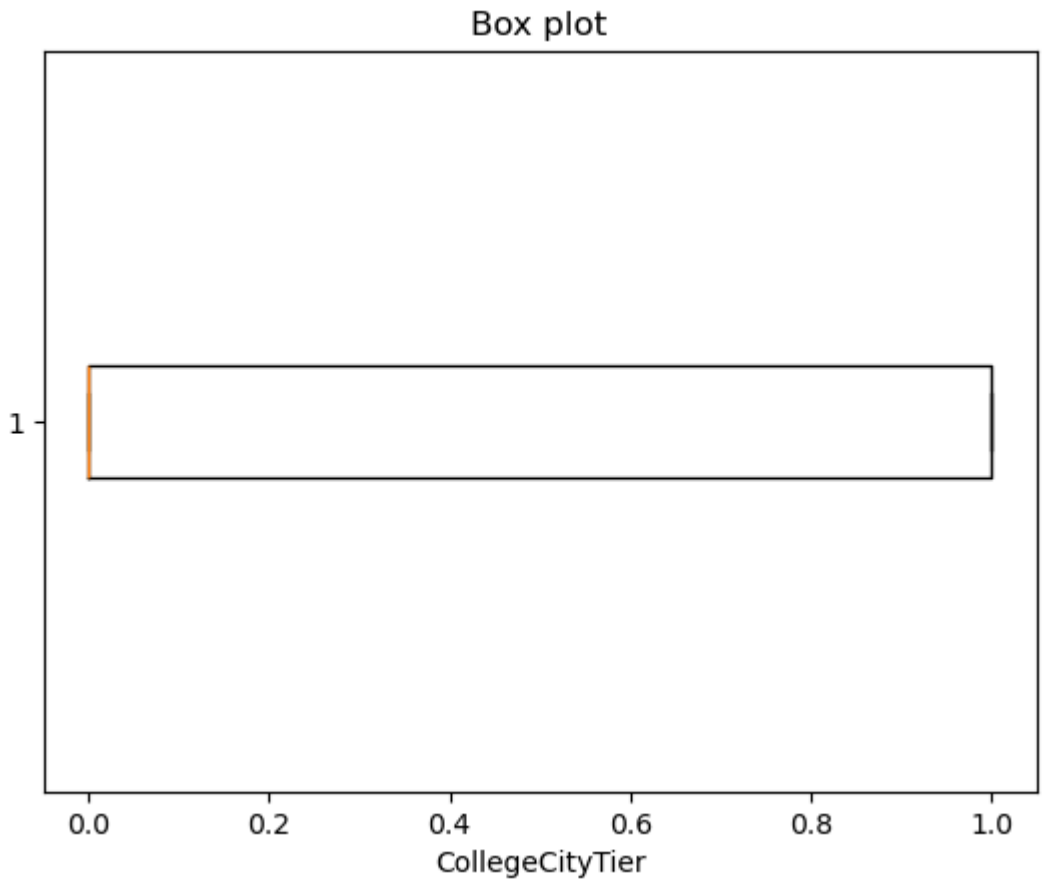

Box plot
1
40
50
60
70
80
90
100
12percentage

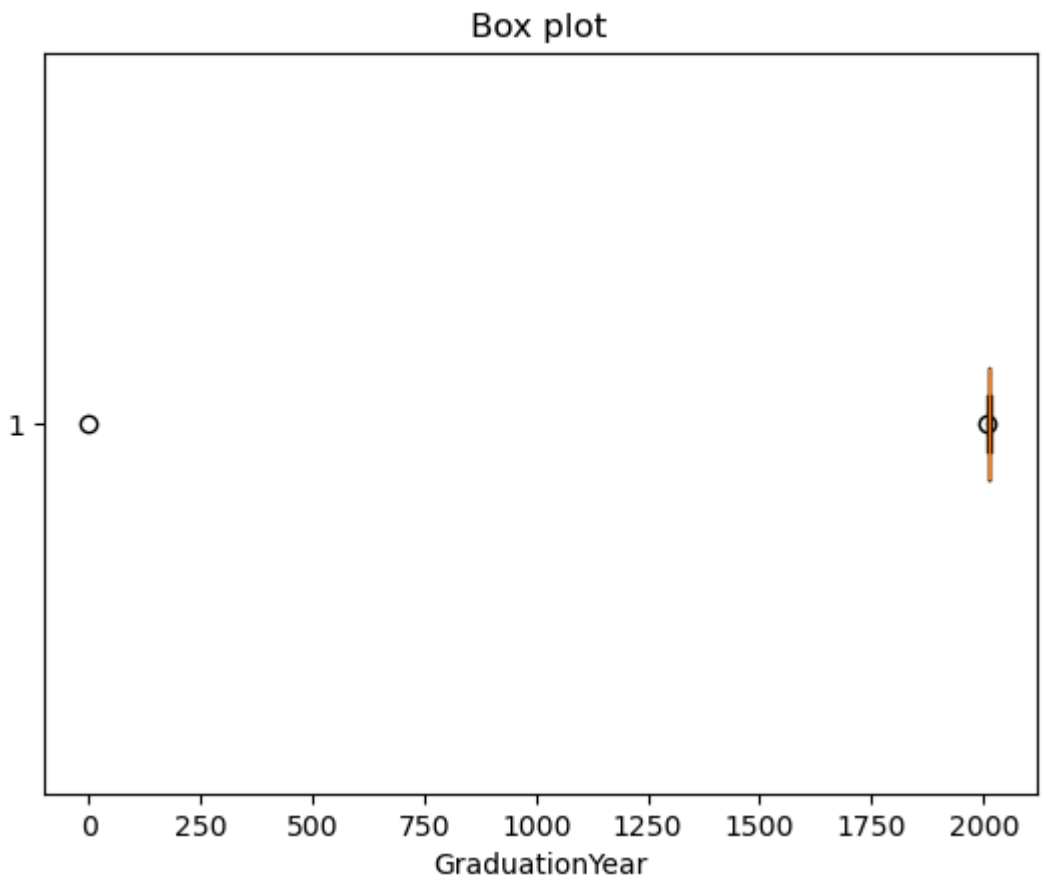

Box plot
1
0
2500
5000
7500
10000
12500
15000
17500
CollegeID

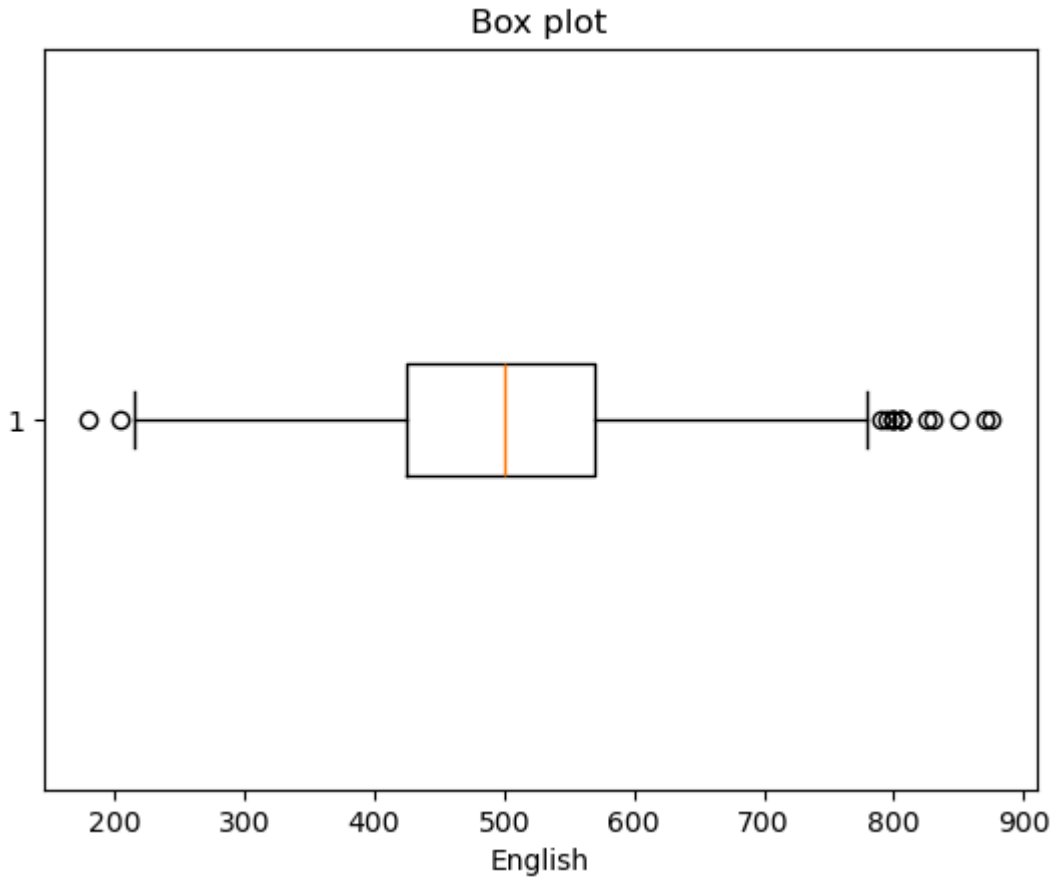

Box plot
1
1.0
1.2
1.4
1.6
1.8
2.0
CollegeTier

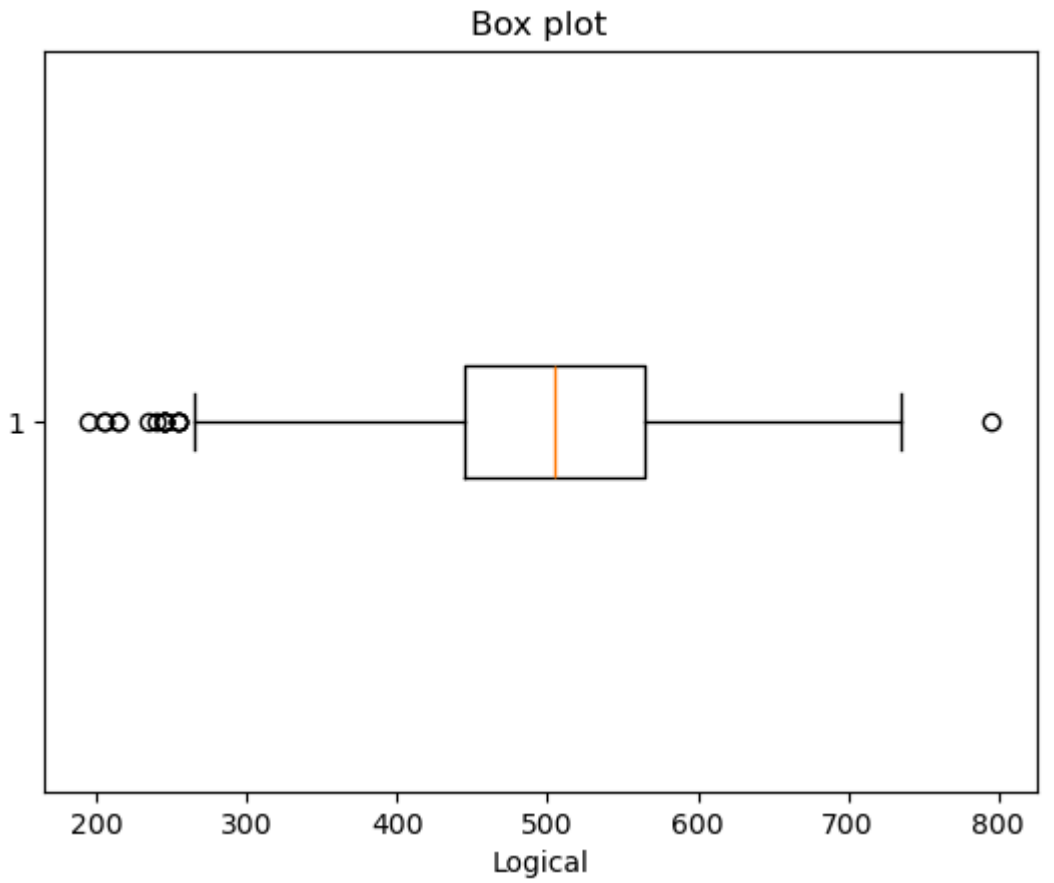

Box plot
1
20
40
60
80
100
collegeGPA

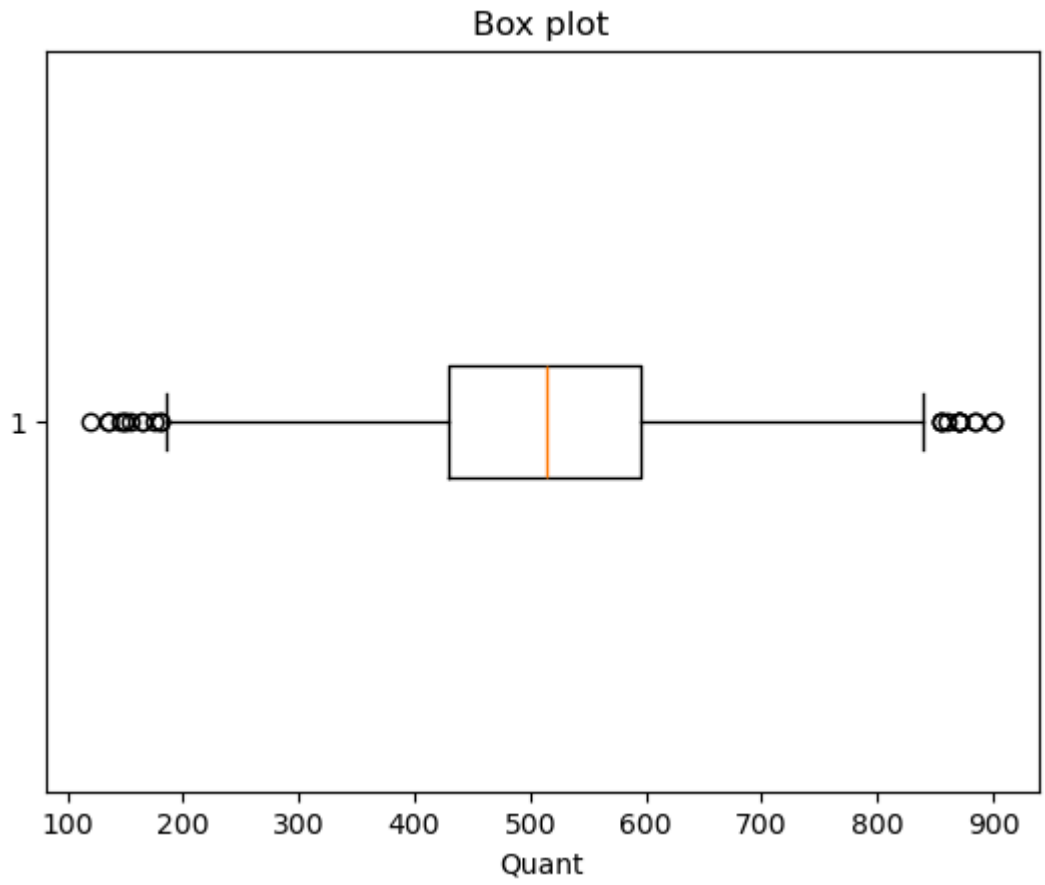

Box plot
1
0
2500
5000
7500
10000
12500
15000
17500
CollegeCityID

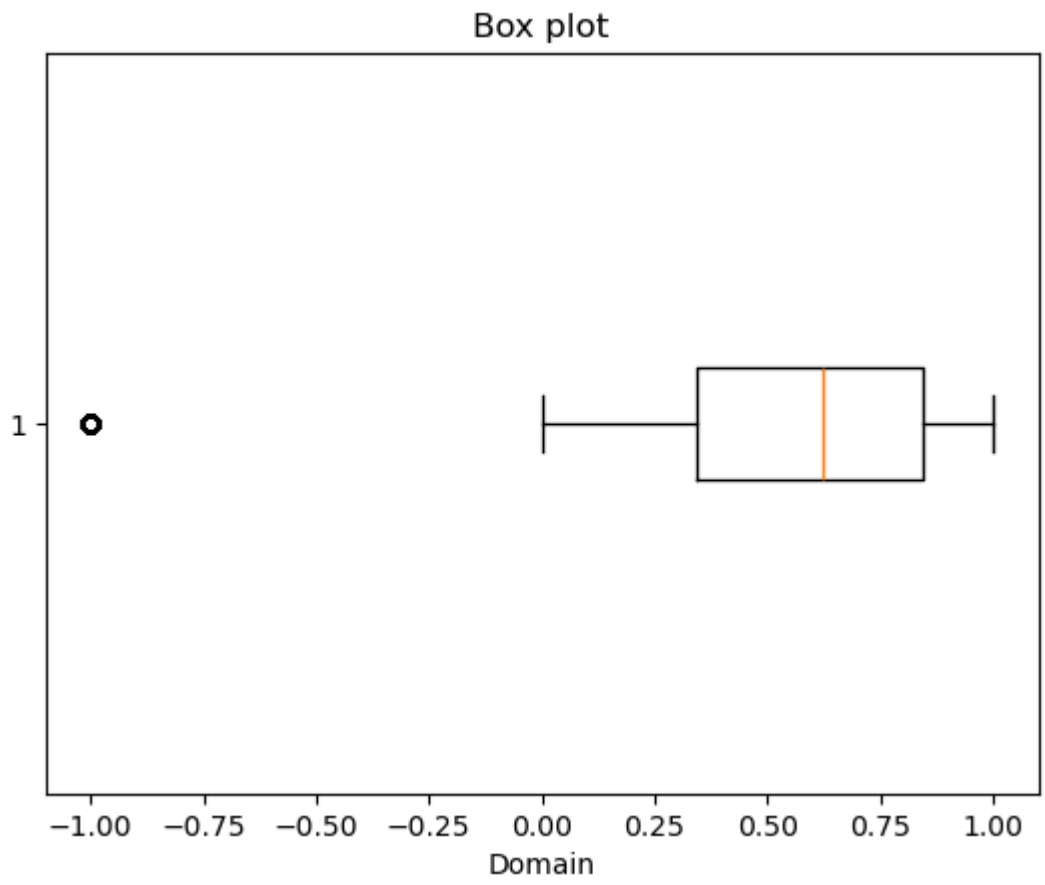

Box plot
1
0.0
0.2
0.4
0.6
0.8
1.0
CollegeCityTier

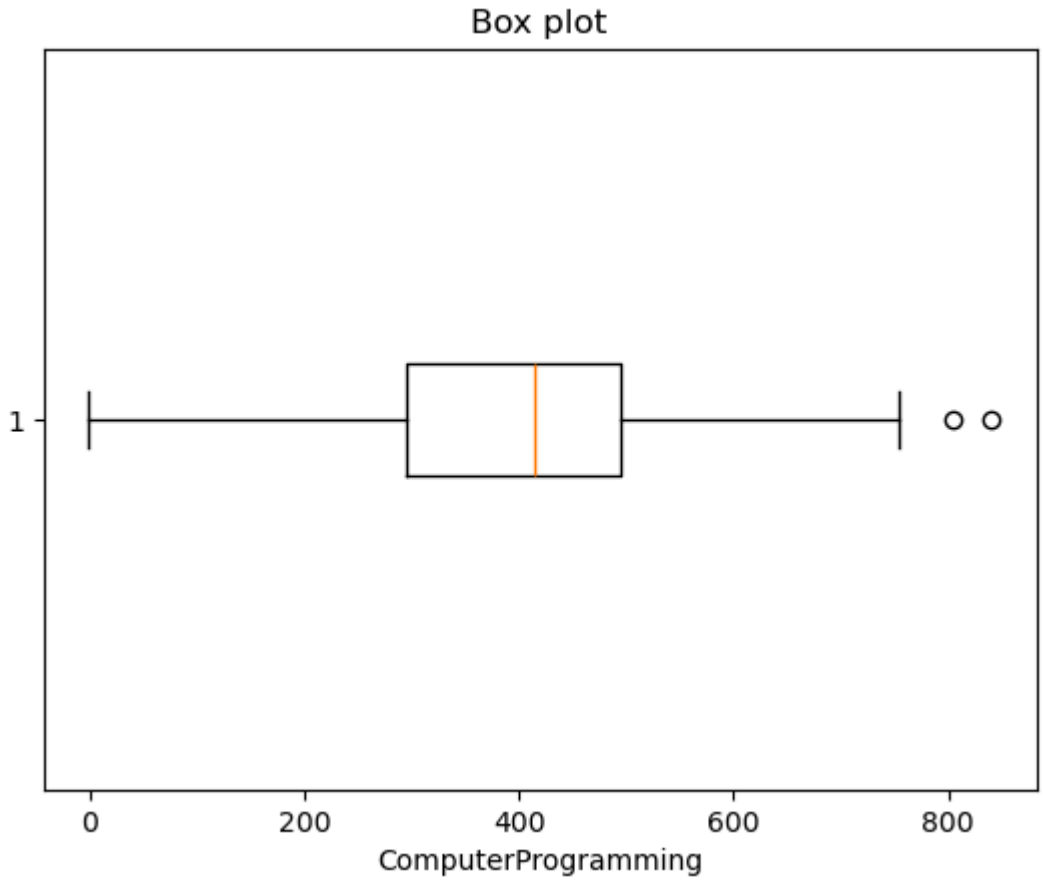

Box plot
1
0
250
500
750
1000
1250
1500
1750
2000
GraduationYear

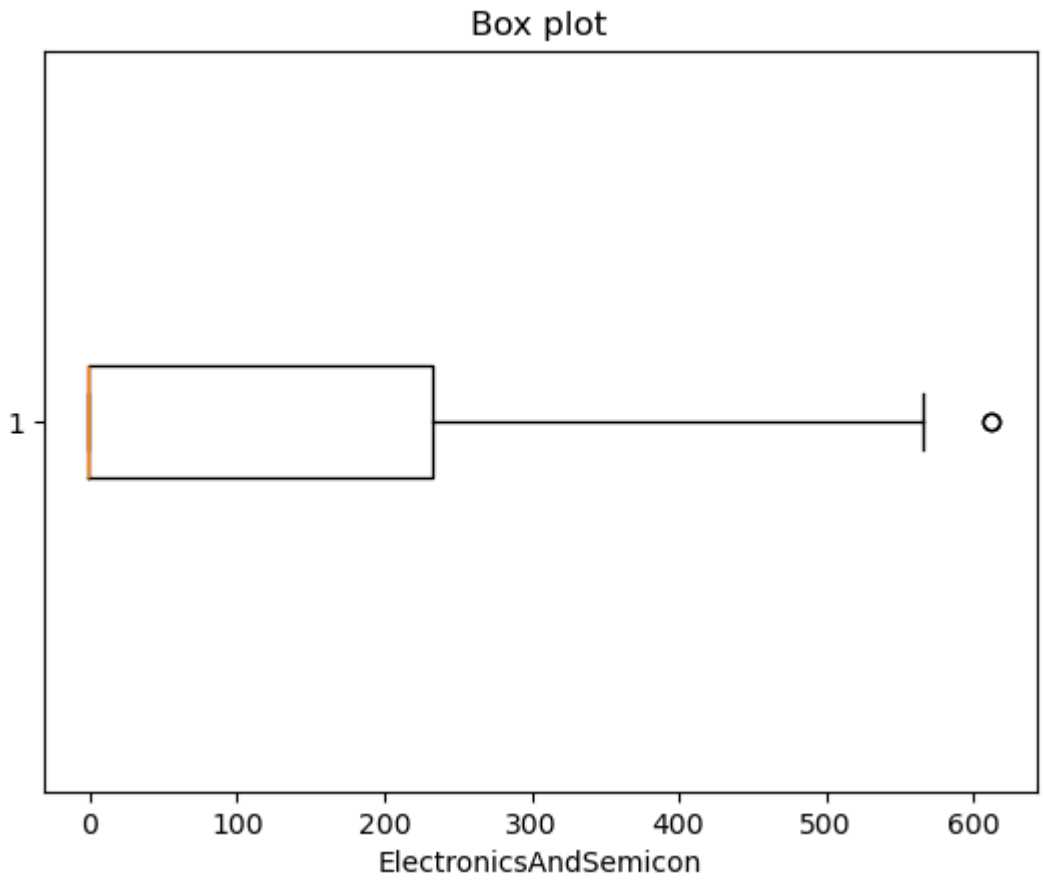

Box plot
1
200
300
400
500
600
700
800
900
English

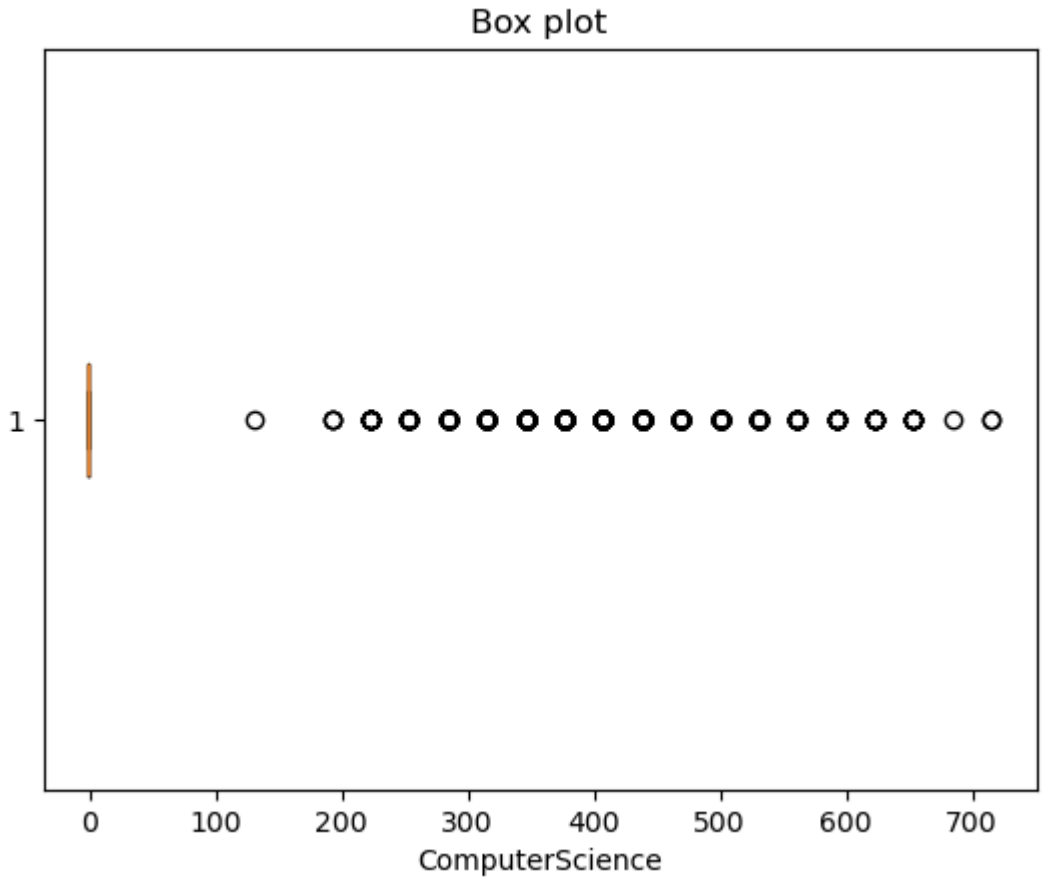

