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

file_path= r'C:\Users\P.DEEPIKA\Downloads\data.xlsx'

data=pd.read_excel(file_path)
```

In [ ]:
```
data
```

Out[ ]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gen
0	train	203097	420000	2012-06-01	present	senior quality engineer	Bangalore	
1	train	579905	500000	2013-09-01	present	assistant manager	Indore	
2	train	810601	325000	2014-06-01	present	systems engineer	Chennai	
3	train	267447	1100000	2011-07-01	present	senior software engineer	Gurgaon	
4	train	343523	200000	2014-03-01	2015-03-01 00:00:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000	2011-10-01	2012-10-01 00:00:00	software engineer	New Delhi	
3994	train	752781	100000	2013-07-01	2013-07-01 00:00:00	technical writer	Hyderabad	
3995	train	355888	320000	2013-07-01	present	associate software engineer	Bangalore	
3996	train	947111	200000	2014-07-01	2015-01-01 00:00:00	software developer	Asifabadbanglore	
3997	train	324966	400000	2013-02-01	present	senior systems engineer	Chennai	

3998 rows × 39 columns

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

Out[ ]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB
0	train	203097	420000	2012-06-01	present	senior quality engineer	Bangalore	f	1990-02-19
1	train	579905	500000	2013-09-01	present	assistant manager	Indore	m	1989-10-04
2	train	810601	325000	2014-06-01	present	systems engineer	Chennai	f	1992-08-03
3	train	267447	1100000	2011-07-01	present	senior software engineer	Gurgaon	m	1989-12-05
4	train	343523	200000	2014-03-01	2015-03-01 00:00:00	get	Manesar	m	1991-02-27

5 rows × 39 columns

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

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

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

Out[ ]:

	ID	Salary	DOJ	DOB	10percentage
count	3.998000e+03	3.998000e+03	3998	3998	3998.000000
mean	6.637945e+05	3.076998e+05	2013-07-02 11:04:10.325162496	1990-12-06 06:01:15.637819008	77.925443
min	1.124400e+04	3.500000e+04	1991-06-01 00:00:00	1977-10-30 00:00:00	43.000000
25%	3.342842e+05	1.800000e+05	2012-10-01 00:00:00	1989-11-16 06:00:00	71.680000
50%	6.396000e+05	3.000000e+05	2013-11-01 00:00:00	1991-03-07 12:00:00	79.150000
75%	9.904800e+05	3.700000e+05	2014-07-01 00:00:00	1992-03-13 18:00:00	85.670000
max	1.298275e+06	4.000000e+06	2015-12-01 00:00:00	1997-05-27 00:00:00	97.760000
std	3.632182e+05	2.127375e+05	NaN	NaN	9.850162

8 rows × 29 columns

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3998 non-null   object
 1   ID                     3998 non-null   int64
 2   Salary                 3998 non-null   int64
 3   DOJ                    3998 non-null   datetime64[ns]
 4   DOL                    3998 non-null   object
 5   Designation            3998 non-null   object
 6   JobCity                3998 non-null   object
 7   Gender                 3998 non-null   object
 8   DOB                    3998 non-null   datetime64[ns]
 9   10percentage           3998 non-null   float64
 10  10board                3998 non-null   object
 11  12graduation           3998 non-null   int64
 12  12percentage           3998 non-null   float64
 13  12board                3998 non-null   object
 14  CollegeID              3998 non-null   int64
 15  CollegeTier            3998 non-null   int64
 16  Degree                 3998 non-null   object
 17  Specialization         3998 non-null   object
 18  collegeGPA             3998 non-null   float64
 19  CollegeCityID          3998 non-null   int64
 20  CollegeCityTier        3998 non-null   int64
 21  CollegeState           3998 non-null   object
 22  GraduationYear         3998 non-null   int64
 23  English                3998 non-null   int64
 24  Logical                3998 non-null   int64
 25  Quant                  3998 non-null   int64
 26  Domain                 3998 non-null   float64
 27  ComputerProgramming    3998 non-null   int64
 28  ElectronicsAndSemicon  3998 non-null   int64
 29  ComputerScience        3998 non-null   int64
 30  MechanicalEngg         3998 non-null   int64
 31  ElectricalEngg         3998 non-null   int64
 32  TelecomEngg            3998 non-null   int64
 33  CivilEngg              3998 non-null   int64
 34  conscientiousness      3998 non-null   float64
 35  agreeableness          3998 non-null   float64
 36  extraversion           3998 non-null   float64
 37  nueroticism            3998 non-null   float64
 38  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](2), float64(9), int64(18), object(10)
memory usage: 1.2+ MB
```

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

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

In [ ]:
```
data["DOL"].unique()
```

Out[ ]:
```
array(['present', datetime.datetime(2015, 3, 1, 0, 0),
       datetime.datetime(2015, 5, 1, 0, 0),
       datetime.datetime(2015, 7, 1, 0, 0),
       datetime.datetime(2015, 4, 1, 0, 0),
       datetime.datetime(2014, 10, 1, 0, 0),
       datetime.datetime(2014, 9, 1, 0, 0),
       datetime.datetime(2014, 6, 1, 0, 0),
       datetime.datetime(2012, 9, 1, 0, 0),
       datetime.datetime(2013, 12, 1, 0, 0),
       datetime.datetime(2015, 6, 1, 0, 0),
       datetime.datetime(2013, 10, 1, 0, 0),
       datetime.datetime(2015, 1, 1, 0, 0),
       datetime.datetime(2014, 4, 1, 0, 0),
       datetime.datetime(2013, 6, 1, 0, 0),
       datetime.datetime(2012, 3, 1, 0, 0),
       datetime.datetime(2014, 7, 1, 0, 0),
       datetime.datetime(2013, 2, 1, 0, 0),
       datetime.datetime(2014, 1, 1, 0, 0),
       datetime.datetime(2013, 4, 1, 0, 0),
       datetime.datetime(2012, 7, 1, 0, 0),
       datetime.datetime(2014, 5, 1, 0, 0),
       datetime.datetime(2013, 9, 1, 0, 0),
       datetime.datetime(2015, 2, 1, 0, 0),
       datetime.datetime(2012, 1, 1, 0, 0),
       datetime.datetime(2015, 8, 1, 0, 0),
       datetime.datetime(2014, 8, 1, 0, 0),
       datetime.datetime(2015, 12, 1, 0, 0),
       datetime.datetime(2014, 12, 1, 0, 0),
       datetime.datetime(2012, 5, 1, 0, 0),
       datetime.datetime(2011, 3, 1, 0, 0),
       datetime.datetime(2011, 7, 1, 0, 0),
       datetime.datetime(2014, 2, 1, 0, 0),
       datetime.datetime(2011, 12, 1, 0, 0),
       datetime.datetime(2015, 10, 1, 0, 0),
       datetime.datetime(2014, 11, 1, 0, 0),
       datetime.datetime(2014, 3, 1, 0, 0),
       datetime.datetime(2011, 11, 1, 0, 0),
       datetime.datetime(2013, 5, 1, 0, 0),
       datetime.datetime(2013, 7, 1, 0, 0),
       datetime.datetime(2013, 11, 1, 0, 0),
       datetime.datetime(2011, 1, 1, 0, 0),
       datetime.datetime(2011, 5, 1, 0, 0),
       datetime.datetime(2012, 2, 1, 0, 0),
       datetime.datetime(2012, 11, 1, 0, 0),
       datetime.datetime(2012, 6, 1, 0, 0),
       datetime.datetime(2013, 8, 1, 0, 0),
       datetime.datetime(2005, 3, 1, 0, 0),
       datetime.datetime(2013, 3, 1, 0, 0),
       datetime.datetime(2012, 10, 1, 0, 0),
       datetime.datetime(2011, 2, 1, 0, 0),
       datetime.datetime(2010, 2, 1, 0, 0),
       datetime.datetime(2013, 1, 1, 0, 0),
       datetime.datetime(2011, 6, 1, 0, 0),
       datetime.datetime(2015, 9, 1, 0, 0),
       datetime.datetime(2012, 4, 1, 0, 0),
       datetime.datetime(2012, 8, 1, 0, 0),
       datetime.datetime(2011, 4, 1, 0, 0),
       datetime.datetime(2011, 10, 1, 0, 0),
       datetime.datetime(2015, 11, 1, 0, 0),
       datetime.datetime(2012, 12, 1, 0, 0),
```

```
       datetime.datetime(2011, 9, 1, 0, 0),
       datetime.datetime(2010, 8, 1, 0, 0),
       datetime.datetime(2011, 8, 1, 0, 0),
       datetime.datetime(2009, 6, 1, 0, 0),
       datetime.datetime(2008, 3, 1, 0, 0),
       datetime.datetime(2010, 10, 1, 0, 0)], dtype=object)
```

In [ ]:
```
data["ComputerScience"].unique()
```

Out[ ]:
```
array([ -1, 407, 346, 376, 500, 438, 315, 253, 469, 192, 530, 284, 223,
       561, 684, 592, 623, 653, 130, 715])
```

In [ ]:
```
data=data.replace(-1,0)
```

In [ ]:
```
data["ComputerScience"].unique()
```

Out[ ]:
```
array([  0, 407, 346, 376, 500, 438, 315, 253, 469, 192, 530, 284, 223,
       561, 684, 592, 623, 653, 130, 715])
```