Box plot
1
200
300
400
500
600
700
800
Logical

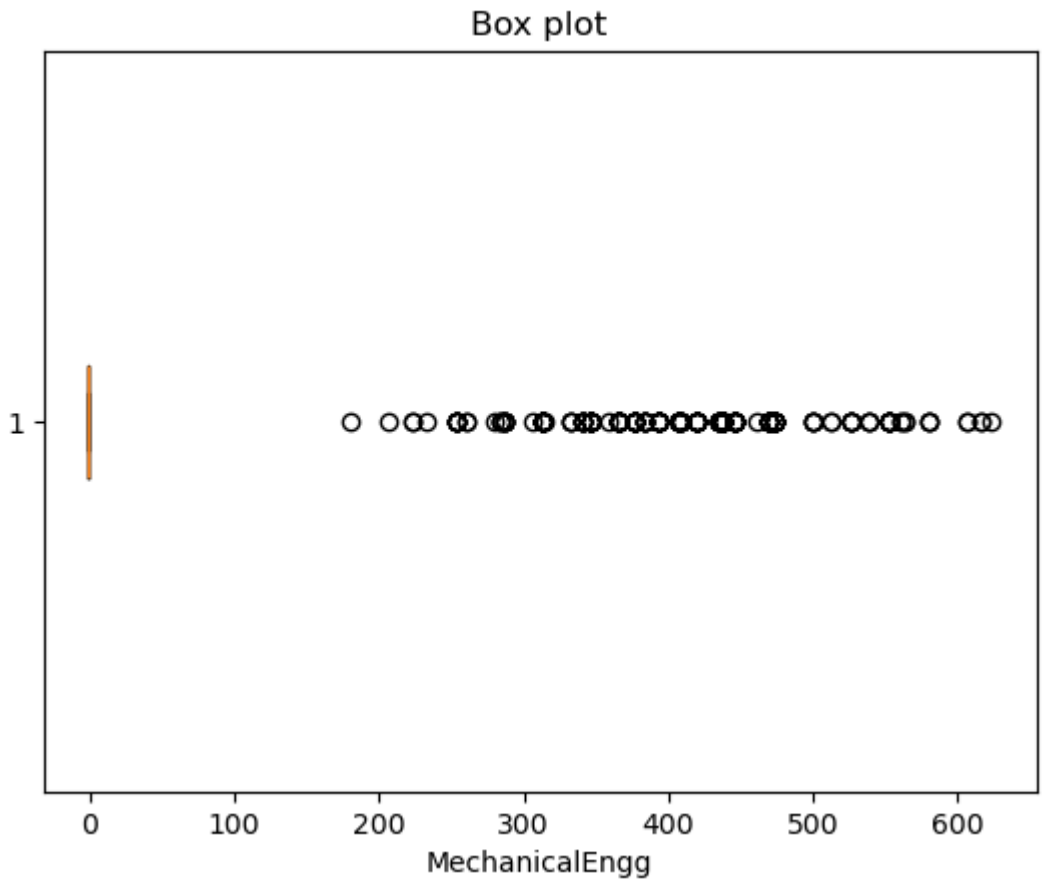

Box plot
1
100
200
300
400
500
600
700
800
900
Quant

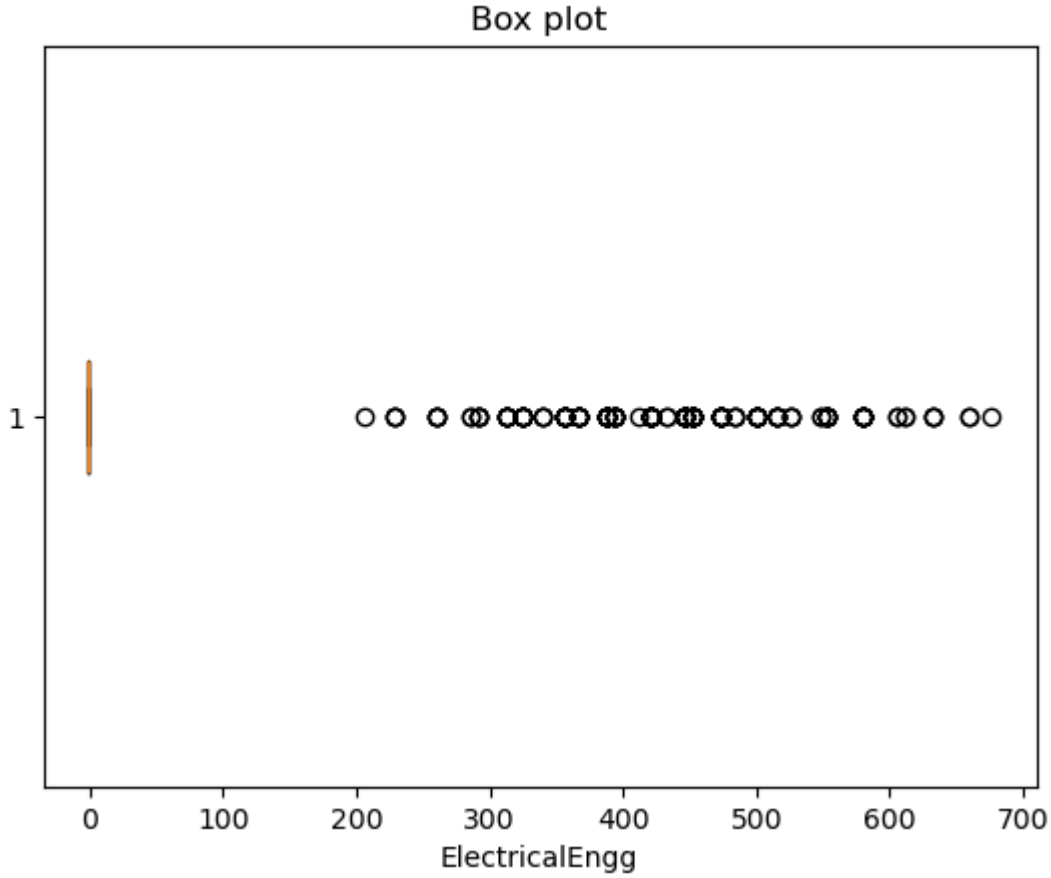

Box plot
1
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Domain

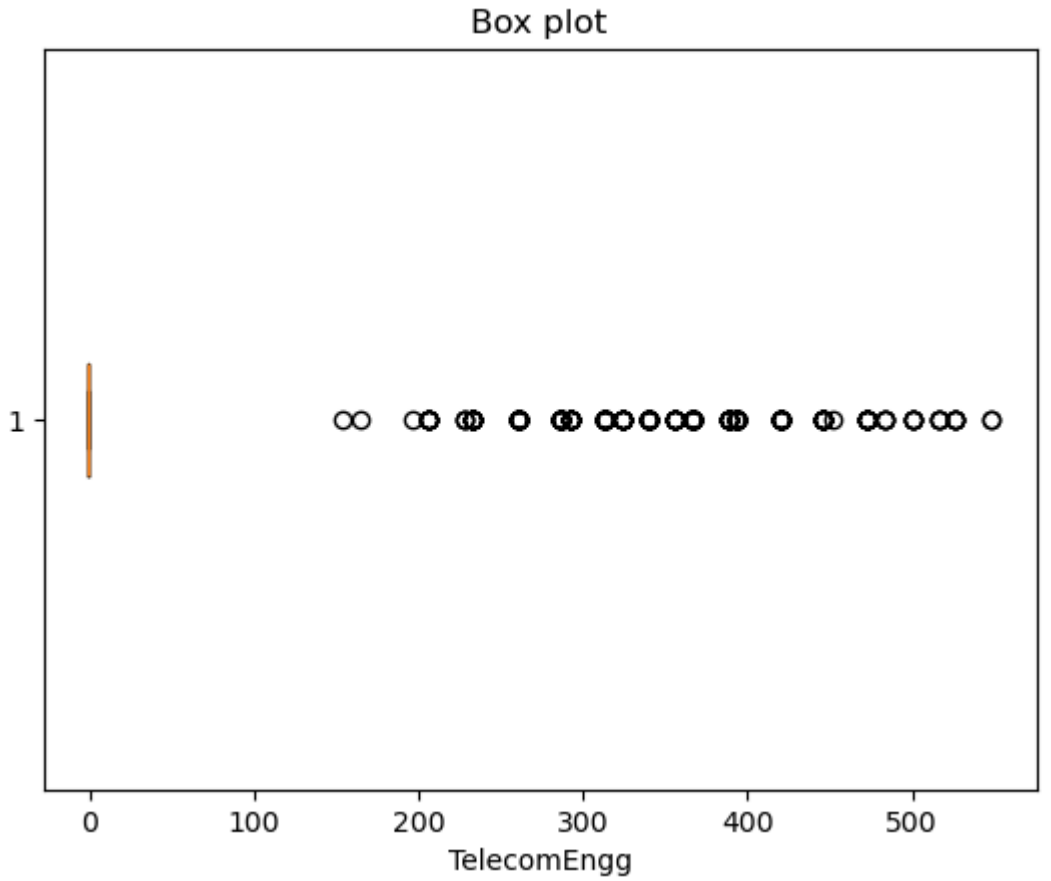

Box plot
1
0
200
400
600
800
ComputerProgramming

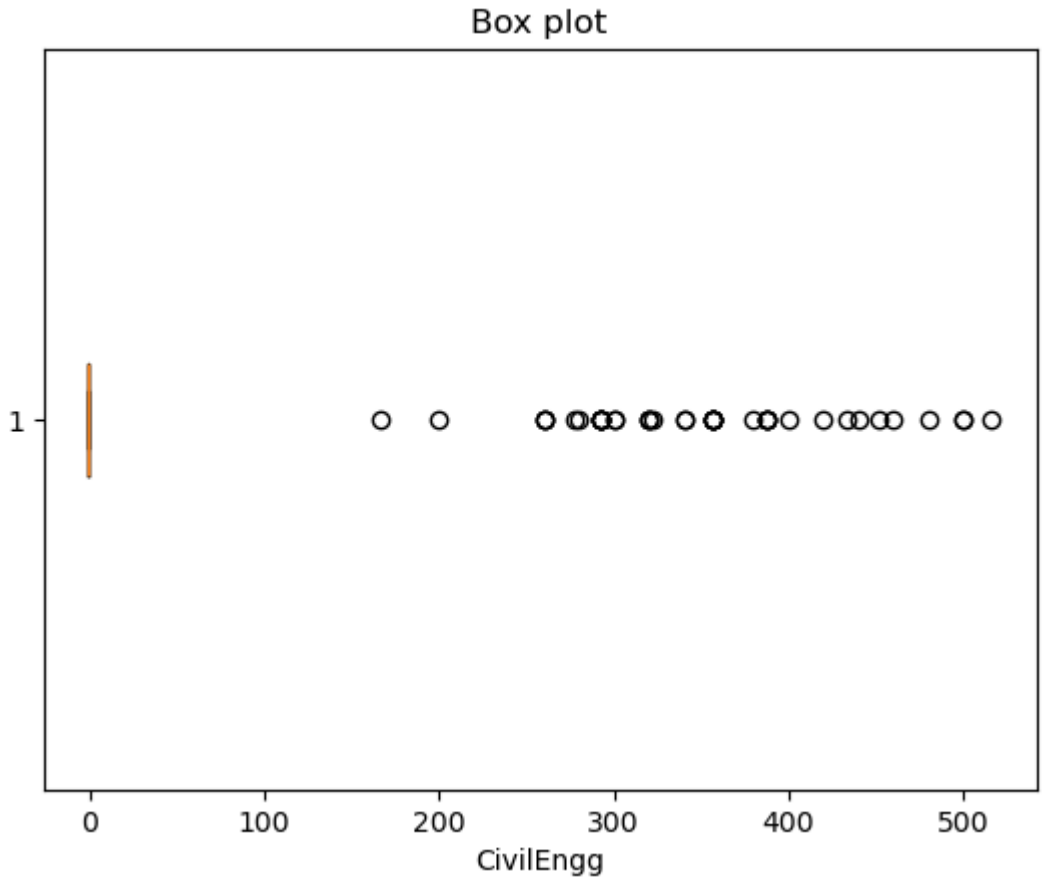

Box plot
1
0
100
200
300
400
500
600
ElectronicsAndSemicon

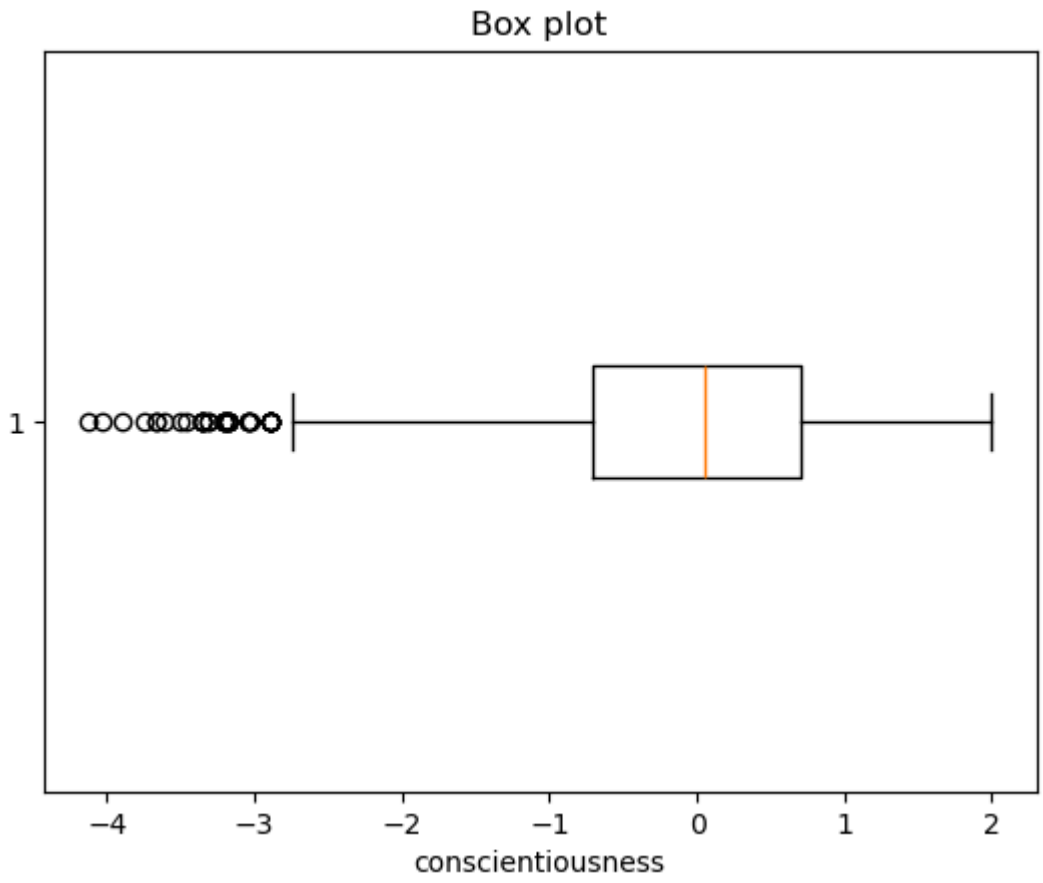

Box plot
1
0
100
200
300
400
500
600
700
ComputerScience

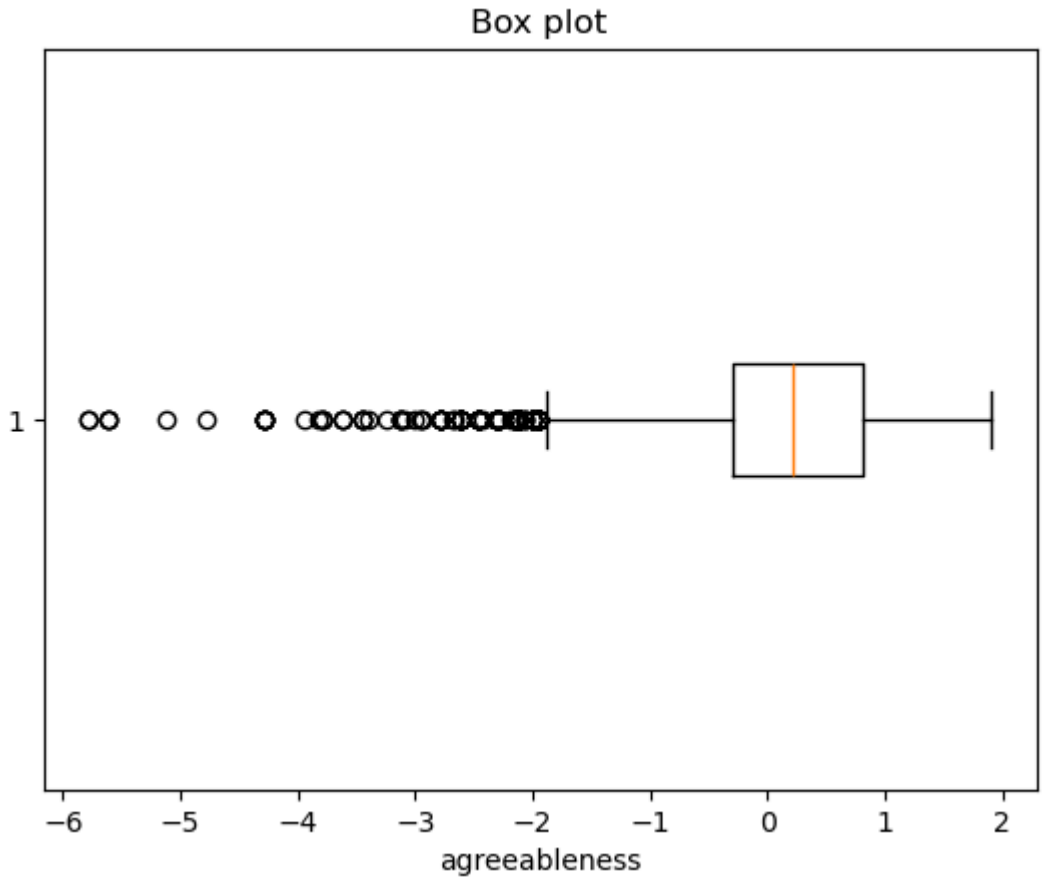

Box plot
1
0
100
200
300
400
500
600
MechanicalEngg

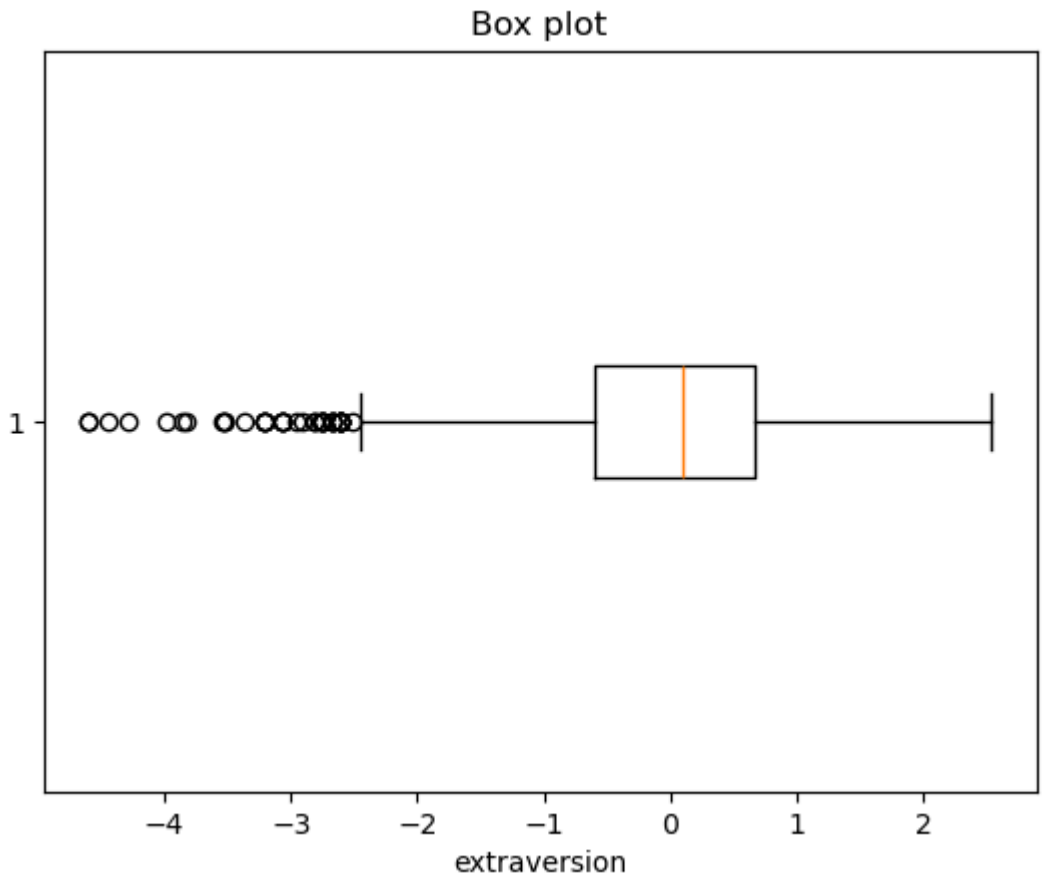

Box plot
1
0
100
200
300
400
500
600
700
ElectricalEngg

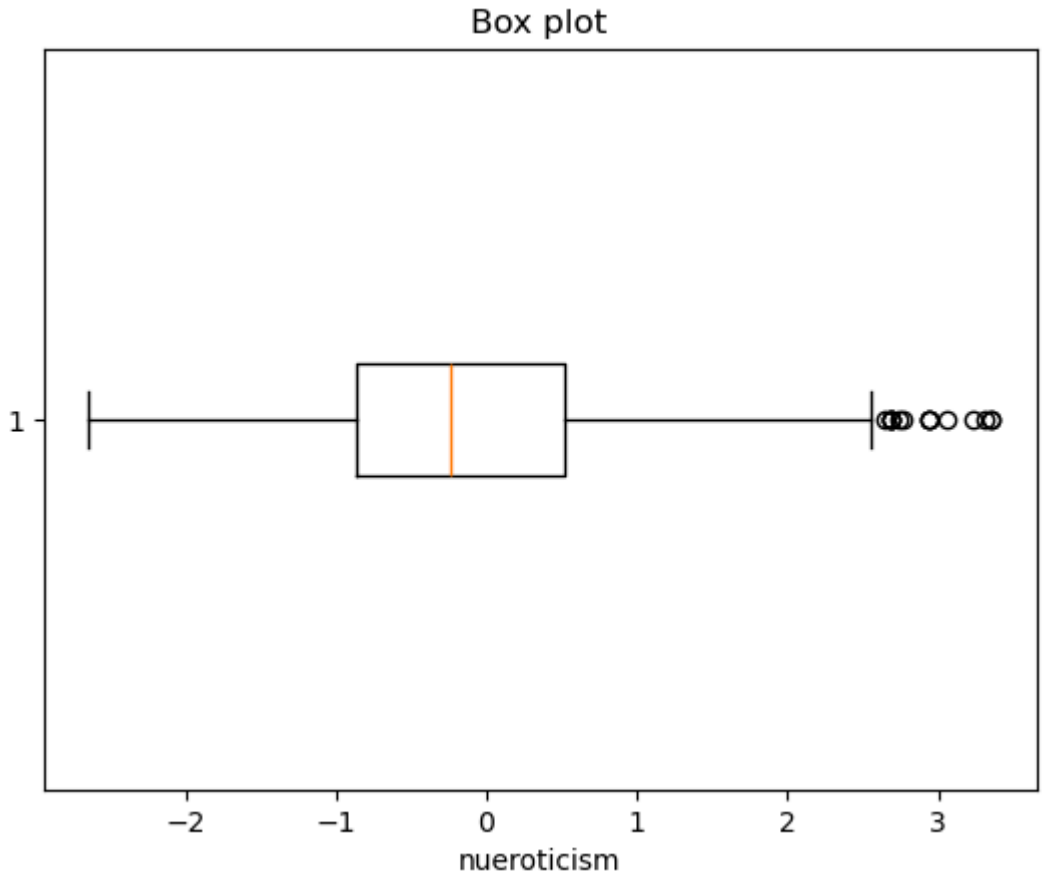

Box plot
1
0
100
200
300
400
500
TelecomEngg

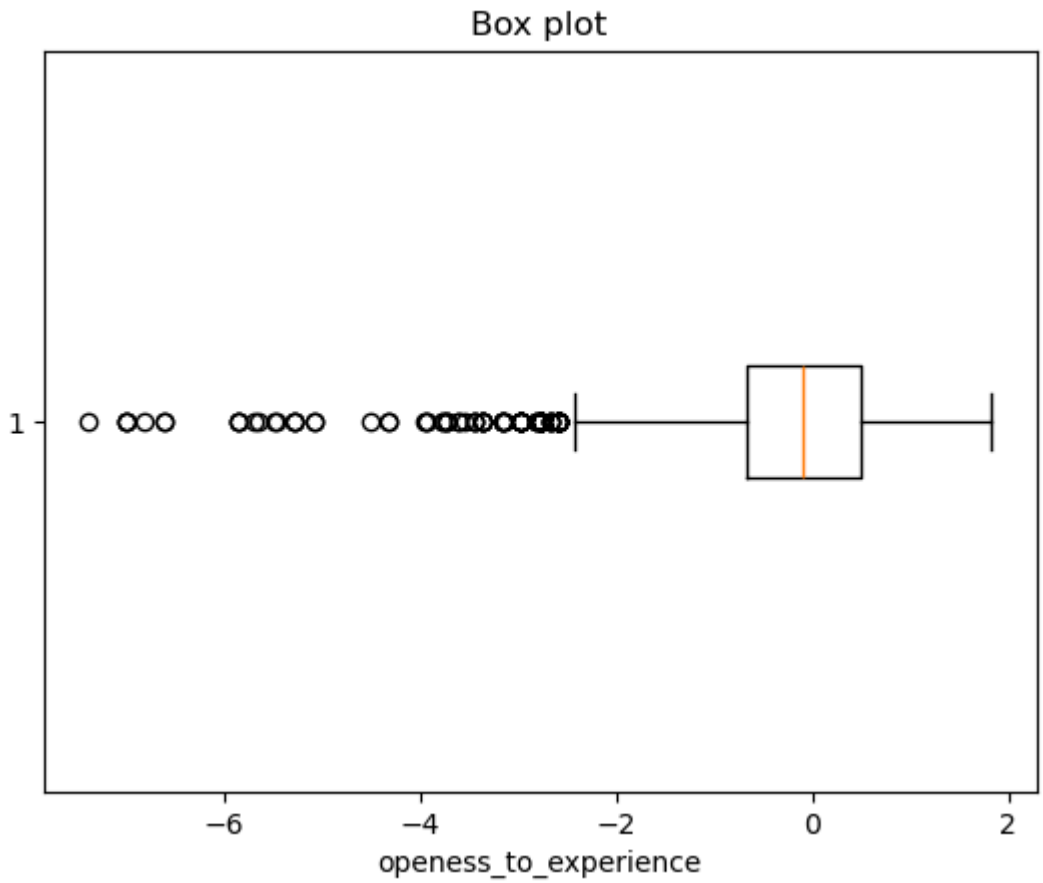

Box plot
1
0
100
200
300
400
500
CivilEngg


Box plot
1
−4
−3
−2
−1
0
1
2
conscientiousness


Box plot
1
−6
−5
−4
−3
−2
−1
0
1
2
agreeableness


Box plot
1
−4
−3
−2
−1
0
1
2
extraversion





In [58]:
```
# by using IQR we are able to remove outliers in data
data=df[i]
q1=np.percentile(data,25)
q2=np.percentile(data,50)
q3=np.percentile(data,75)
```

```
IQR=q3-q1

lb=(q1-(1.5*IQR))
ub=(q3+(1.5*IQR))

con1=data>lb
con2=data<ub
con3=con1&con2
non_outliers_data=data[con3]
non_outliers_data
```

Out[58]:
```
0      -0.4455
1       0.8637
2       0.6721
3      -0.9194
4      -0.1295
         ...
3993   -0.9194
3994   -0.0943
3995   -0.7615
3996   -0.0943
3997   -0.6035