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3998 non-null   object
 1   ID                     3998 non-null   int64
 2   Salary                 3998 non-null   int64
 3   DOJ                    3998 non-null   datetime64[ns]
 4   DOL                    3998 non-null   object
 5   Designation            3998 non-null   object
 6   JobCity                3998 non-null   object
 7   Gender                 3998 non-null   object
 8   DOB                    3998 non-null   datetime64[ns]
 9   10percentage           3998 non-null   float64
 10  10board                3998 non-null   object
 11  12graduation           3998 non-null   int64
 12  12percentage           3998 non-null   float64
 13  12board                3998 non-null   object
 14  CollegeID              3998 non-null   int64
 15  CollegeTier            3998 non-null   int64
 16  Degree                 3998 non-null   object
 17  Specialization         3998 non-null   object
 18  collegeGPA             3998 non-null   float64
 19  CollegeCityID          3998 non-null   int64
 20  CollegeCityTier        3998 non-null   int64
 21  CollegeState           3998 non-null   object
 22  GraduationYear         3998 non-null   int64
 23  English                3998 non-null   int64
 24  Logical                3998 non-null   int64
 25  Quant                  3998 non-null   int64
 26  Domain                 3998 non-null   float64
 27  ComputerProgramming    3998 non-null   int64
 28  ElectronicsAndSemicon  3998 non-null   int64
 29  ComputerScience        3998 non-null   int64
 30  MechanicalEngg         3998 non-null   int64
 31  ElectricalEngg         3998 non-null   int64
 32  TelecomEngg            3998 non-null   int64
 33  CivilEngg              3998 non-null   int64
 34  conscientiousness      3998 non-null   float64
 35  agreeableness          3998 non-null   float64
 36  extraversion           3998 non-null   float64
 37  nueroticism            3998 non-null   float64
 38  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](2), float64(9), int64(18), object(10)
memory usage: 1.2+ MB
```

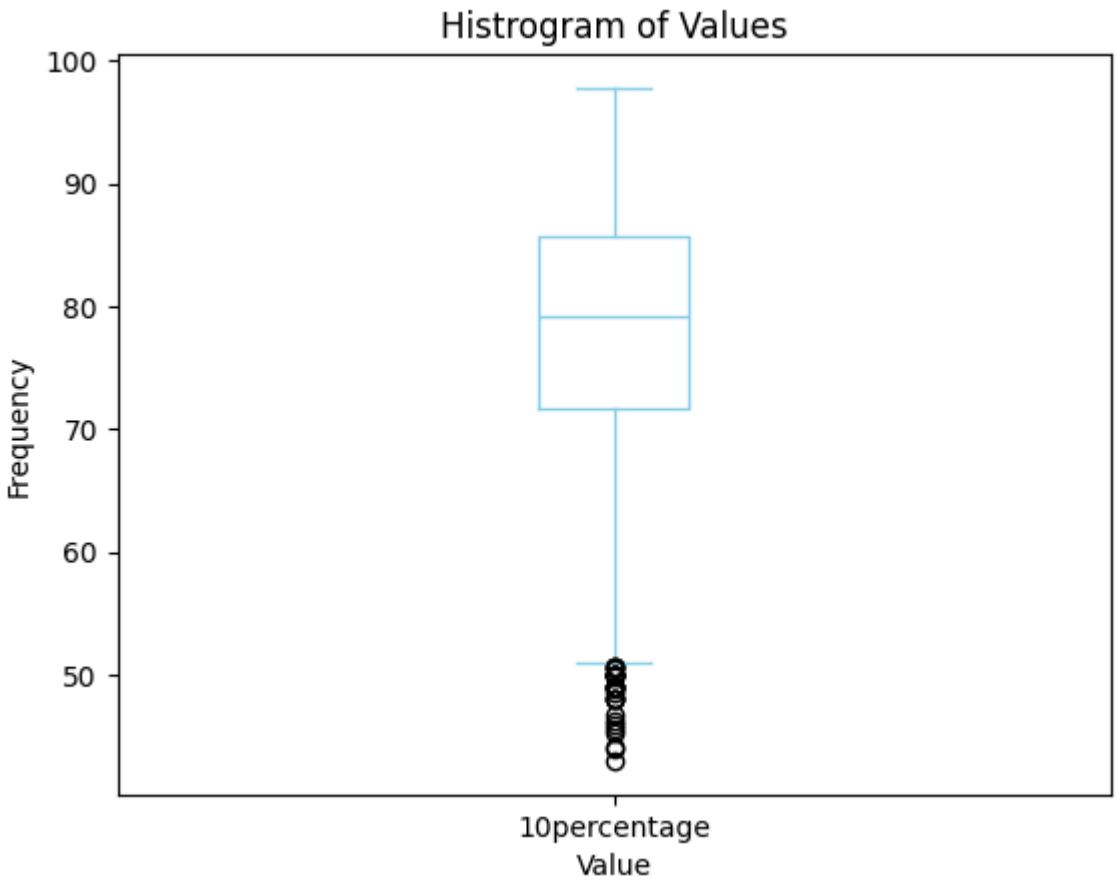
In [ ]:
```
num=['Salary','10percentage','12percentage']
```

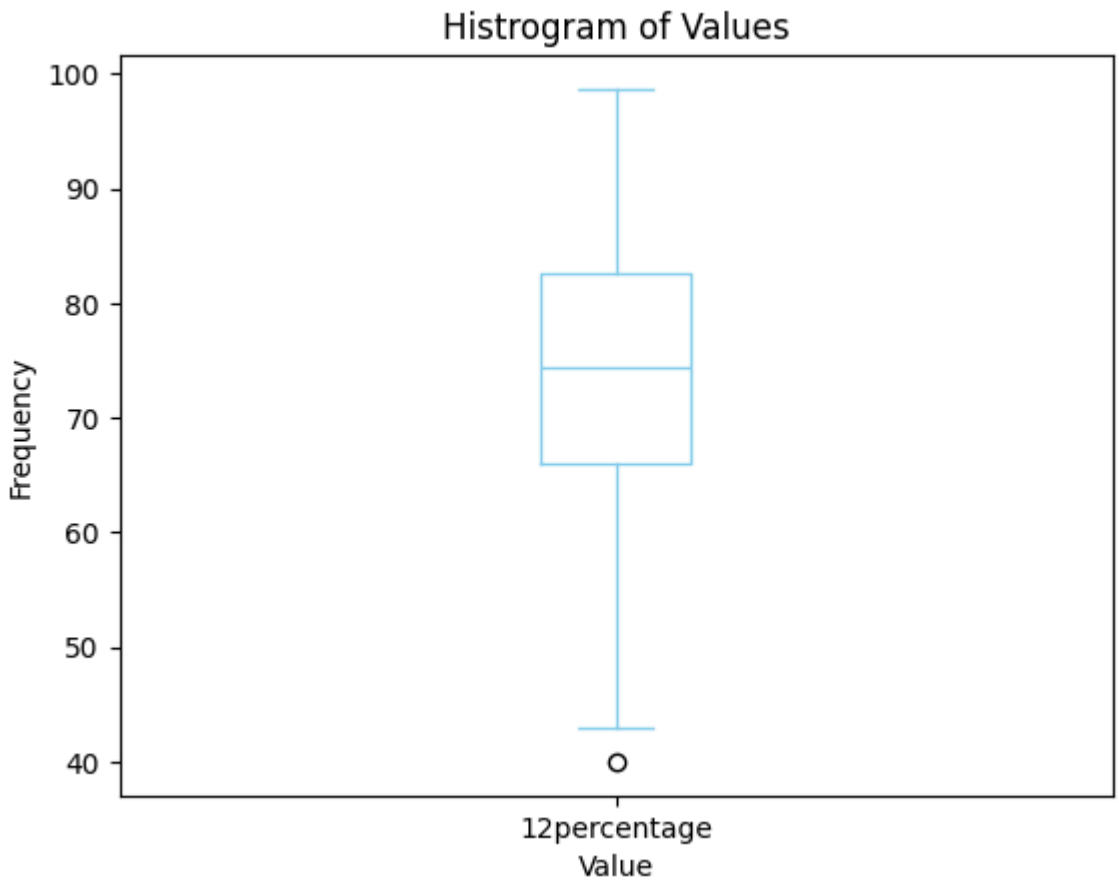
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** Salary **********
min       3.500000e+04
max       4.000000e+06
mean      3.076998e+05
median    3.000000e+05
std       2.127375e+05
skew      6.451081e+00
kurt      8.093000e+01
Name: Salary, dtype: float64

********** 10percentage **********
min       43.000000
max       97.760000
mean      77.925443
median    79.150000
std        9.850162
skew      -0.591019
kurt      -0.110284
Name: 10percentage, dtype: float64

********** 12percentage **********
min       40.000000
max       98.700000
mean      74.466366
median    74.400000
std       10.999933
skew      -0.032607
kurt      -0.630737
Name: 12percentage, dtype: float64
```

In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```


Histrogram of Values
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
Salary
Value