Name: openess_to_experience, Length: 3903, dtype: float64
```

Extracted dataframe without any outliers

In [59]:
```
import warnings
warnings.filterwarnings('ignore')
non_outliers_df=df[con3]
non_outliers_df.dropna(inplace=True)
non_outliers_df
```

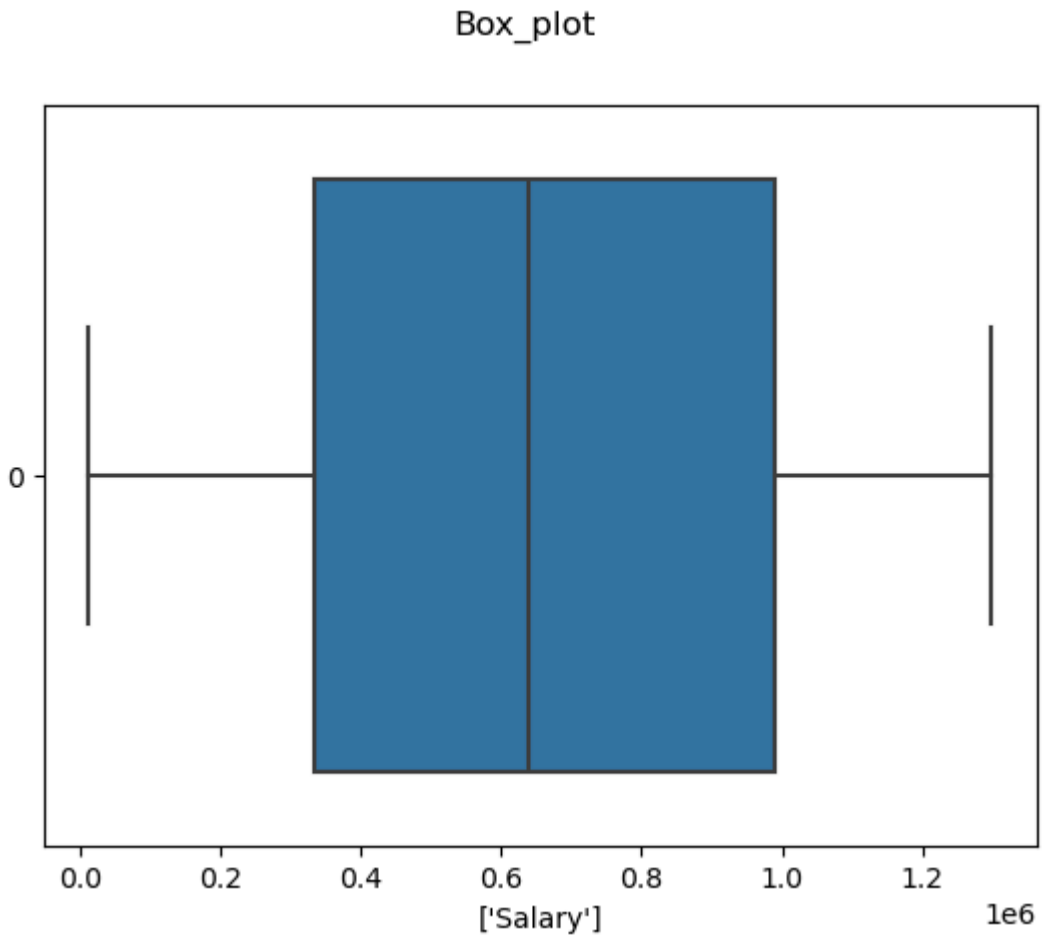
Out[59]:

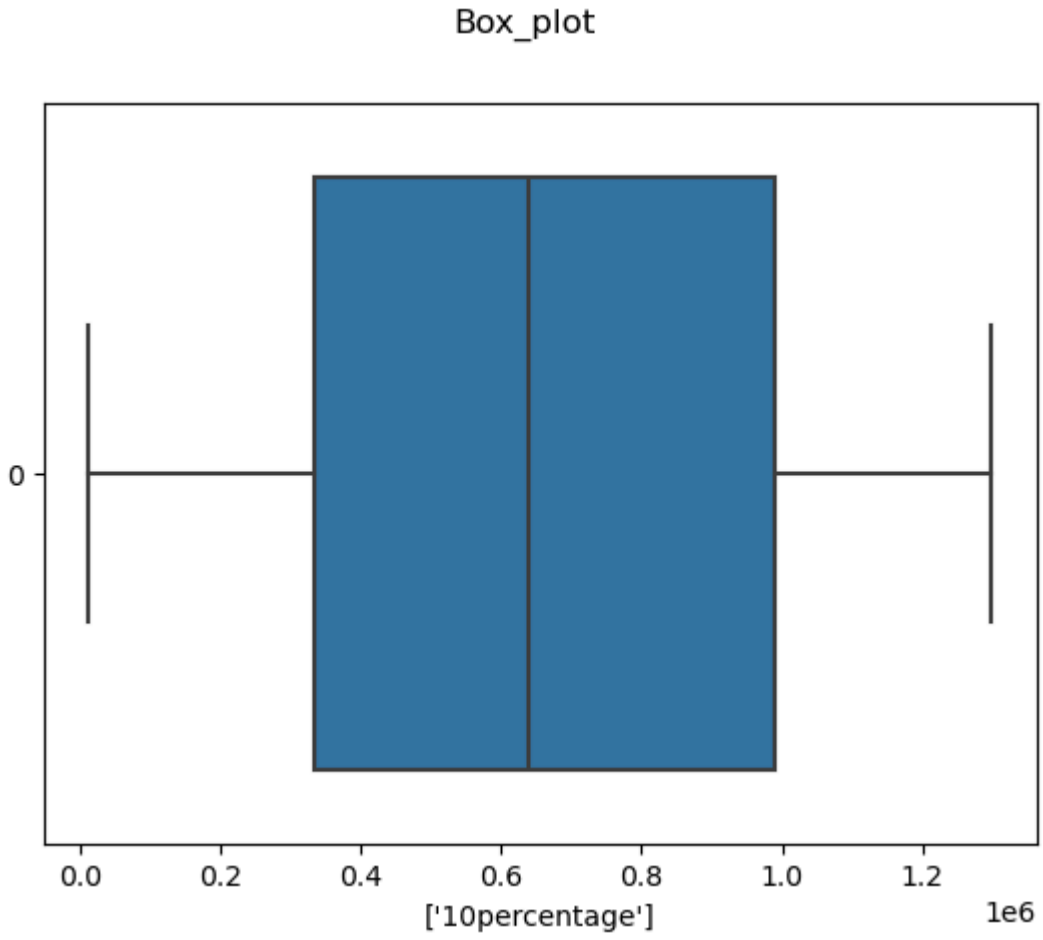
	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	G
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	

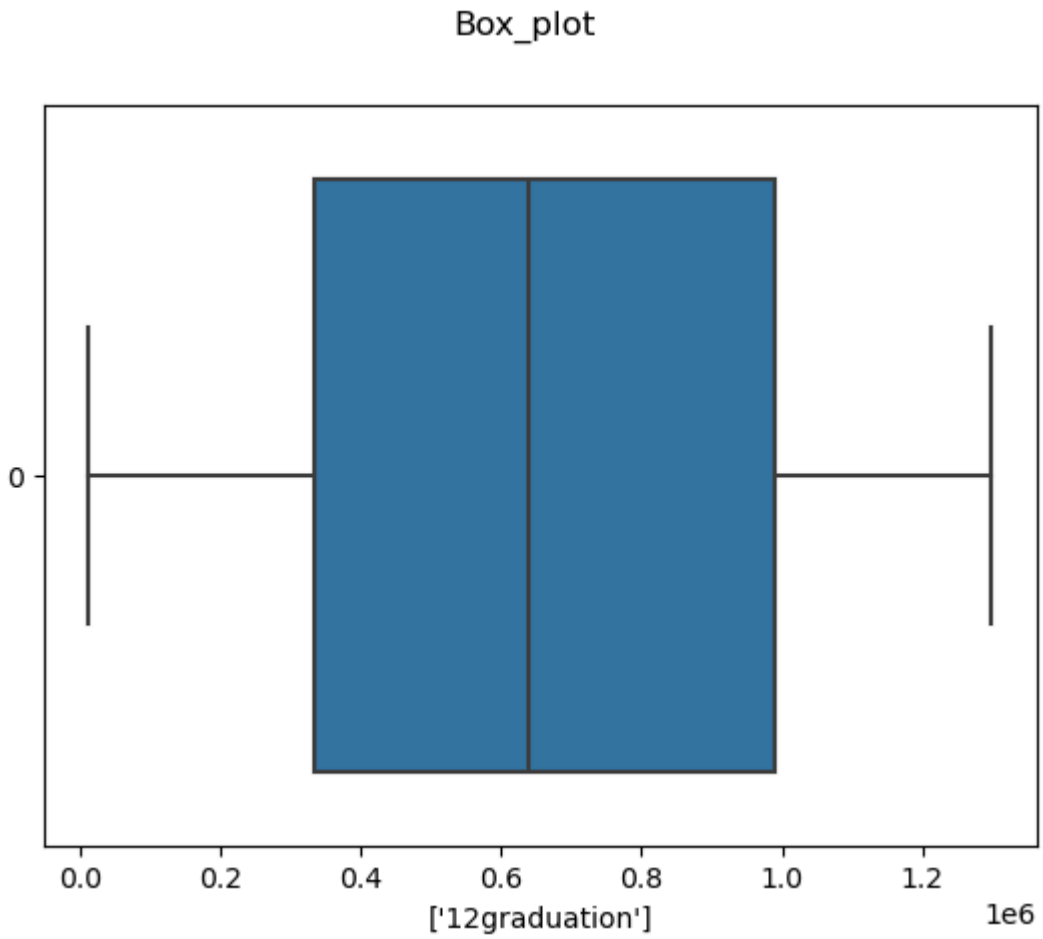
3903 rows × 39 columns

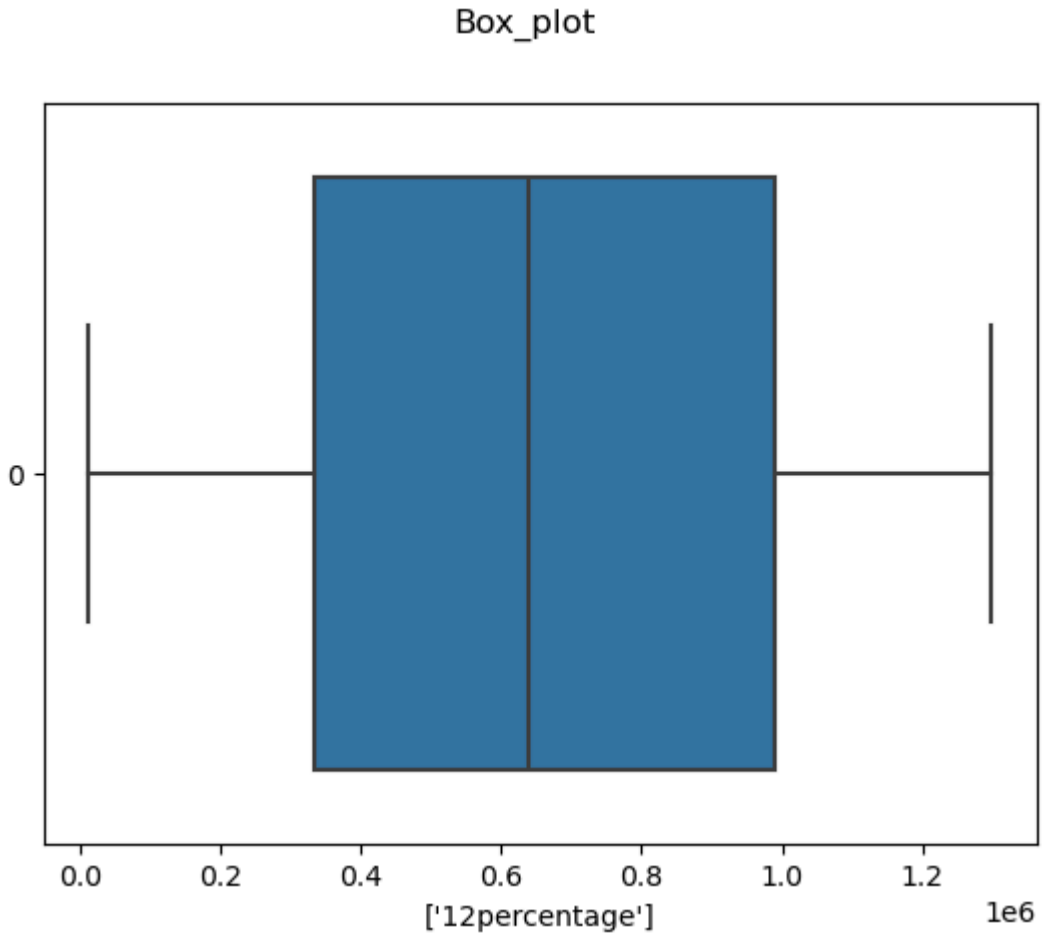
Box-plot After removal of outliers

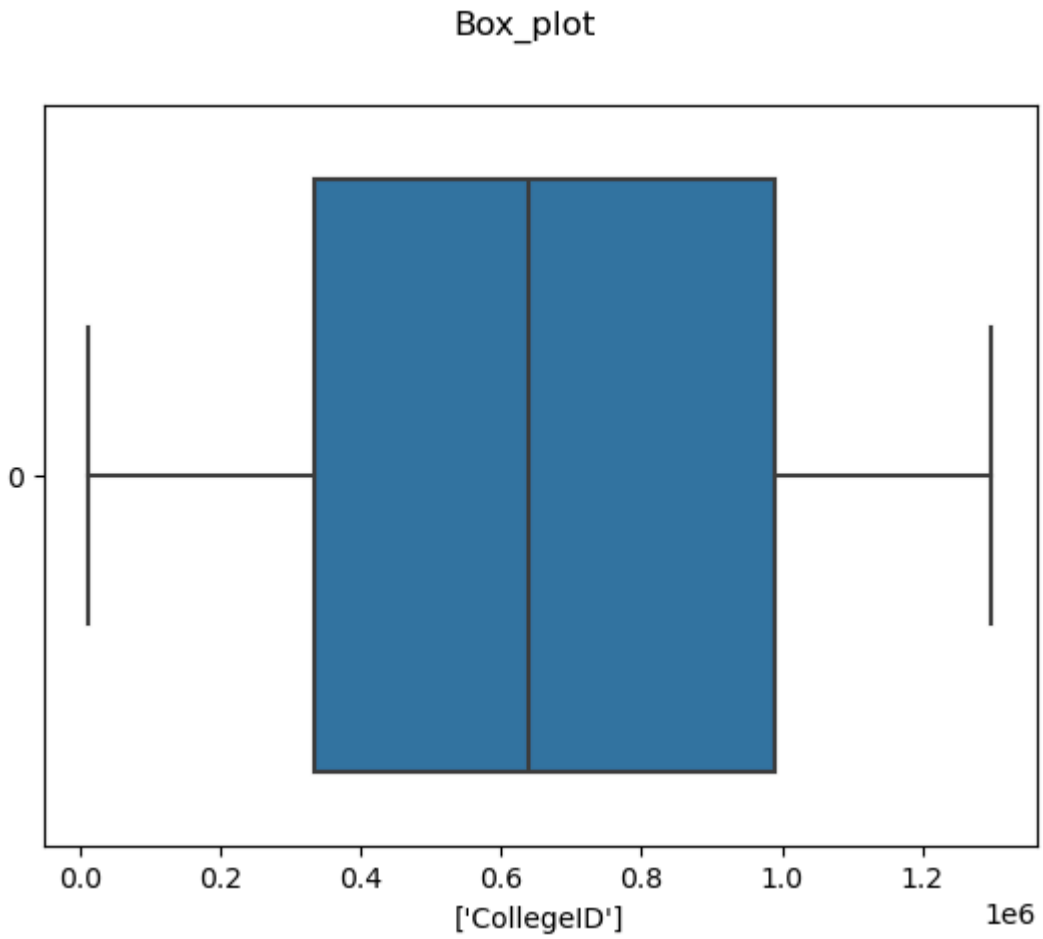
In [61]:
```
for i in num_cols[1:]:
    height=non_outliers_df['ID']
    sns.boxplot(height,orient='h')
    plt.xlabel([i])
    plt.suptitle('Box_plot')
    plt.show()
```

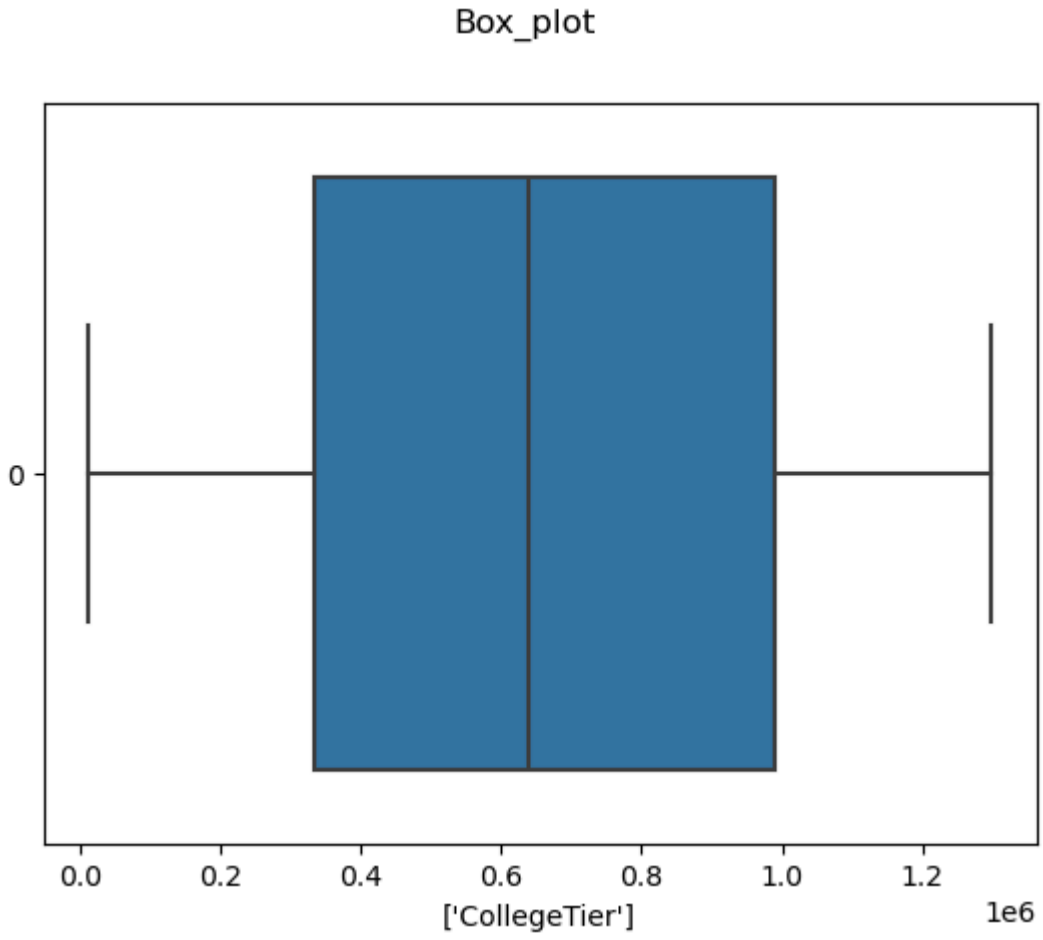

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['Salary']
1e6

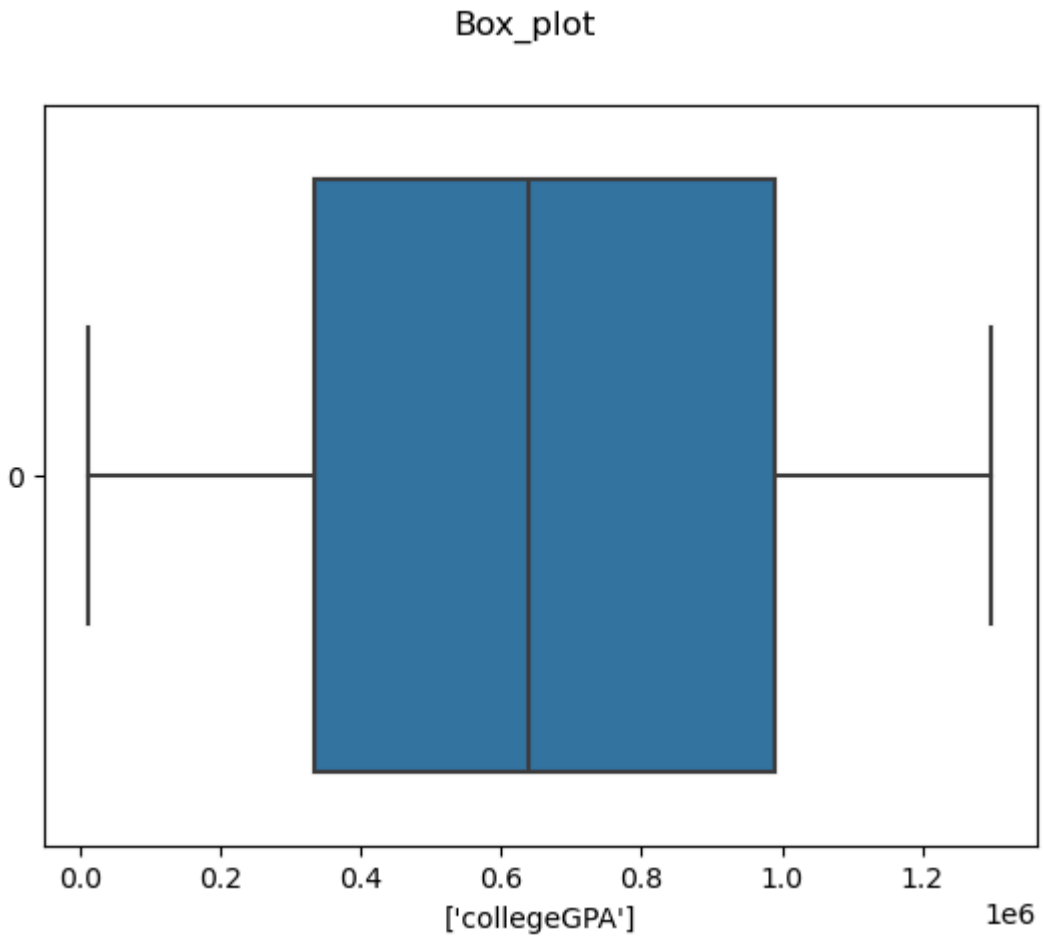

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['10percentage']
1e6

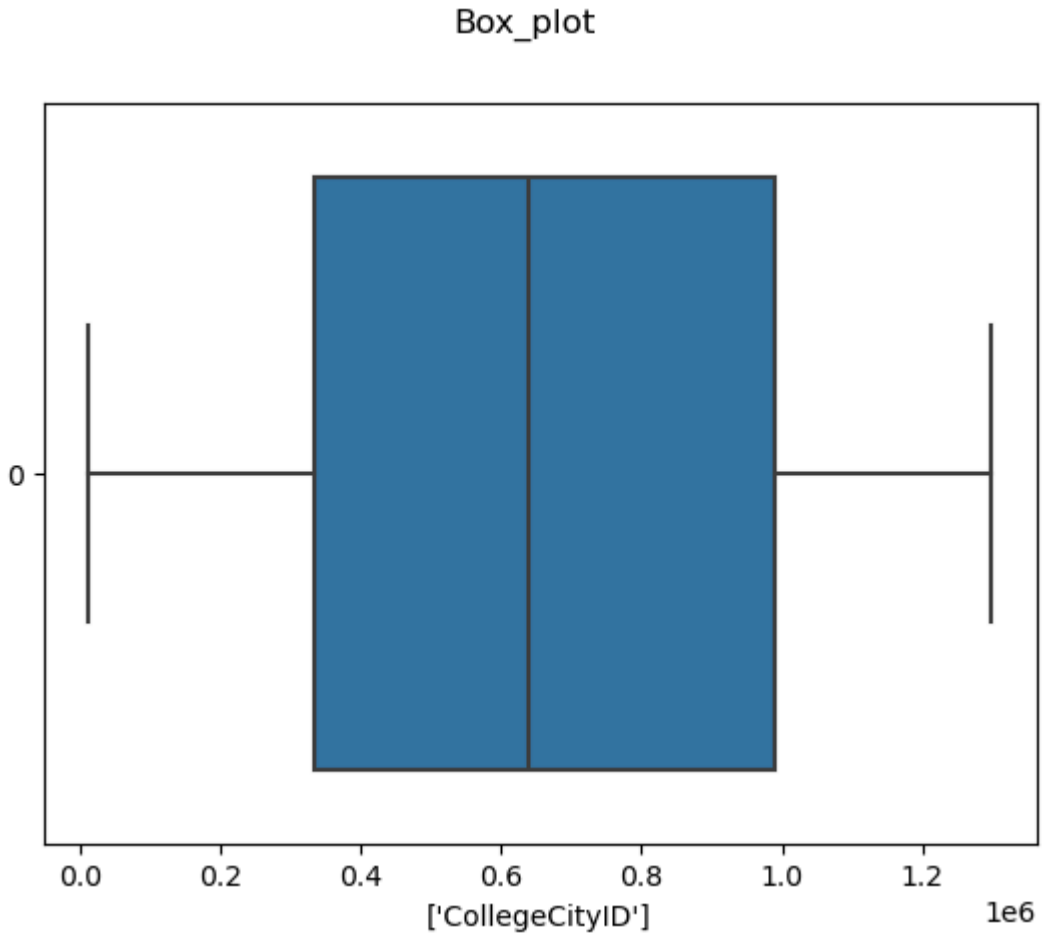

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['12graduation']
1e6

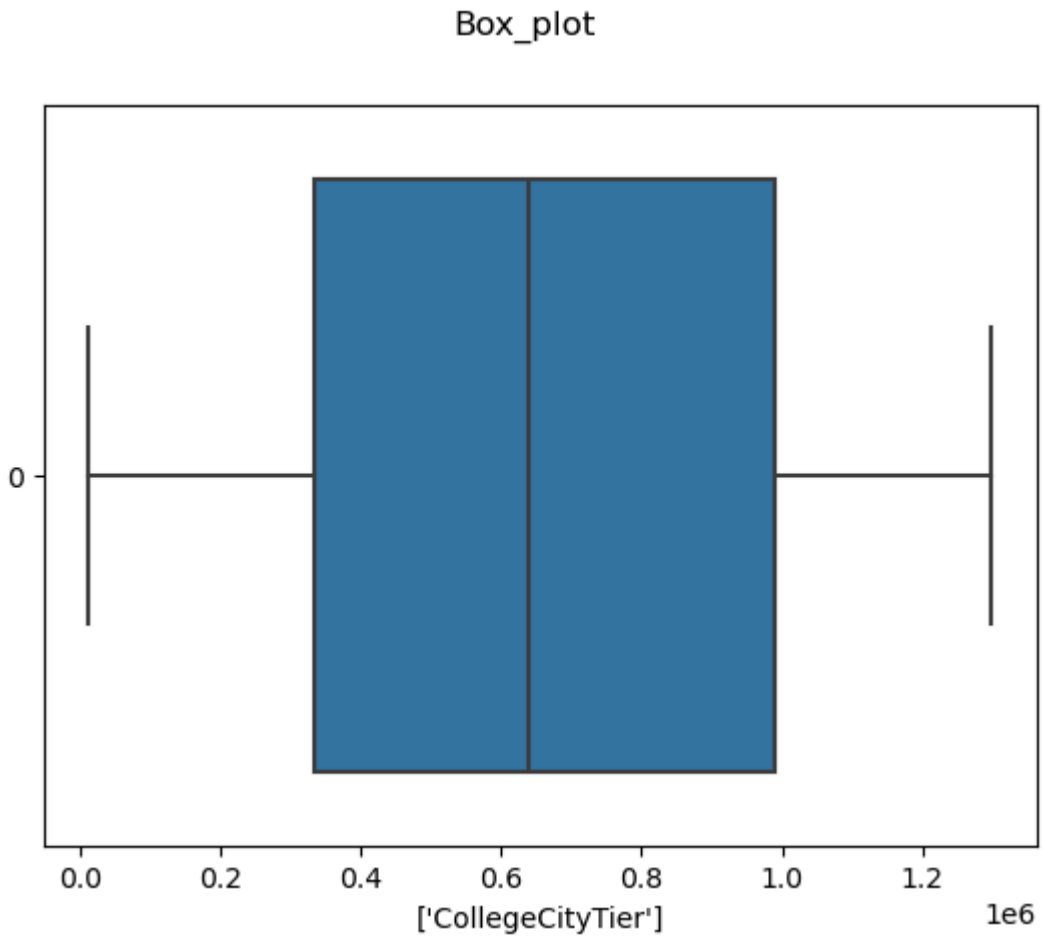

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['12percentage']
1e6

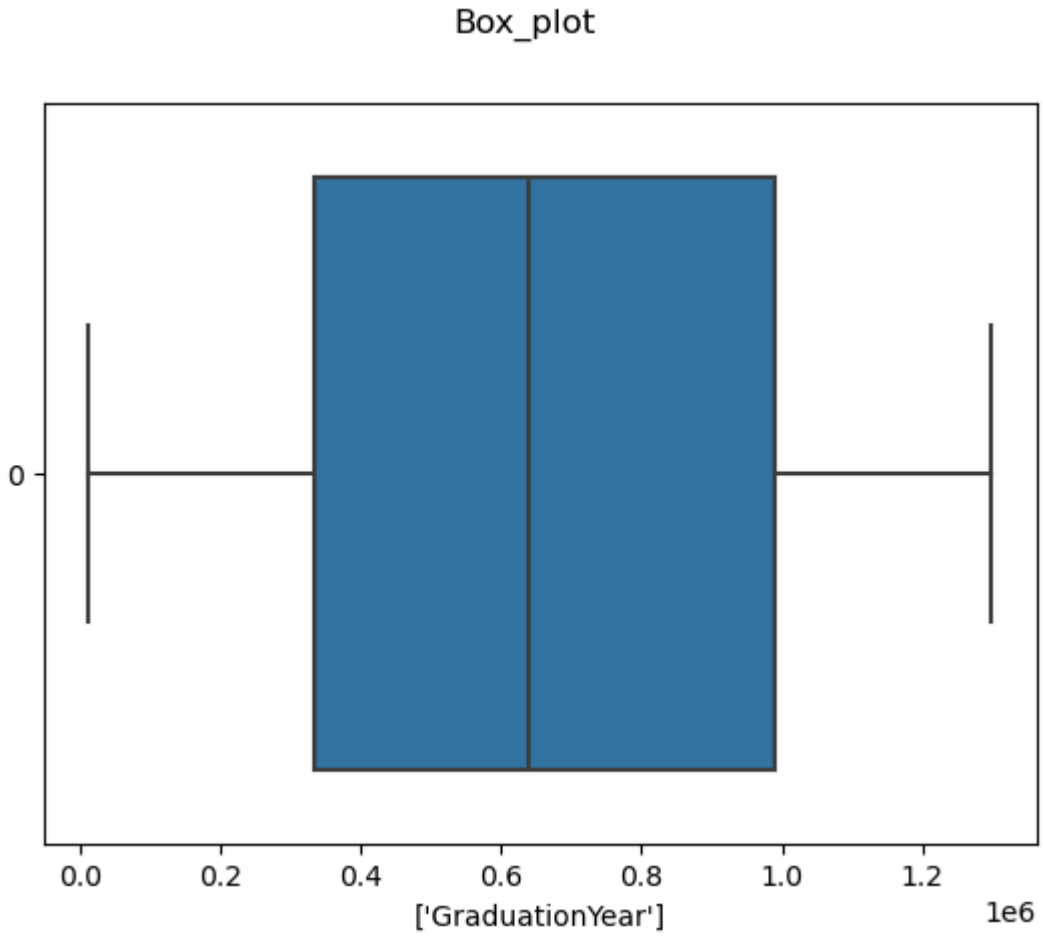

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['CollegeID']
1e6

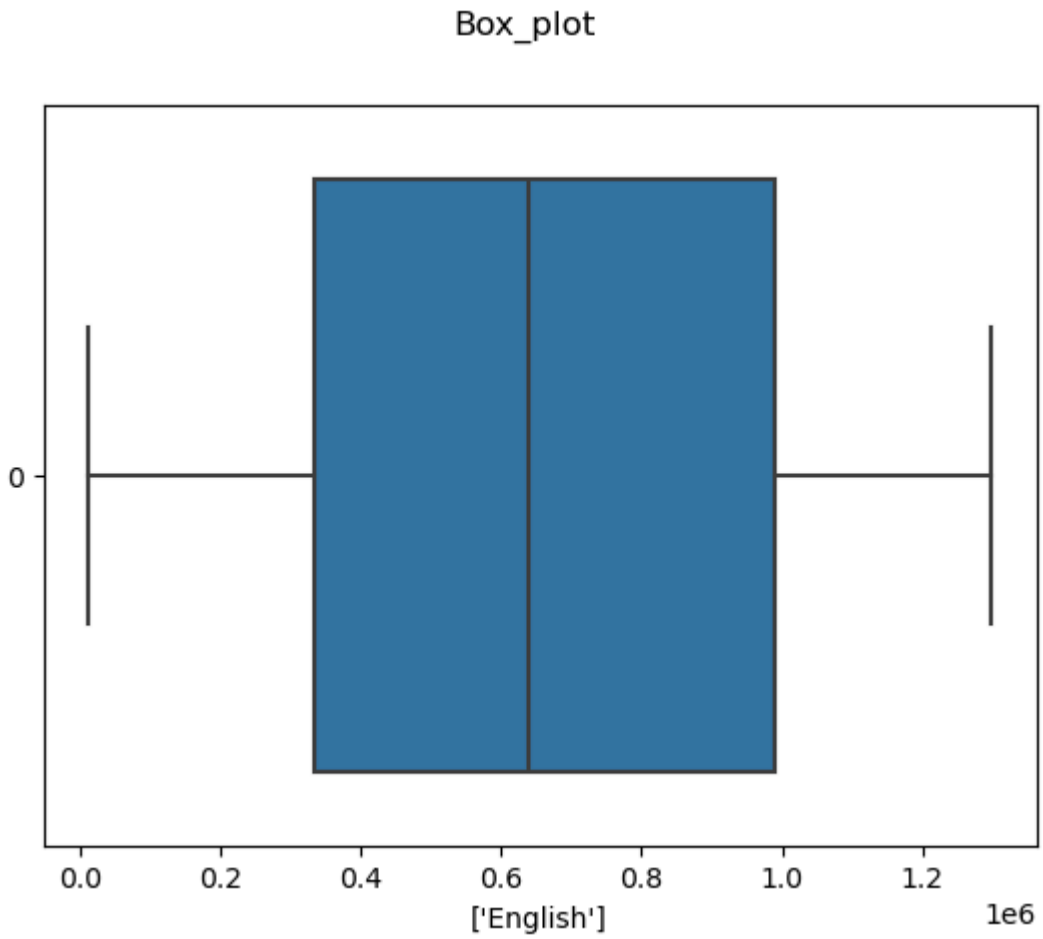