Histrogram of Values
100
90
80
70
60
50
Frequency
10percentage
Value

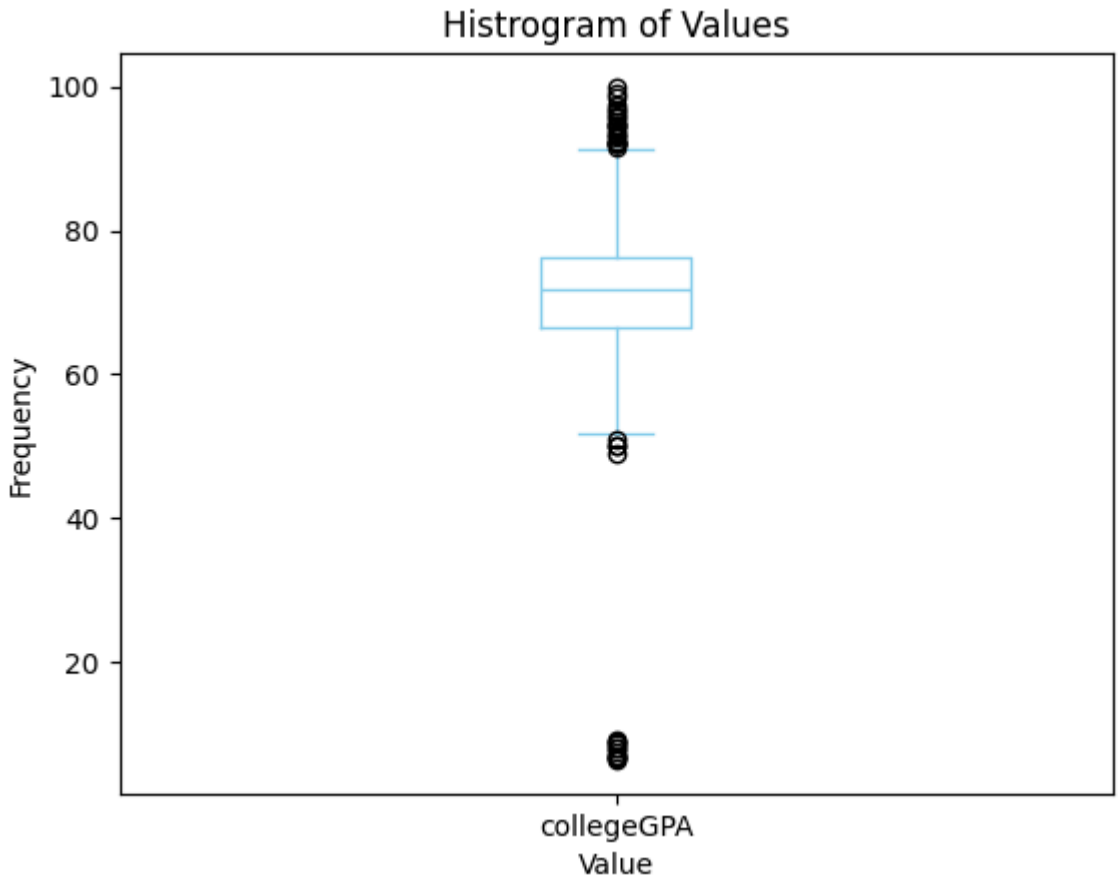


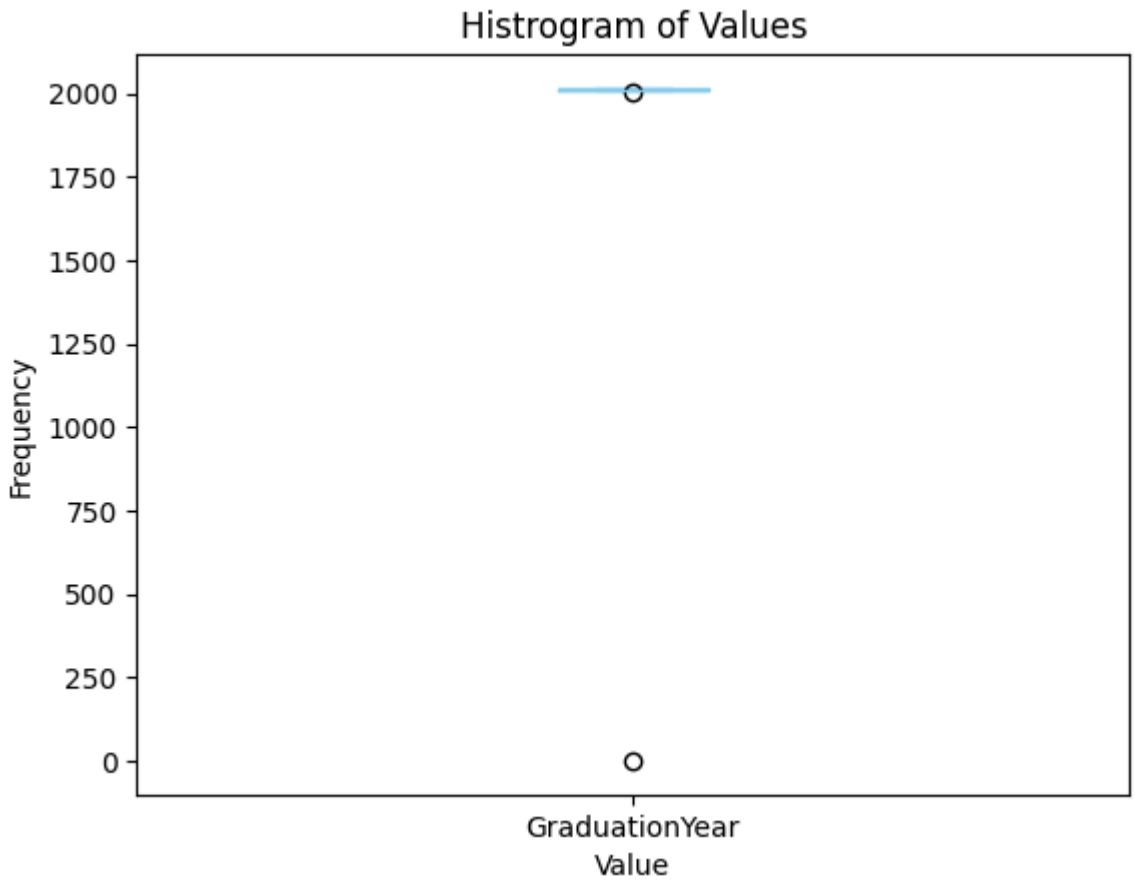
In [ ]:
```
num=['collegeGPA','GraduationYear','English']
```

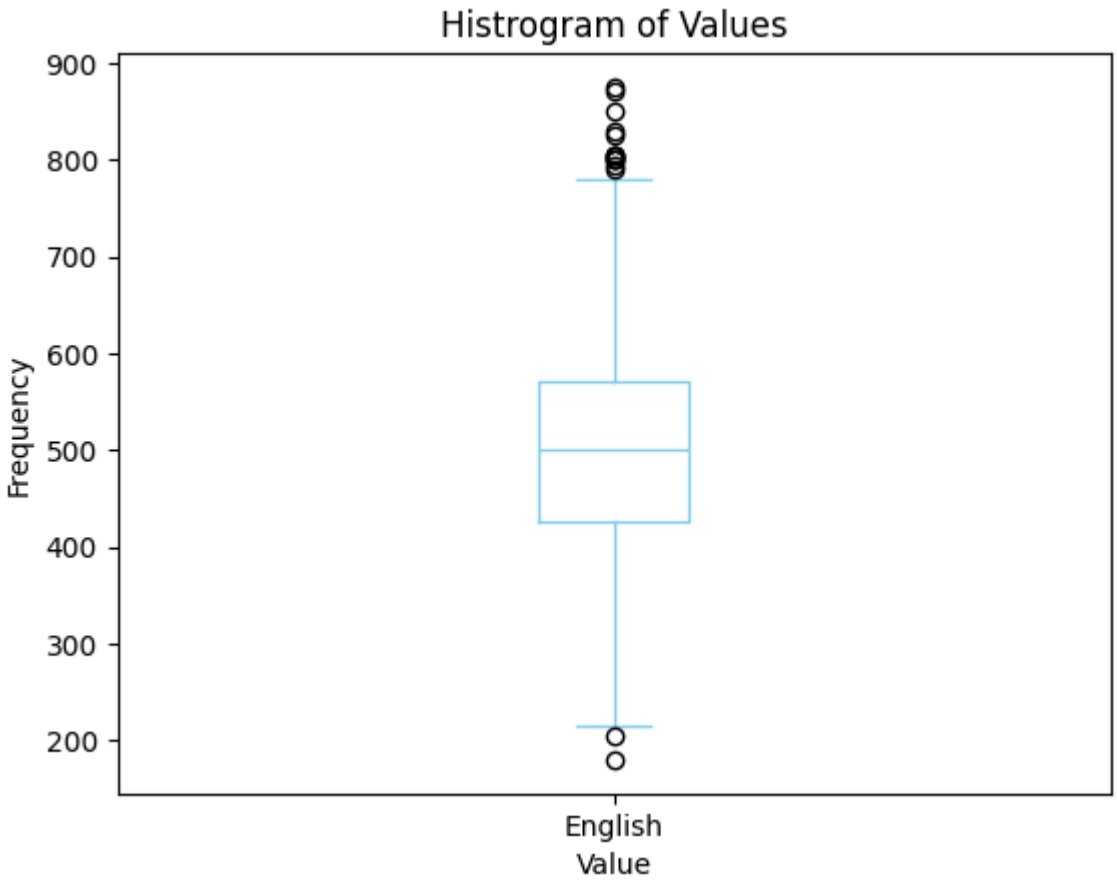
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** collegeGPA **********
min        6.450000
max       99.930000
mean      71.486171
median    71.720000
std        8.167338
skew      -1.249209
kurt      10.234244
Name: collegeGPA, dtype: float64

********** GraduationYear **********
min          0.000000
max       2017.000000
mean      2012.105803
median    2013.000000
std         31.857271
skew       -63.068064
kurt      3984.369696
Name: GraduationYear, dtype: float64