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['CollegeTier']
1e6

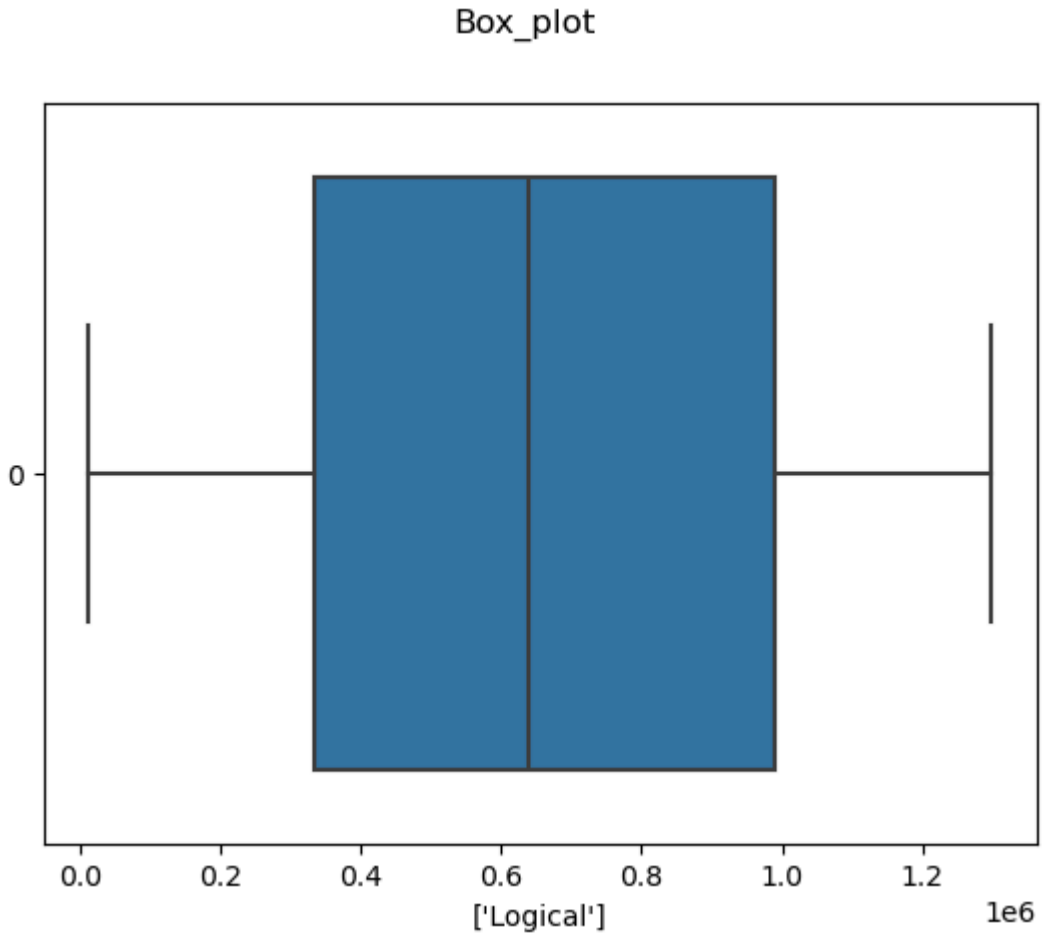

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['collegeGPA']
1e6

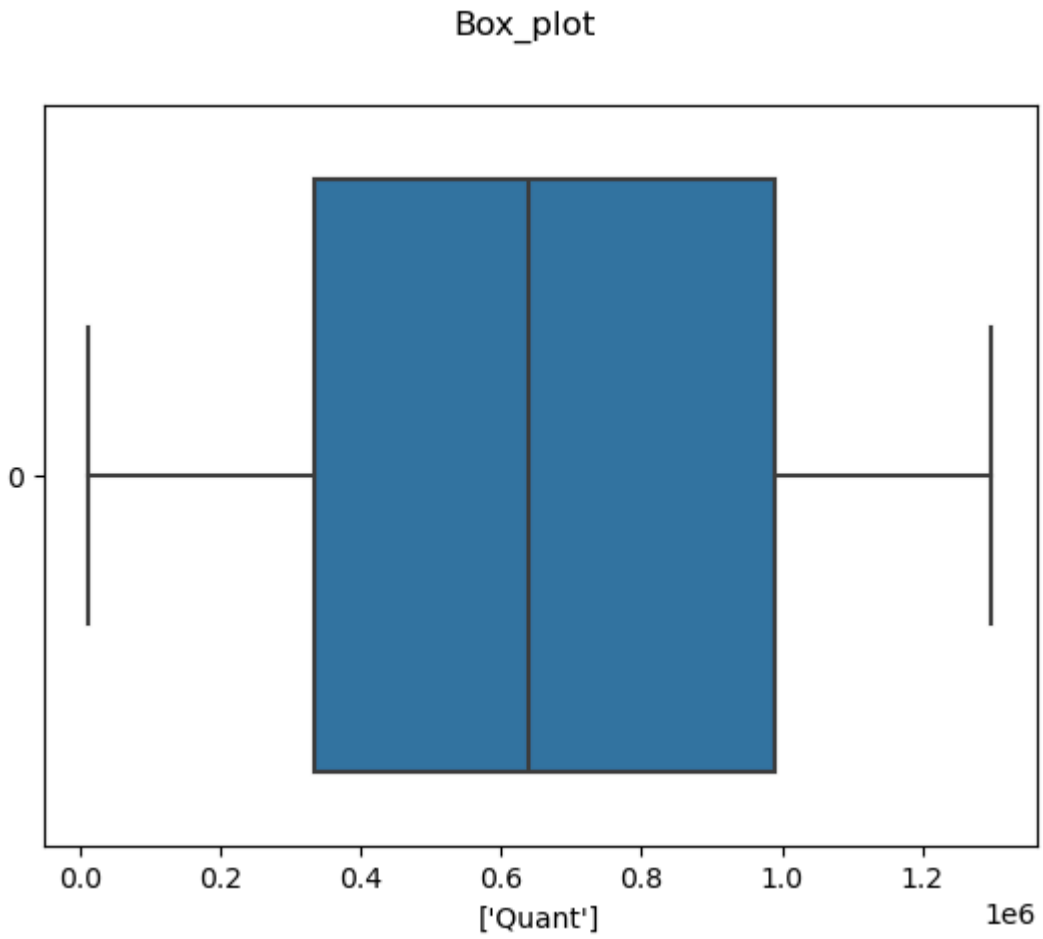

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['CollegeCityID']
1e6

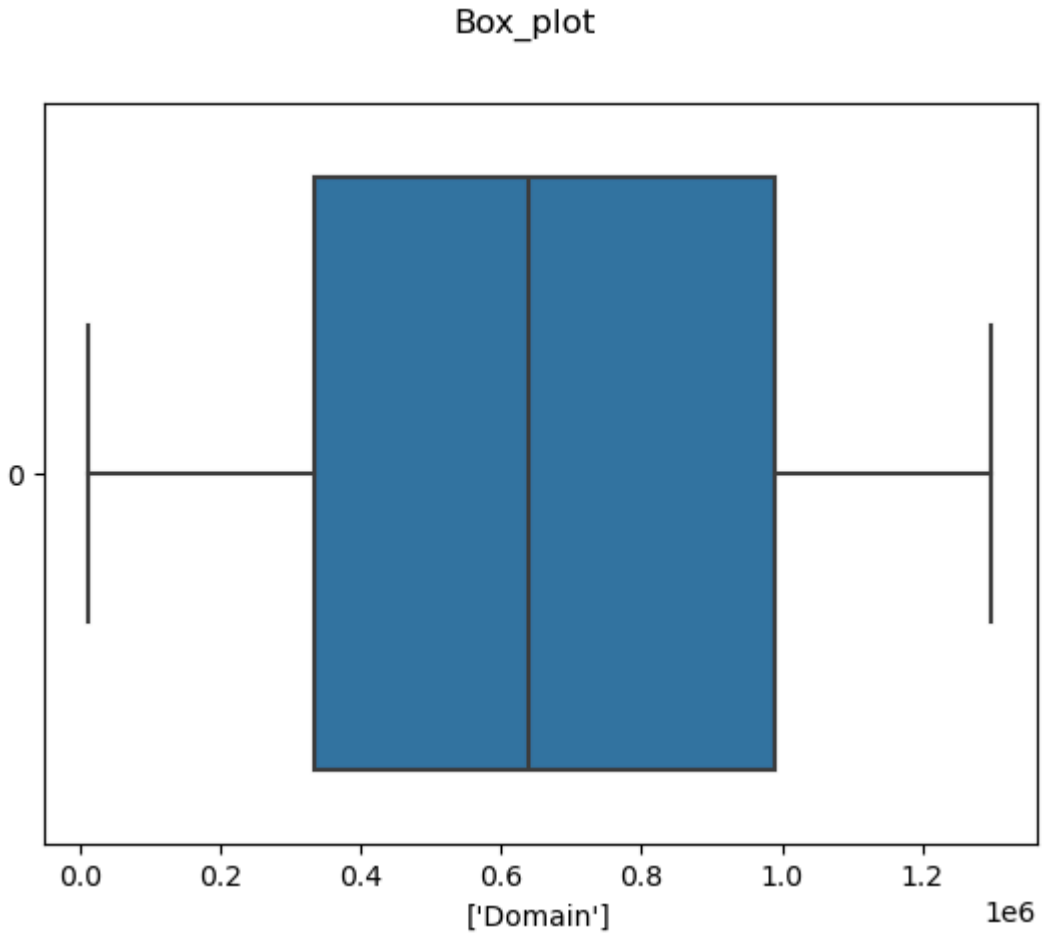

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['CollegeCityTier']
1e6

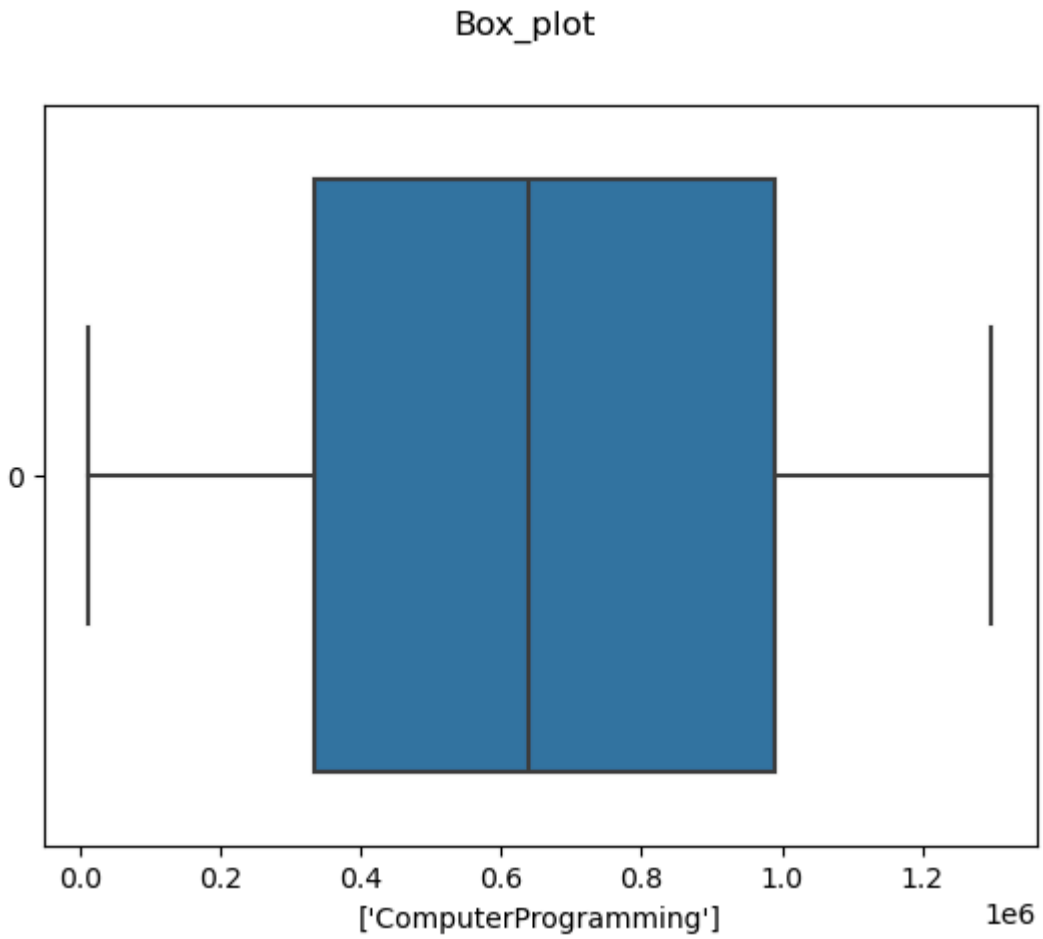

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['GraduationYear']
1e6

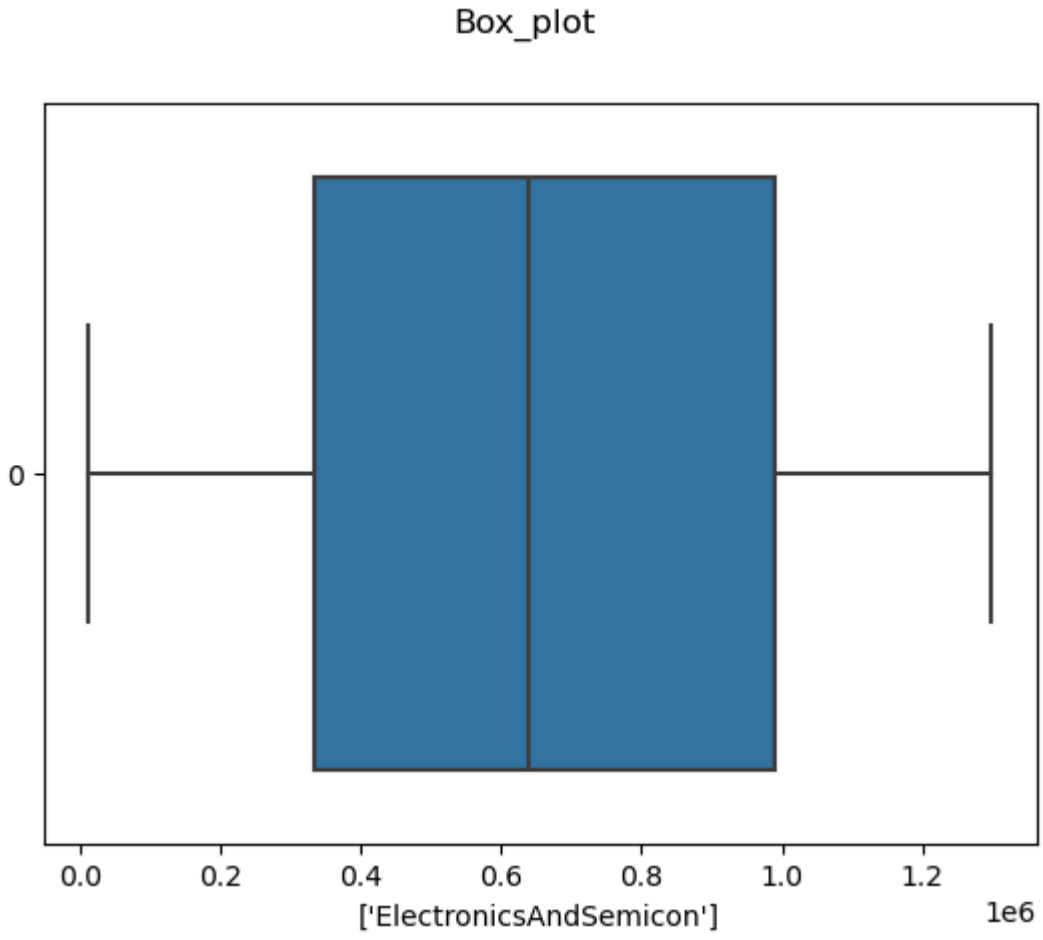

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['English']
1e6

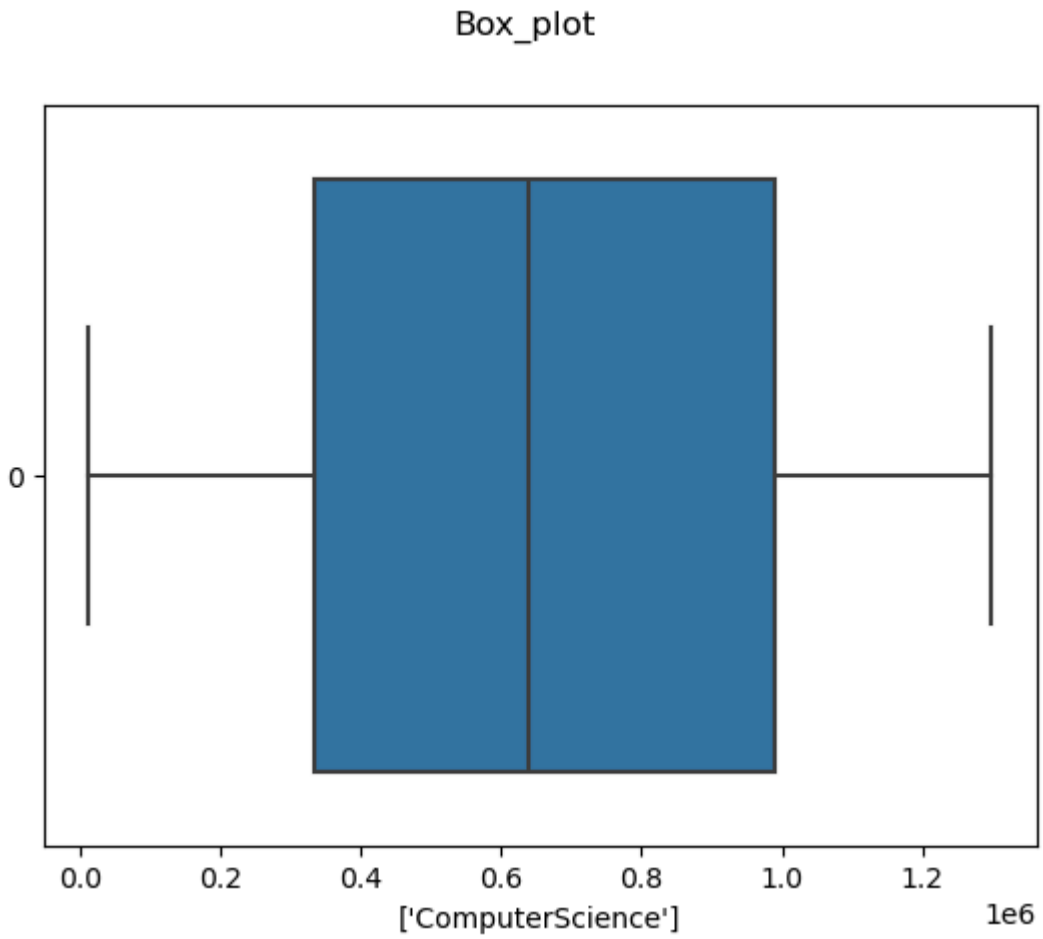

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['Logical']
1e6

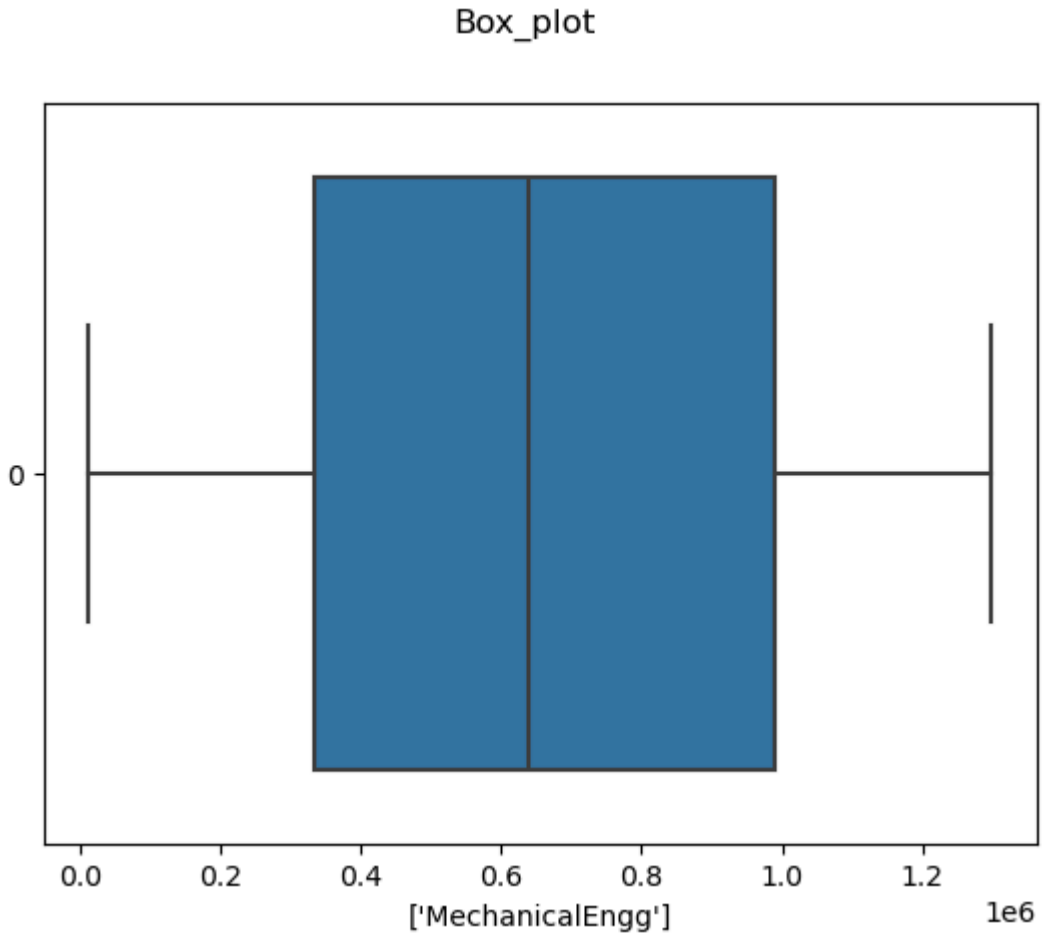

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['Quant']
1e6

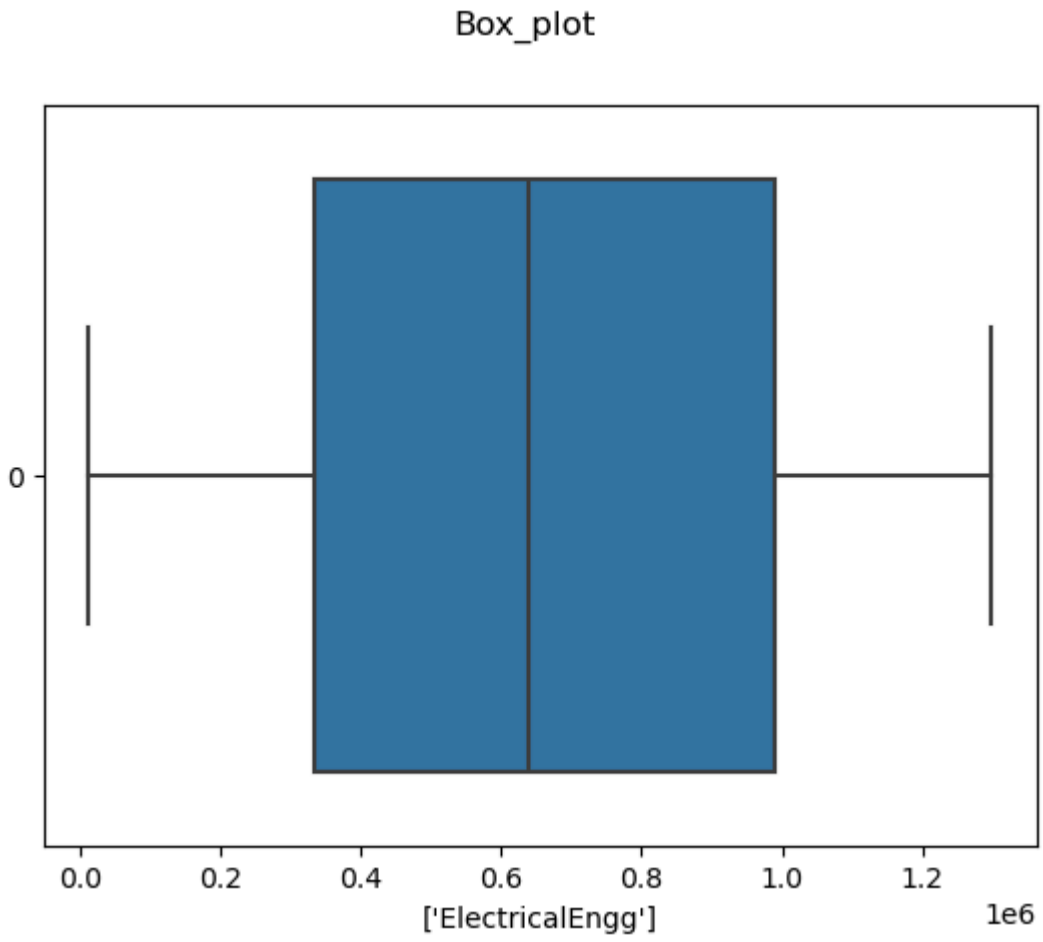

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['Domain']
1e6

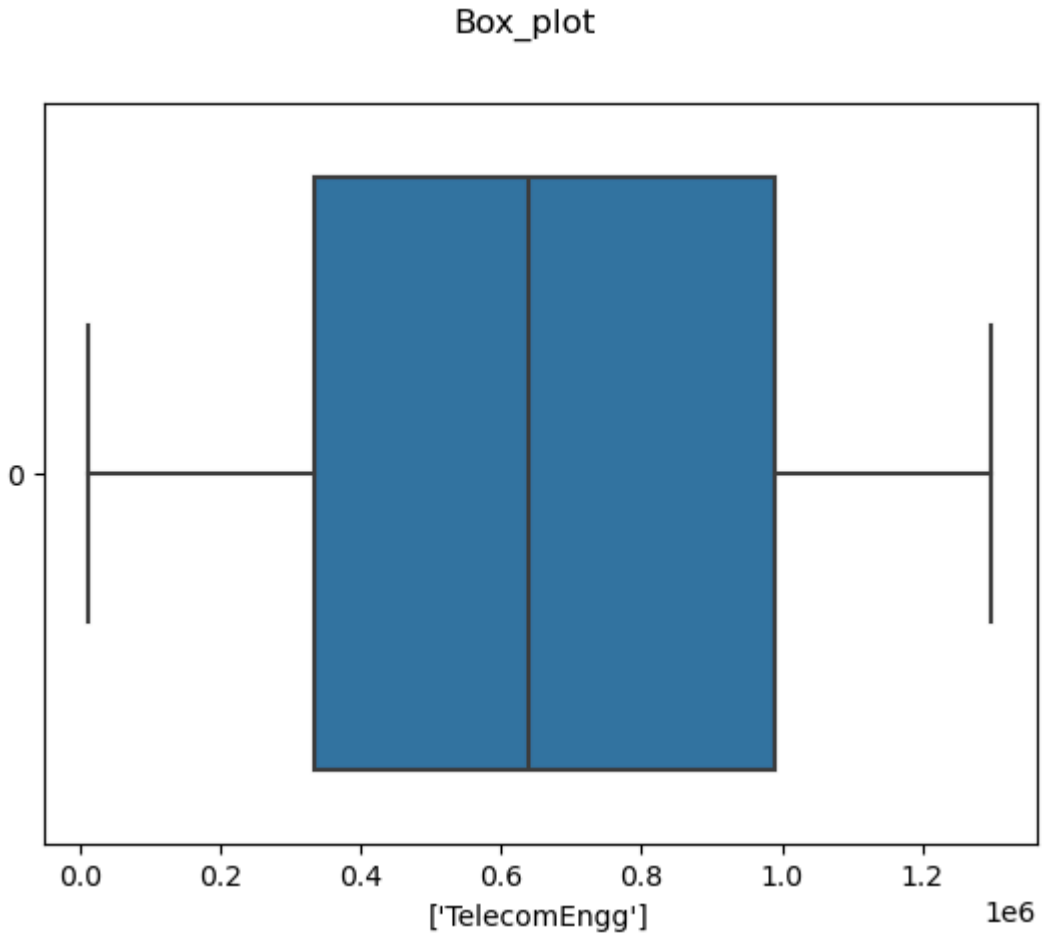

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['ComputerProgramming']
1e6

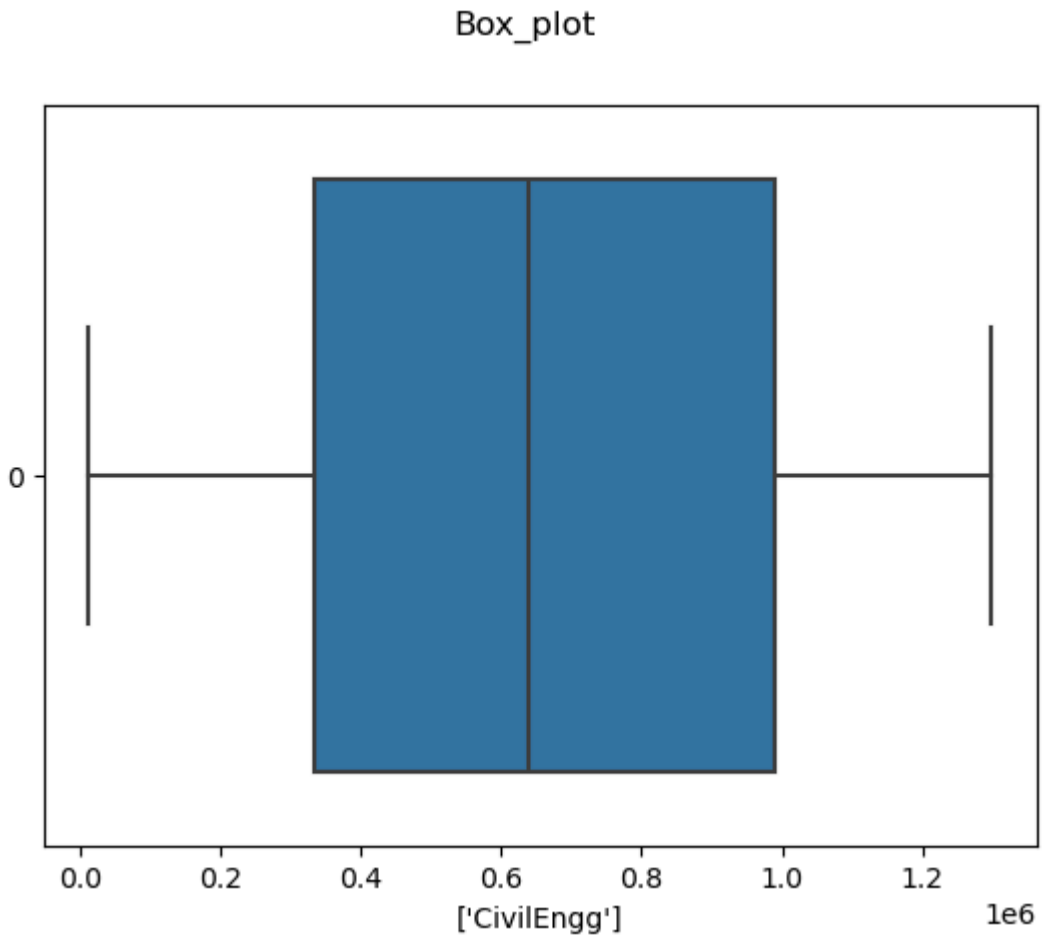

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['ElectronicsAndSemicon']
1e6