********** English **********
min       180.000000
max       875.000000
mean      501.649075
median    500.000000
std       104.940021
skew        0.191997
kurt       -0.254133
Name: English, dtype: float64
```

In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```


Histrogram of Values
Frequency
100
80
60
40
20
collegeGPA
Value


Histrogram of Values
Frequency
2000
1750
1500
1250
1000
750
500
250
0
GraduationYear
Value

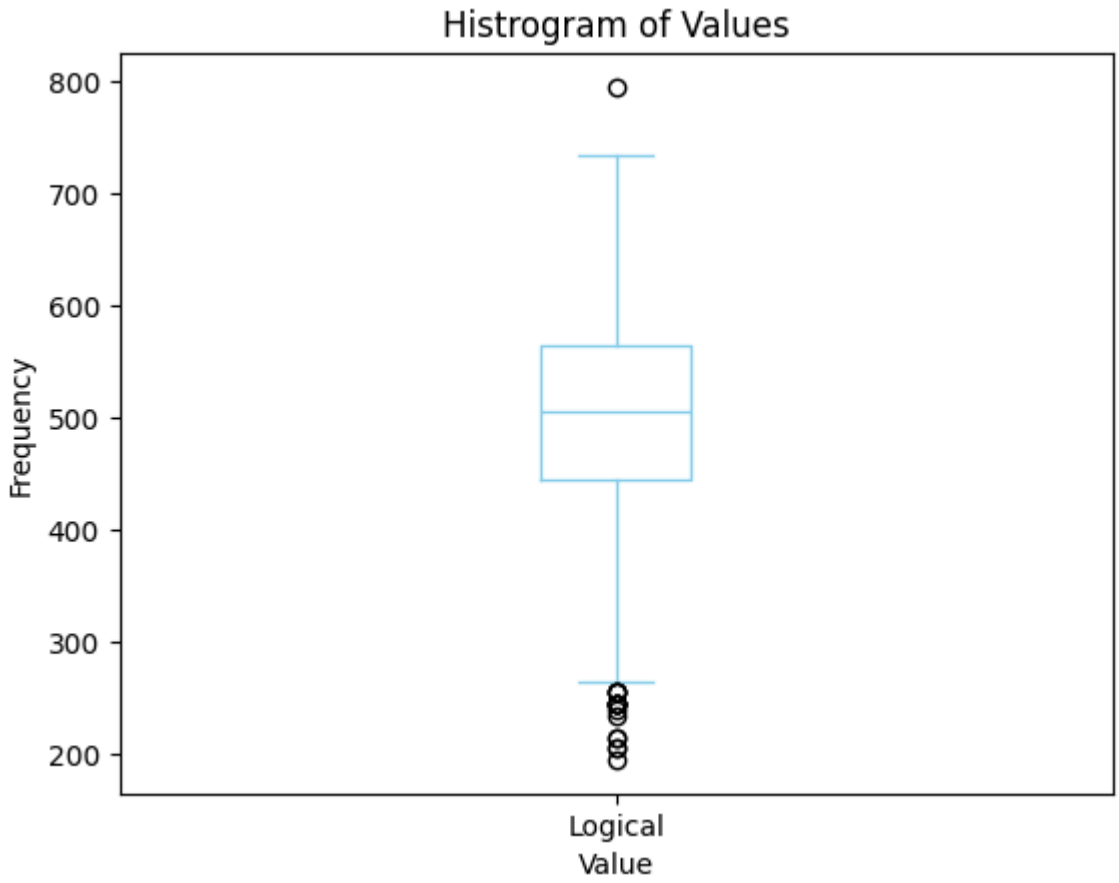


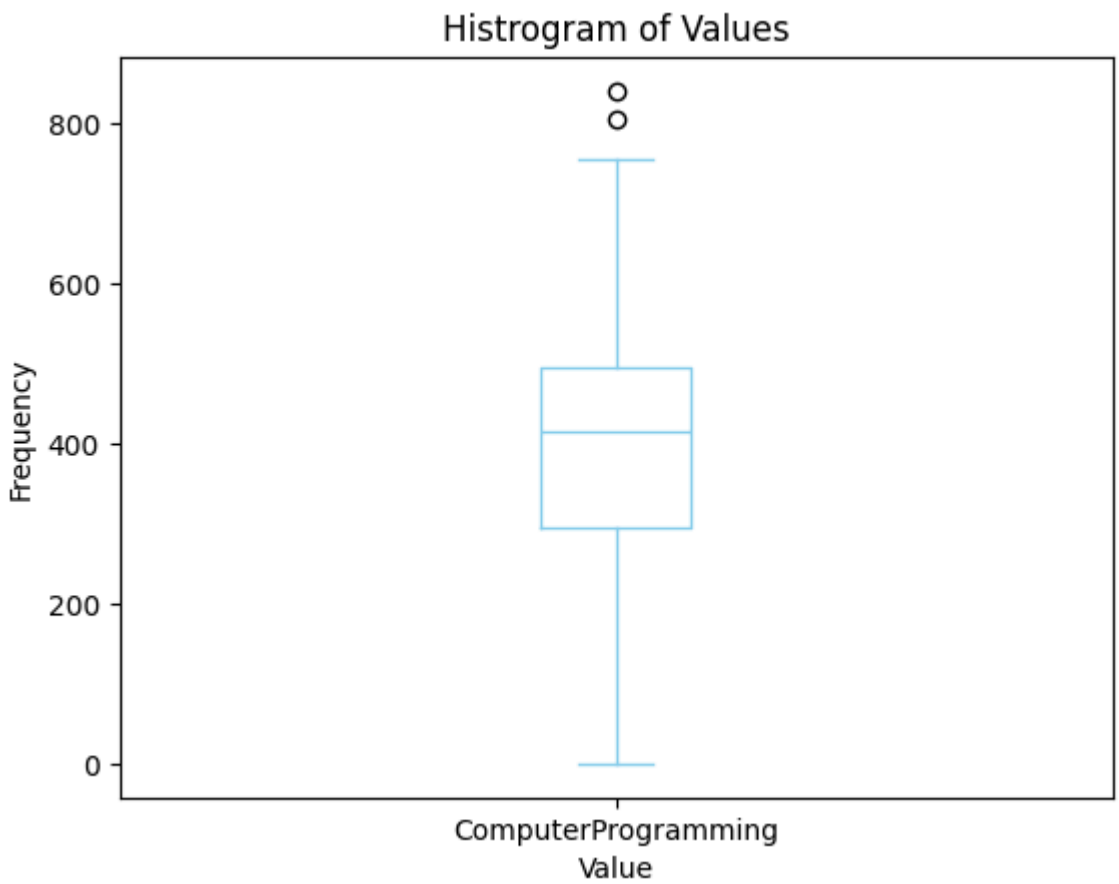
In [ ]:
```
num=['Logical','ComputerProgramming','ElectronicsAndSemicon']
```

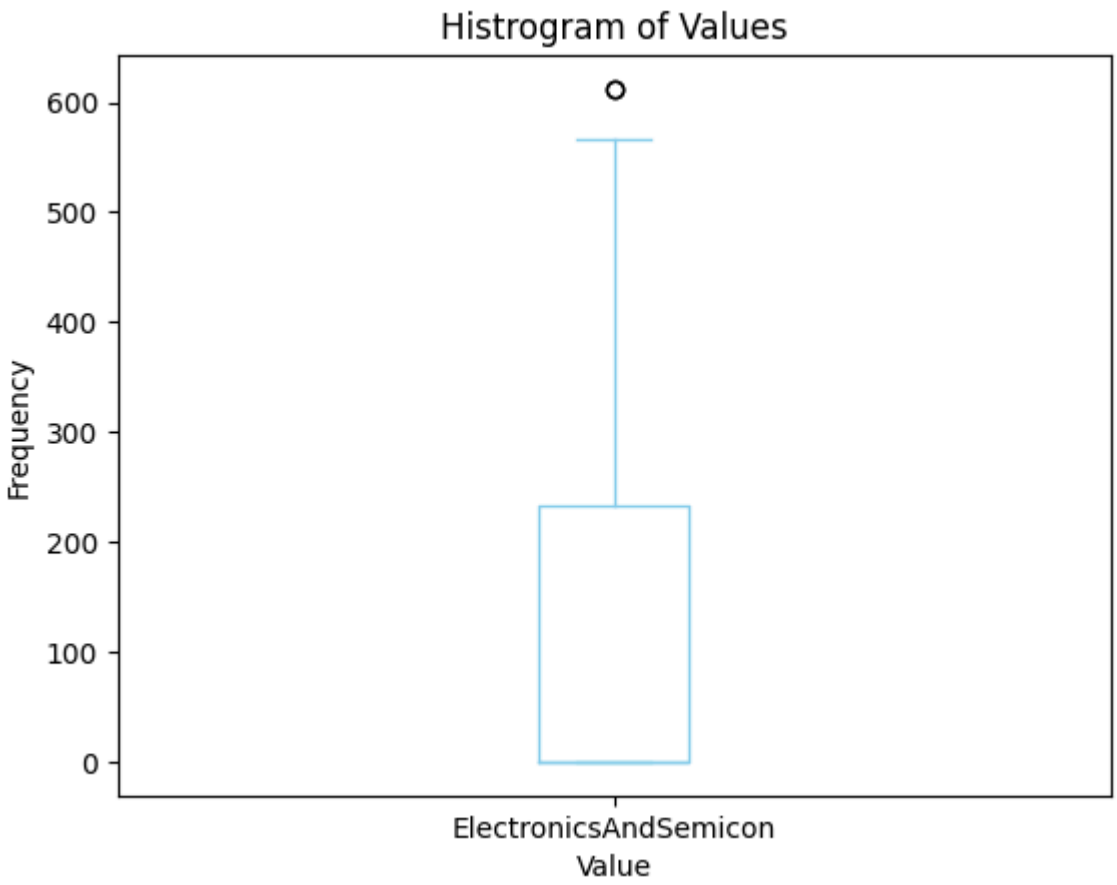
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** Logical **********
min       195.000000
max       795.000000
mean      501.598799
median    505.000000
std        86.783297
skew       -0.216602
kurt       -0.224761
Name: Logical, dtype: float64

********** ComputerProgramming **********
min         0.000000
max       840.000000
mean      353.319910
median    415.000000
std       204.981129
skew       -0.776093
kurt       -0.667741
Name: ComputerProgramming, dtype: float64

********** ElectronicsAndSemicon **********
min         0.000000
max       612.000000
mean       96.042271
median      0.000000
std       157.806602
skew        1.197314
kurt       -0.205359
Name: ElectronicsAndSemicon, dtype: float64
```

In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```


Histrogram of Values
800
700
600
500
400
300
200
Frequency
Logical
Value


Histrogram of Values
800
600
400
200
0
Frequency
ComputerProgramming
Value



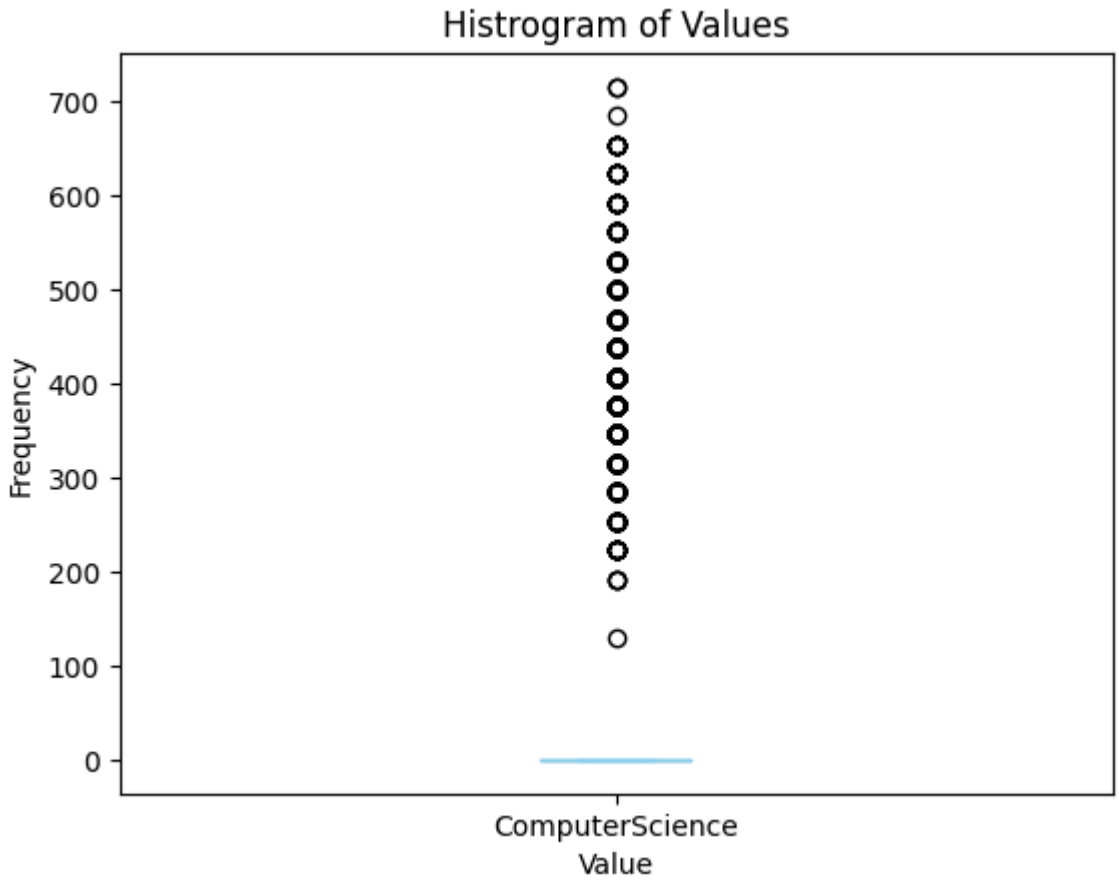
In [ ]:
```
num=['ComputerScience','MechanicalEngg','ElectricalEngg']
```

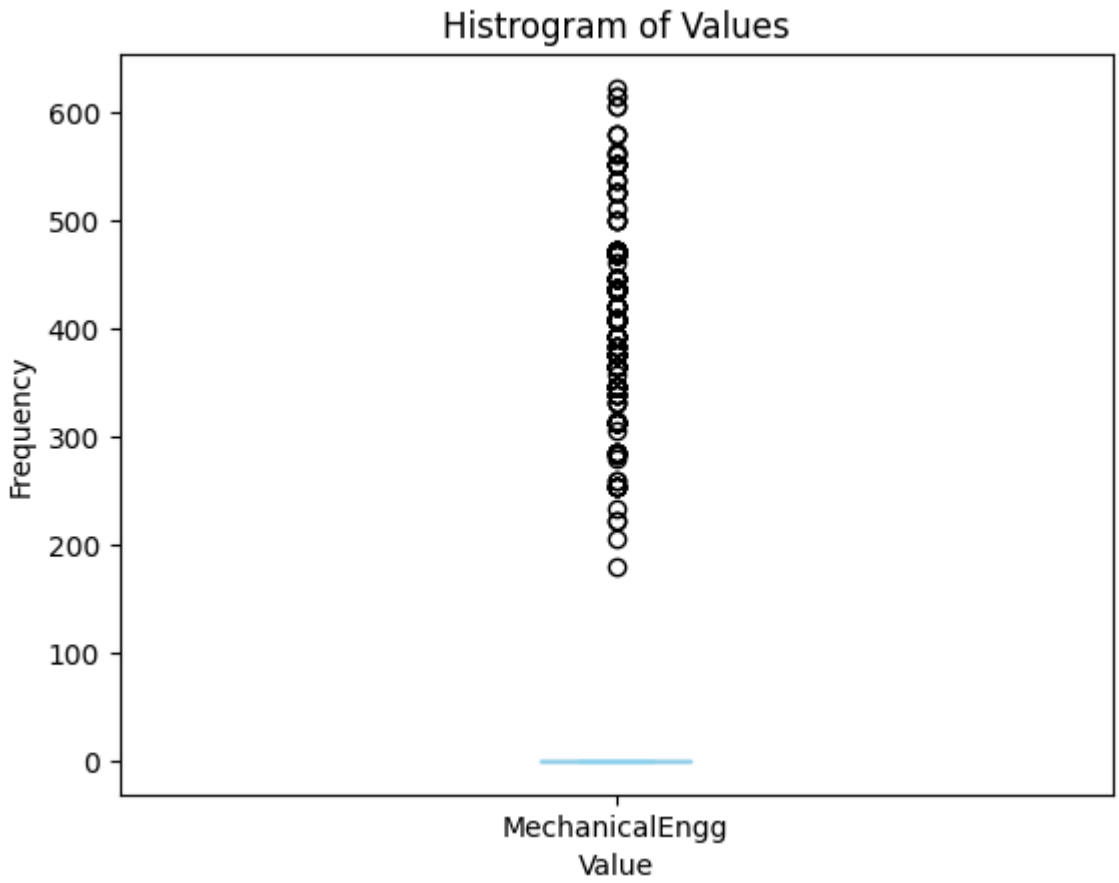
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

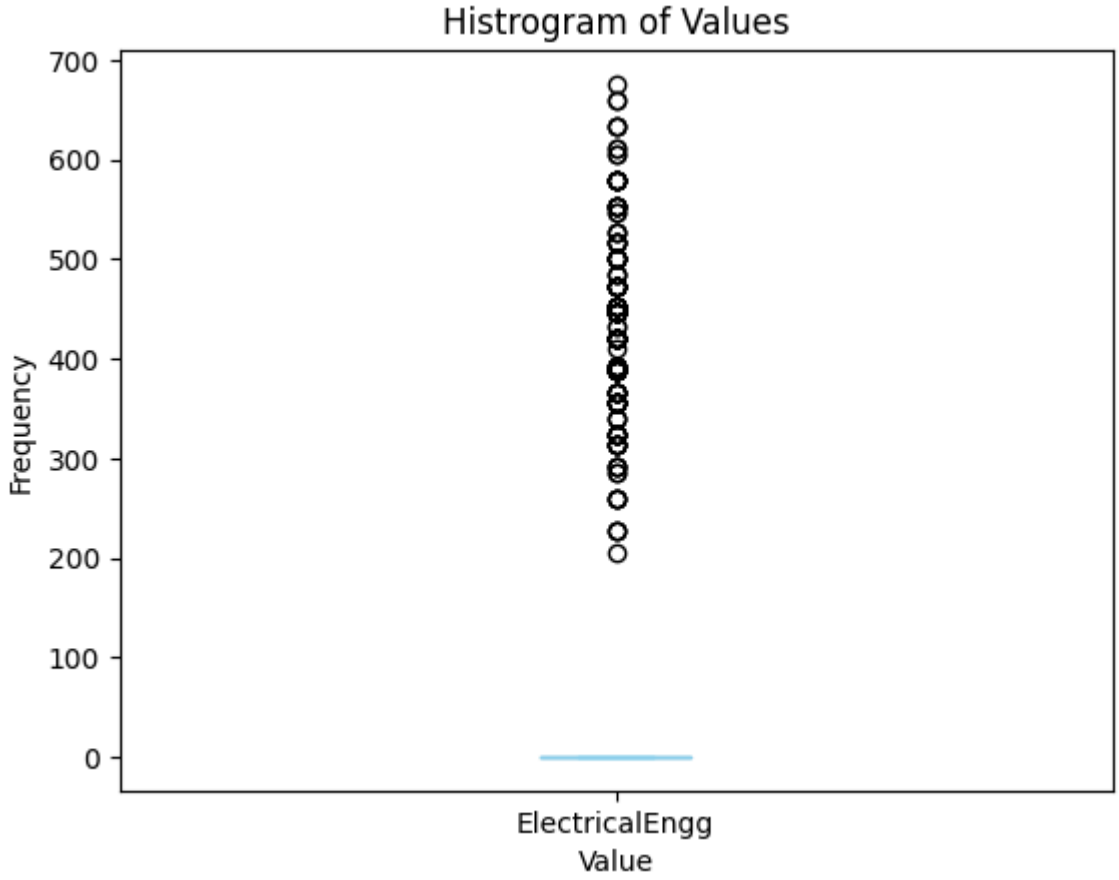
```
********** ComputerScience **********
min         0.000000
max       715.000000
mean       91.516758
median      0.000000
std       174.867677
skew        1.530511
kurt        0.697248
Name: ComputerScience, dtype: float64

********** MechanicalEngg **********
min         0.000000
max       623.000000
mean       23.915958
median      0.000000
std        97.893295
skew        4.030835
kurt       15.032665
Name: MechanicalEngg, dtype: float64

********** ElectricalEngg **********
min         0.000000
max       676.000000
mean       17.438469
median      0.000000
std        87.394072
skew        5.062039
kurt       24.899819
Name: ElectricalEngg, dtype: float64

********** TelecomEngg **********
min         0.000000
max       548.000000
mean       32.757629
median      0.000000
std       104.568796
skew        3.042584
kurt        7.821100
Name: TelecomEngg, dtype: float64

********** CivilEngg **********
min         0.000000
max       516.000000
mean        3.673337
median      0.000000
std        36.559052
skew       10.319461
kurt      109.142713
Name: CivilEngg, dtype: float64

********** conscientiousness **********
min      -4.126700
max       1.995300
mean     -0.037831
median    0.046400
std       1.028666
skew     -0.527003
kurt      0.122596
Name: conscientiousness, dtype: float64
```

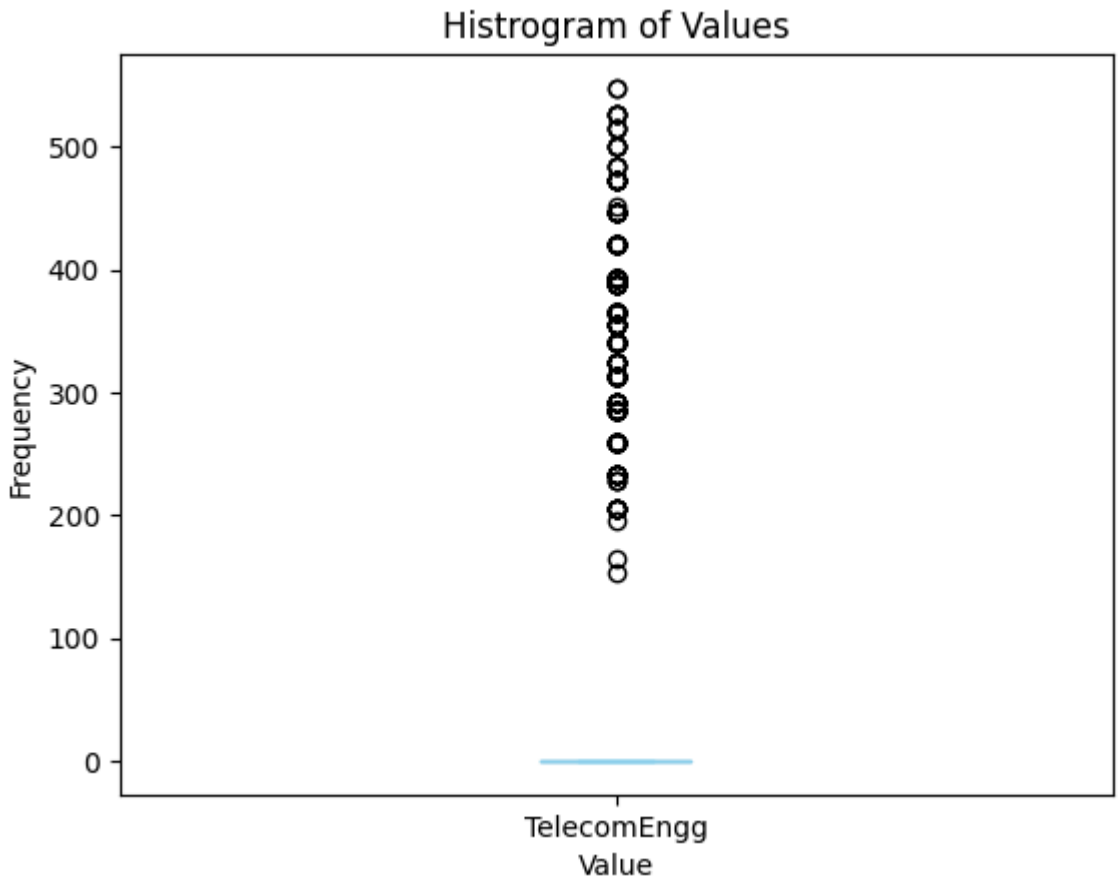
```
********** agreeableness **********
min      -5.781600
max       1.904800
mean      0.146496
median    0.212400
std       0.941782
skew     -1.204915
kurt      3.391242
Name: agreeableness, dtype: float64

********** extraversion **********
min      -4.600900
max       2.535400
mean      0.002763
median    0.091400
std       0.951471
skew     -0.523267
kurt      0.643969
Name: extraversion, dtype: float64

********** nueroticism **********
min      -2.643000
max       3.352500
mean     -0.169033
median   -0.234400
std       1.007580
skew      0.165710
kurt     -0.191539
Name: nueroticism, dtype: float64

********** openess_to_experience **********
min      -7.375700
max       1.822400
mean     -0.138110
median   -0.094300
std       1.008075
skew     -1.506962
kurt      5.788327
Name: openess_to_experience, dtype: float64
```

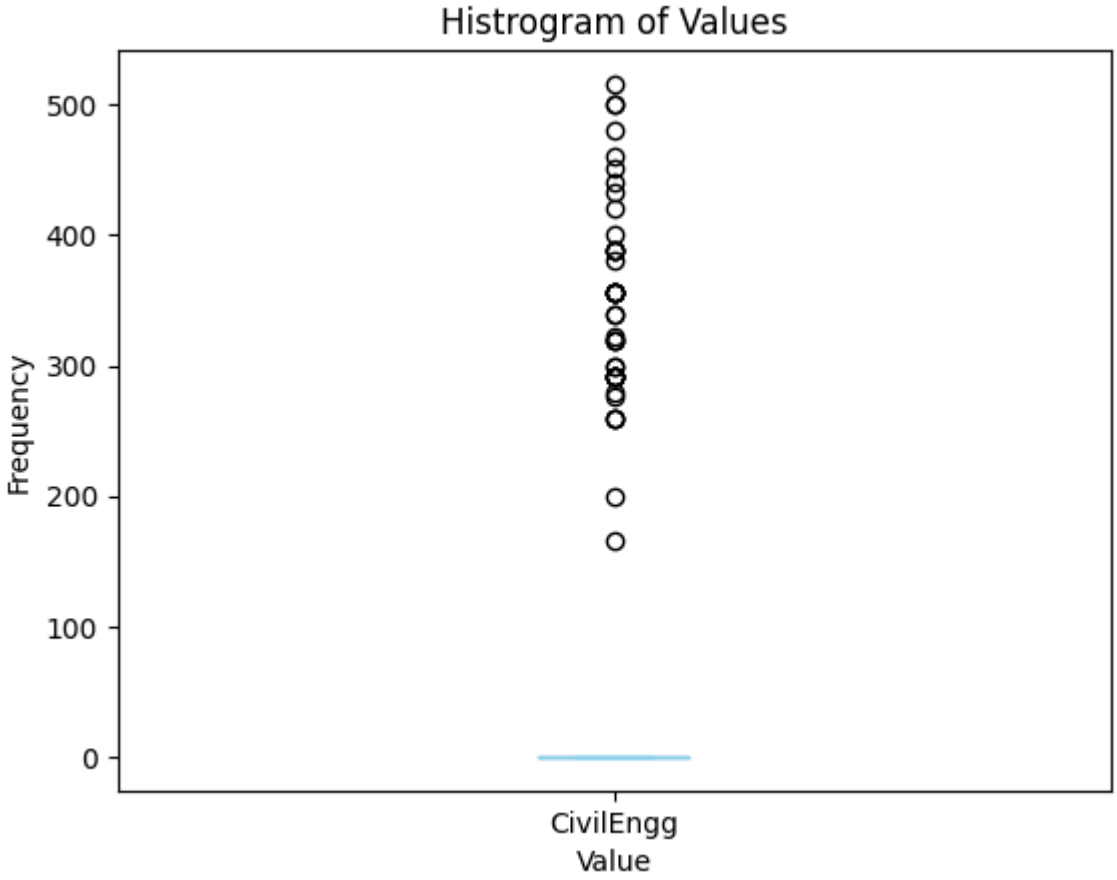
In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```

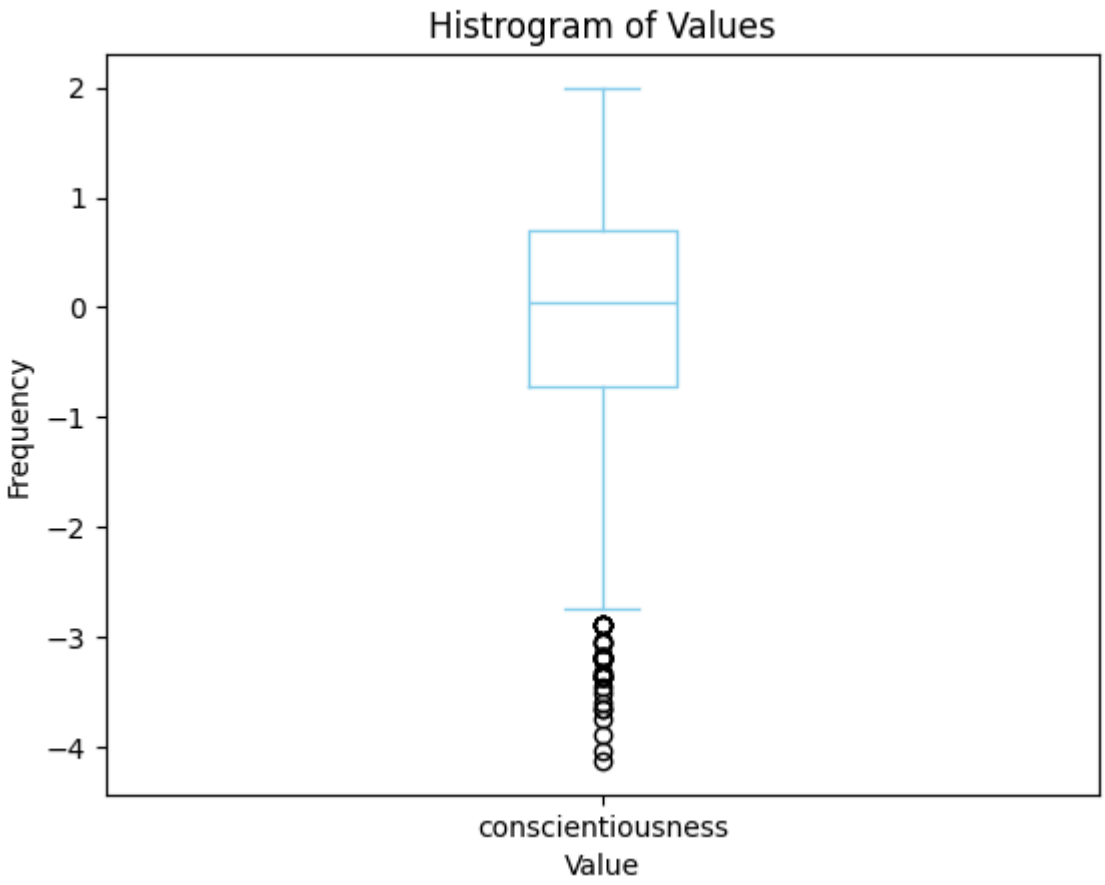

Histrogram of Values
Frequency
700
600
500
400
300
200
100
0
ComputerScience
Value

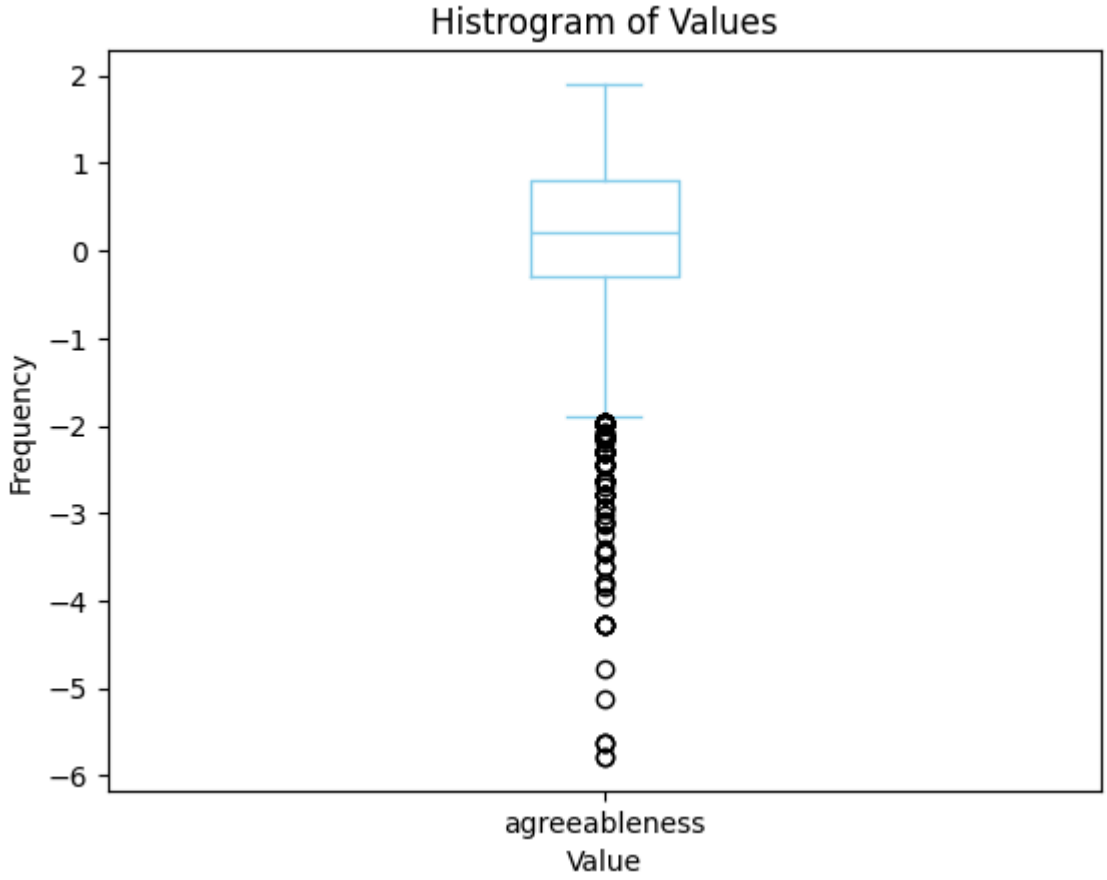

Histrogram of Values
Frequency
600
500
400
300
200
100
0
MechanicalEngg
Value

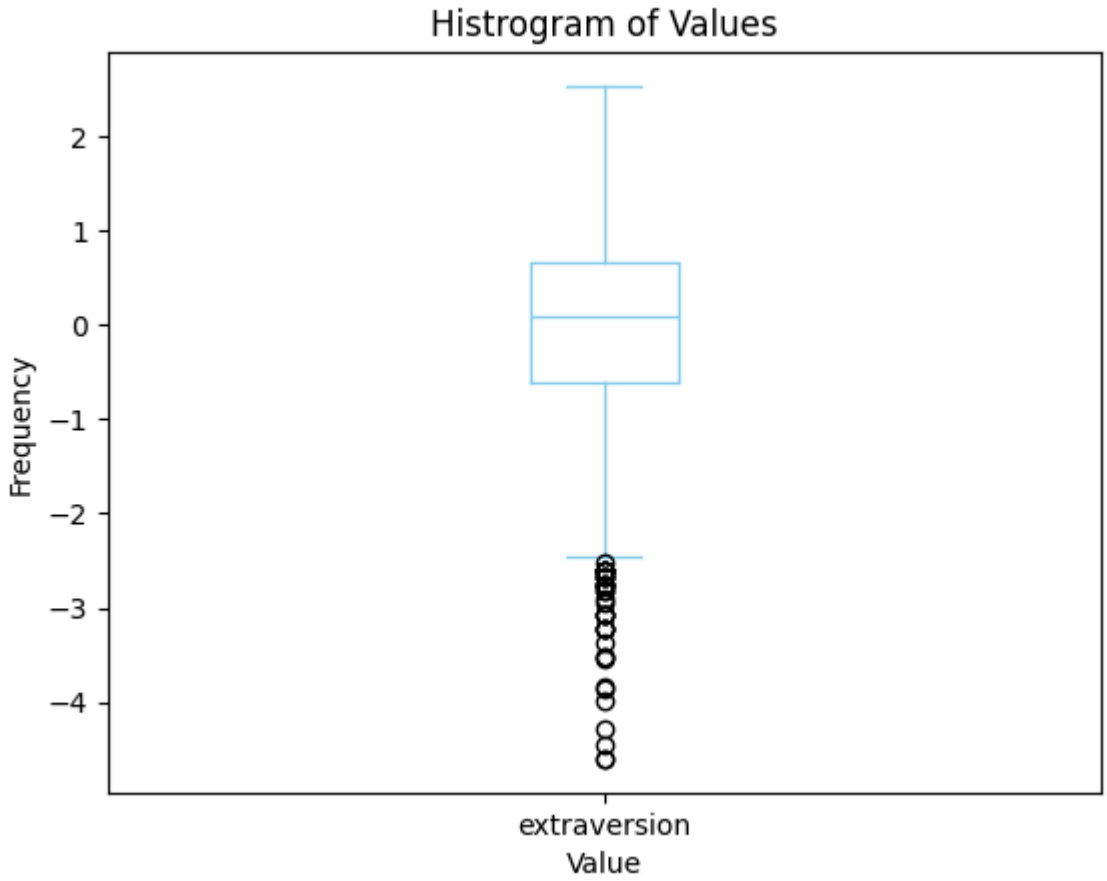

Histrogram of Values
700
600
500
400
300
200
100
0
Frequency
ElectricalEngg
Value

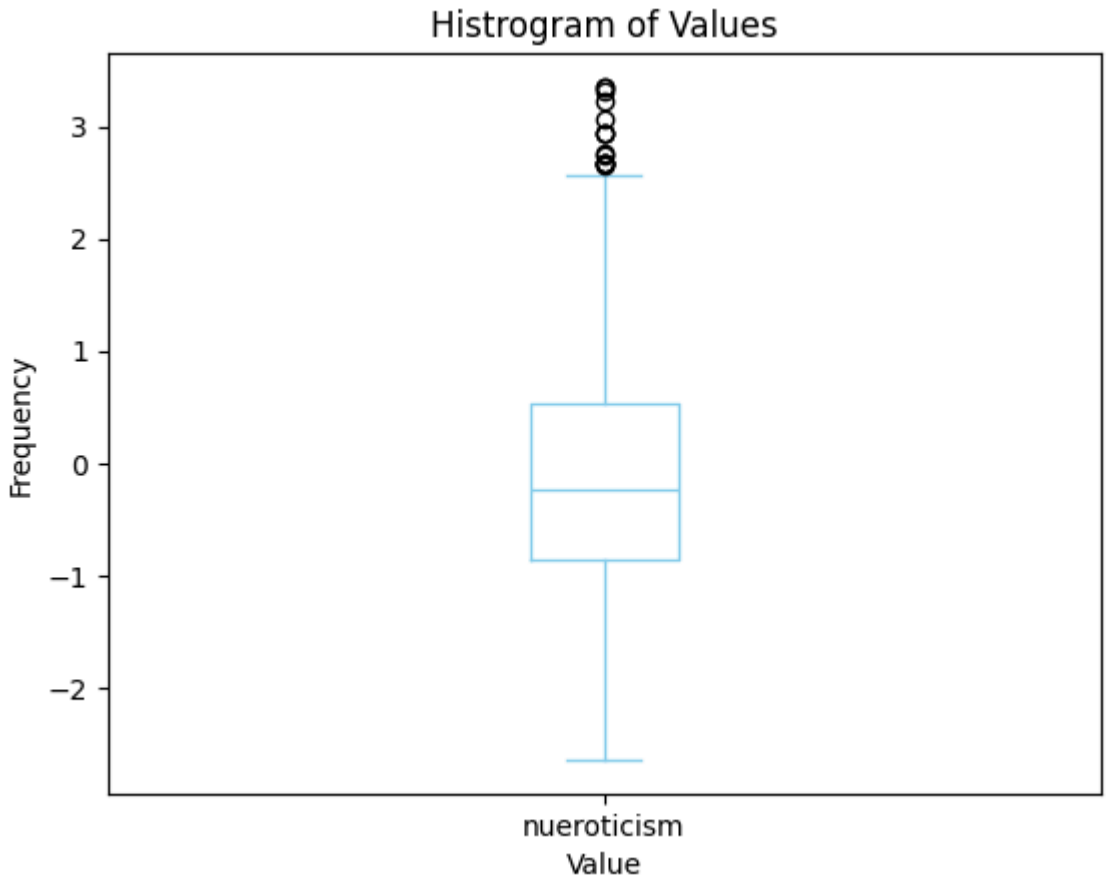

Histrogram of Values
500
400
300
200
100
0
Frequency
TelecomEngg
Value

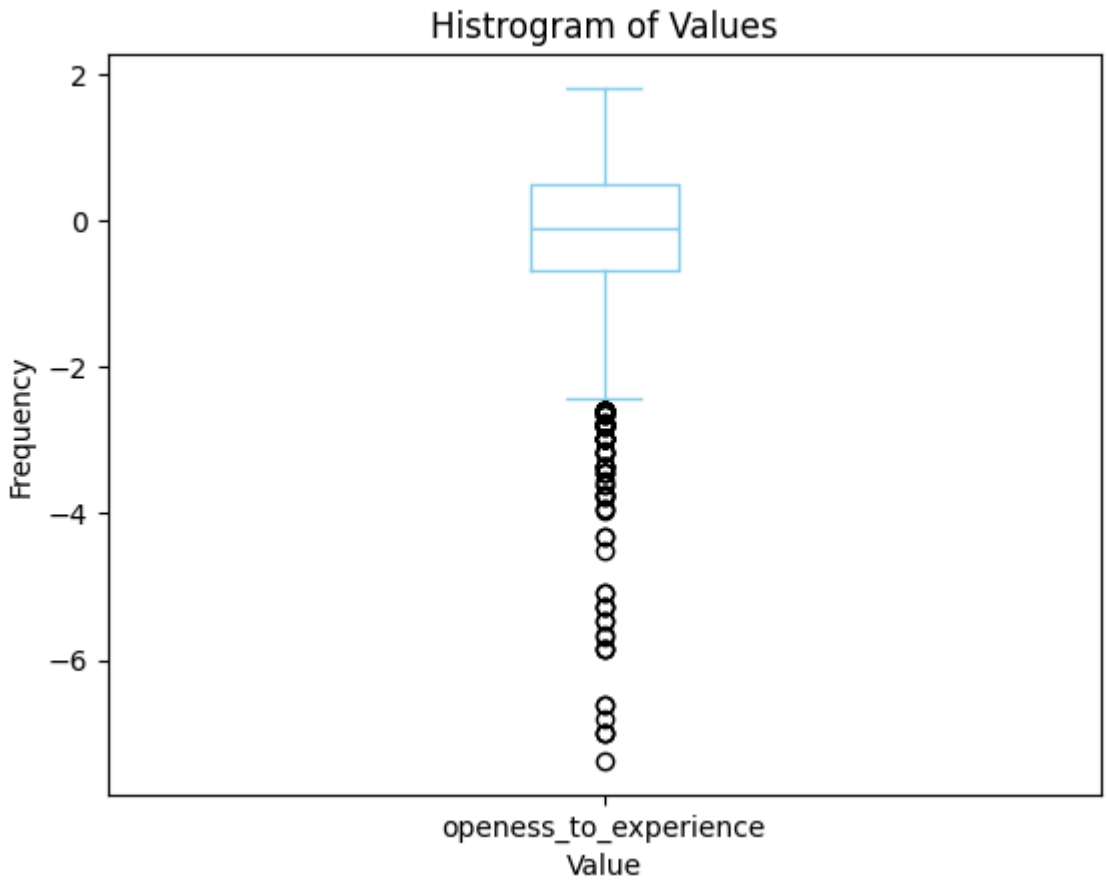

Histrogram of Values
Frequency
500
400
300
200
100
0
CivilEngg
Value


Histrogram of Values
Frequency
2
1
0
−1
−2
−3
−4
conscientiousness
Value


Histrogram of Values
Frequency
2
1
0
−1
−2
−3
−4
−5
−6
agreeableness
Value


Histrogram of Values
Frequency
2
1
0
−1
−2
−3
−4
extraversion
Value





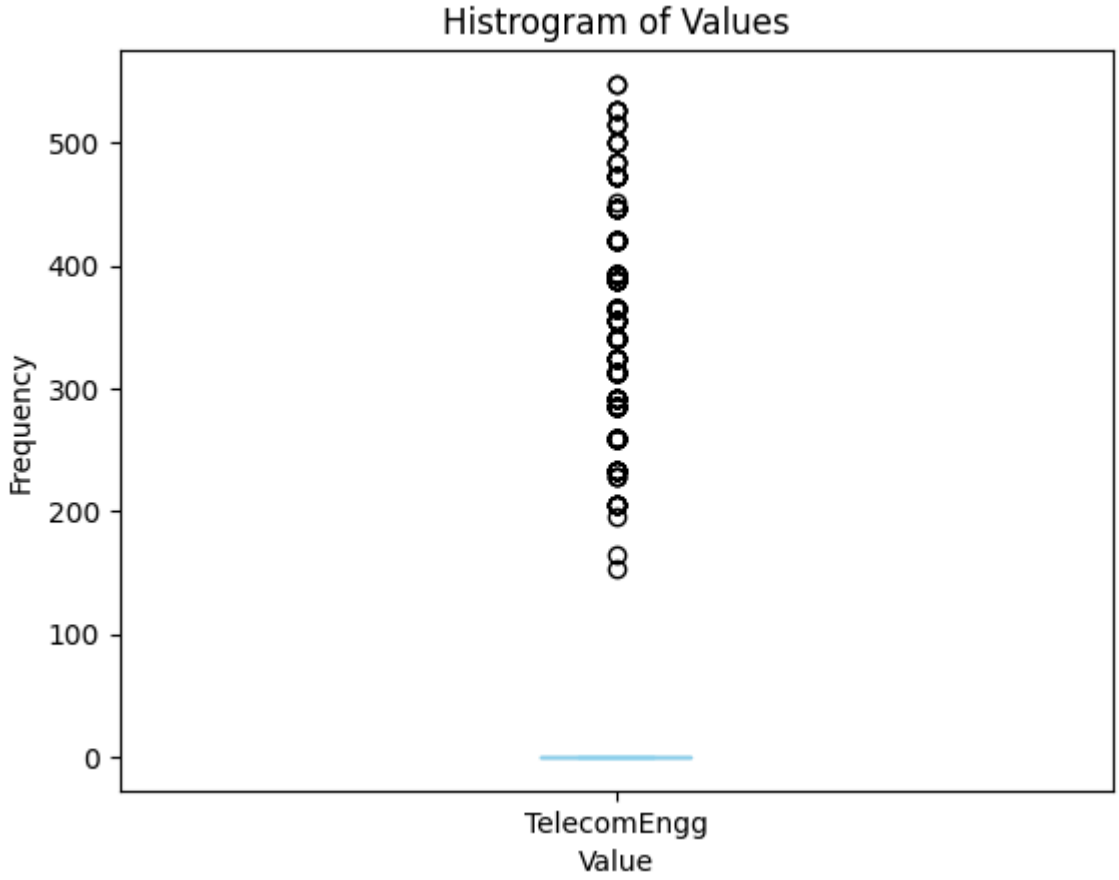
In [ ]:
```
num=['TelecomEngg','CivilEngg','conscientiousness']
```