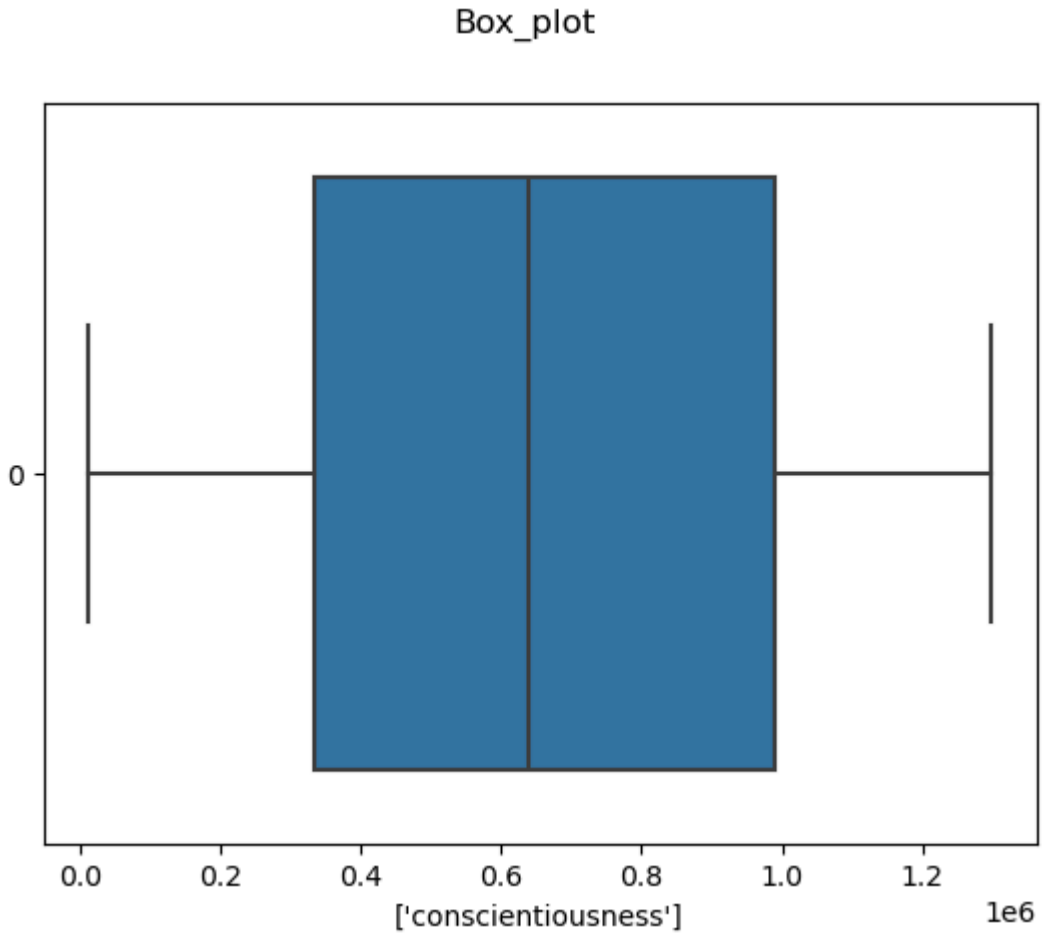

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['ComputerScience']
1e6

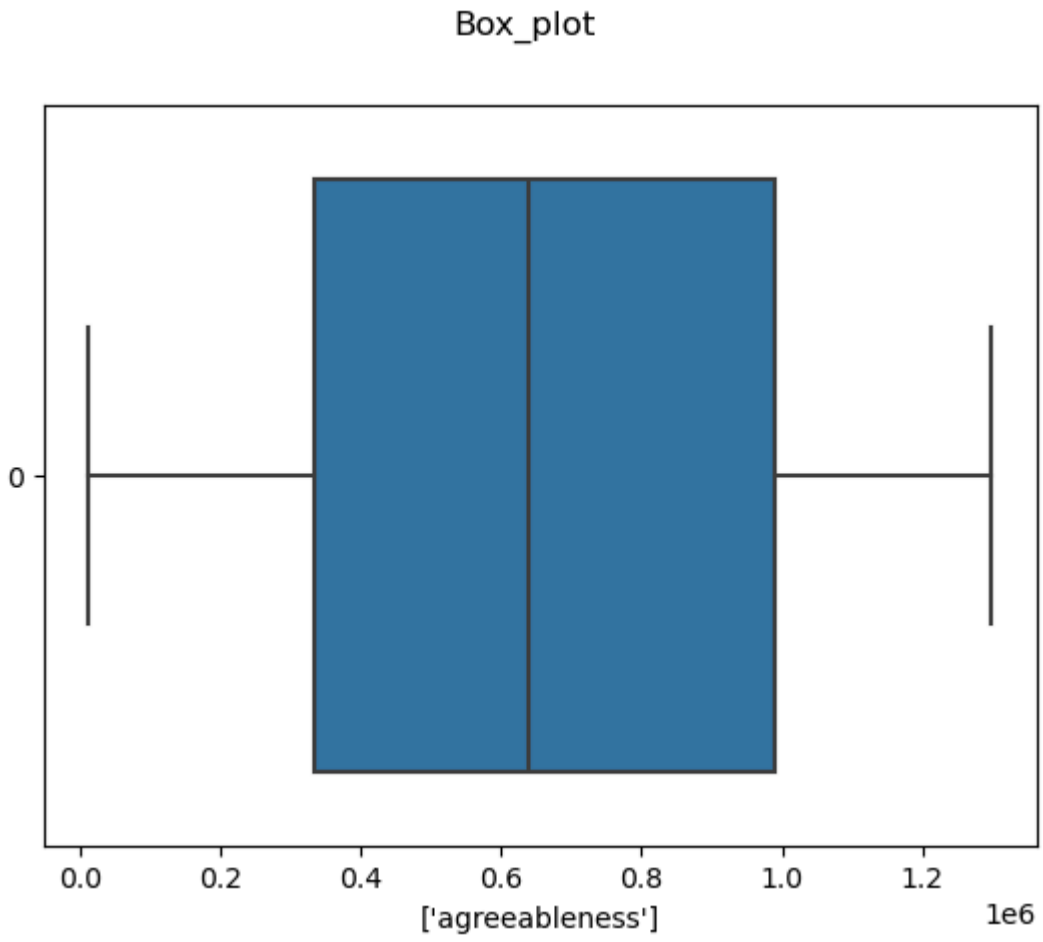

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['MechanicalEngg']
1e6

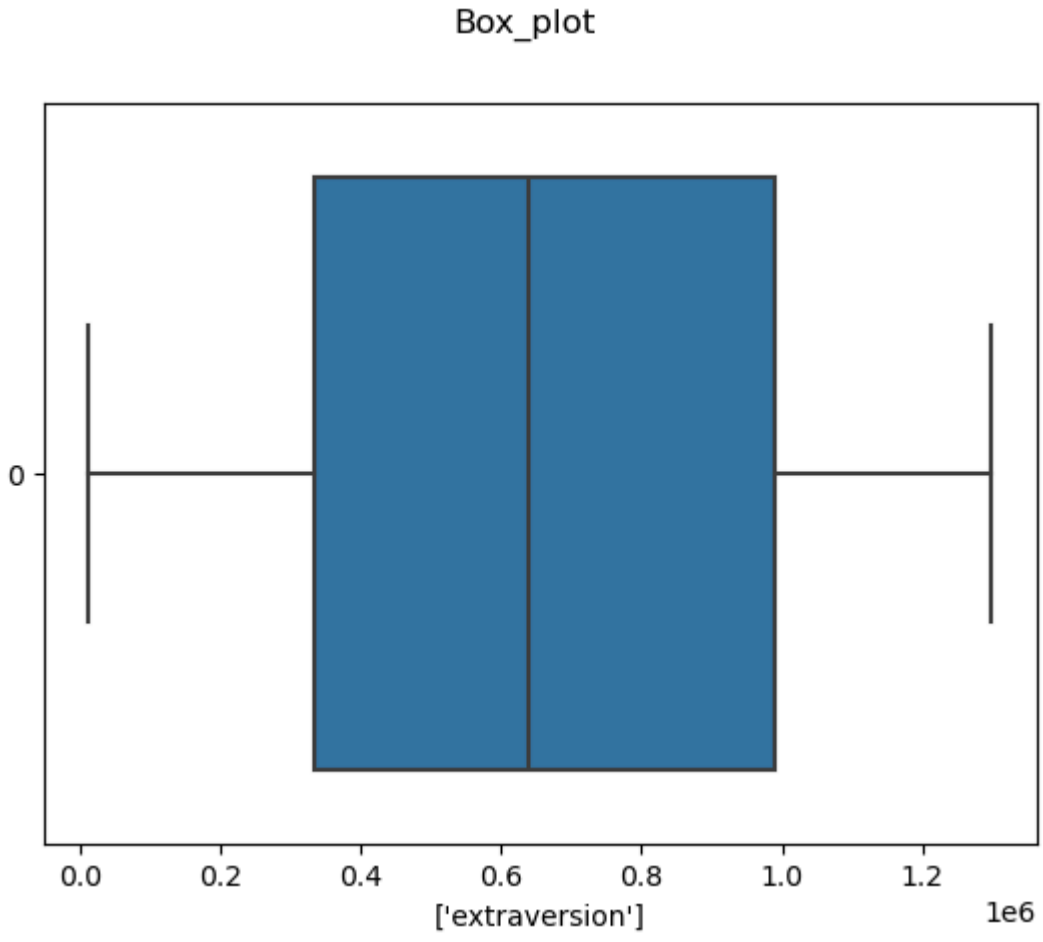

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['ElectricalEngg']
1e6

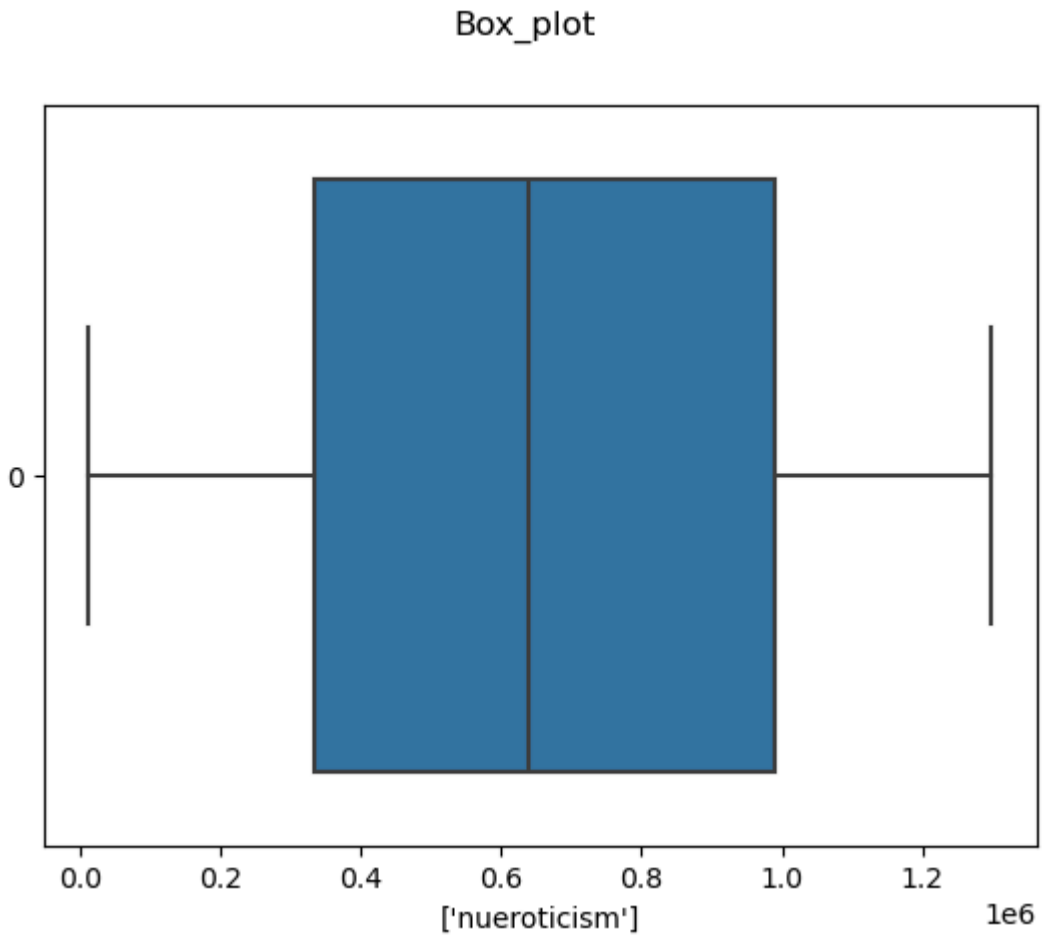

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
['TelecomEngg']
1e6

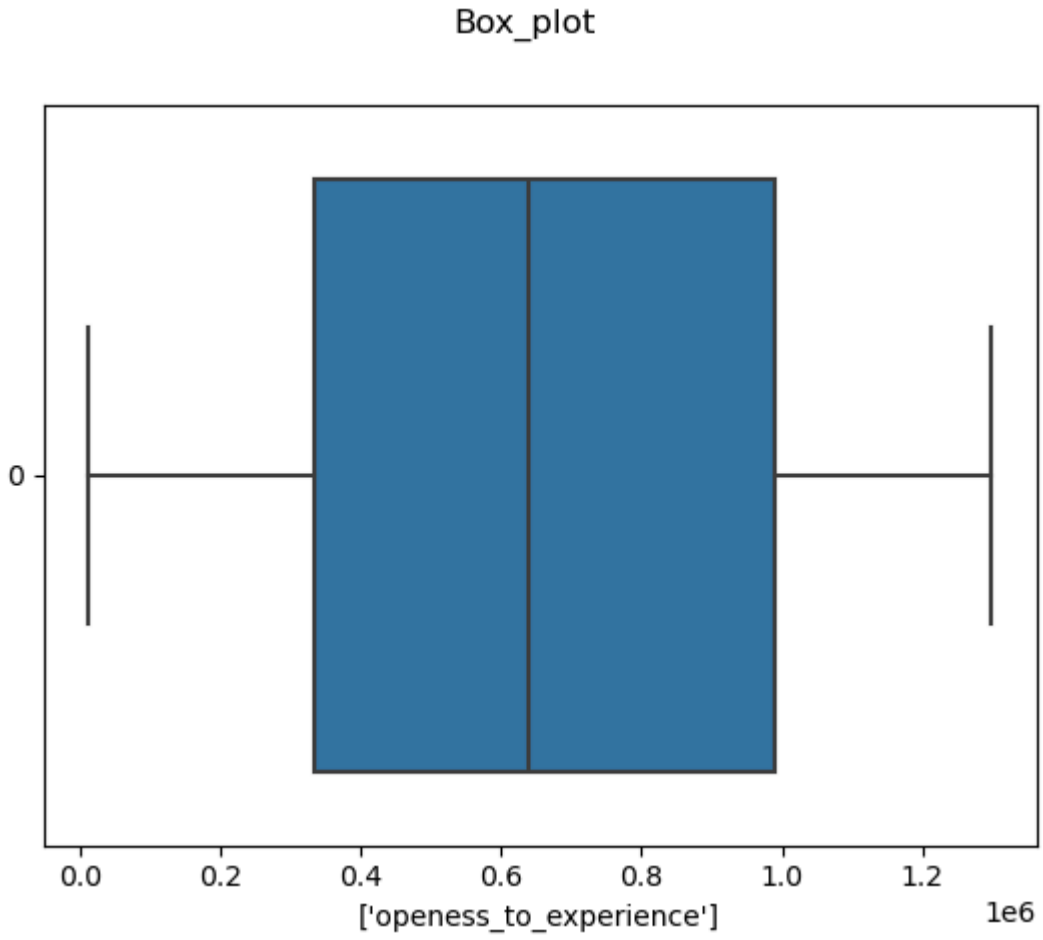

Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
['CivilEngg']


Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
['conscientiousness']


Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
['agreeableness']


Box_plot
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
['extraversion']





In [62]:
```
len(non_outliers_df)
```

Out[62]: 3903

The difference between actaul dataframe without removing outliers and after removing outliers

In [64]:
```
count=len(df)-len(non_outliers_df)
print(f"The total number of outliers are :{count}")
```

```
The total number of outliers are :95
```

Data without any outliers and missing values

In [72]:
```
non_outliers_df
```

Out[72]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	G
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	

3903 rows × 39 columns

In [ ]: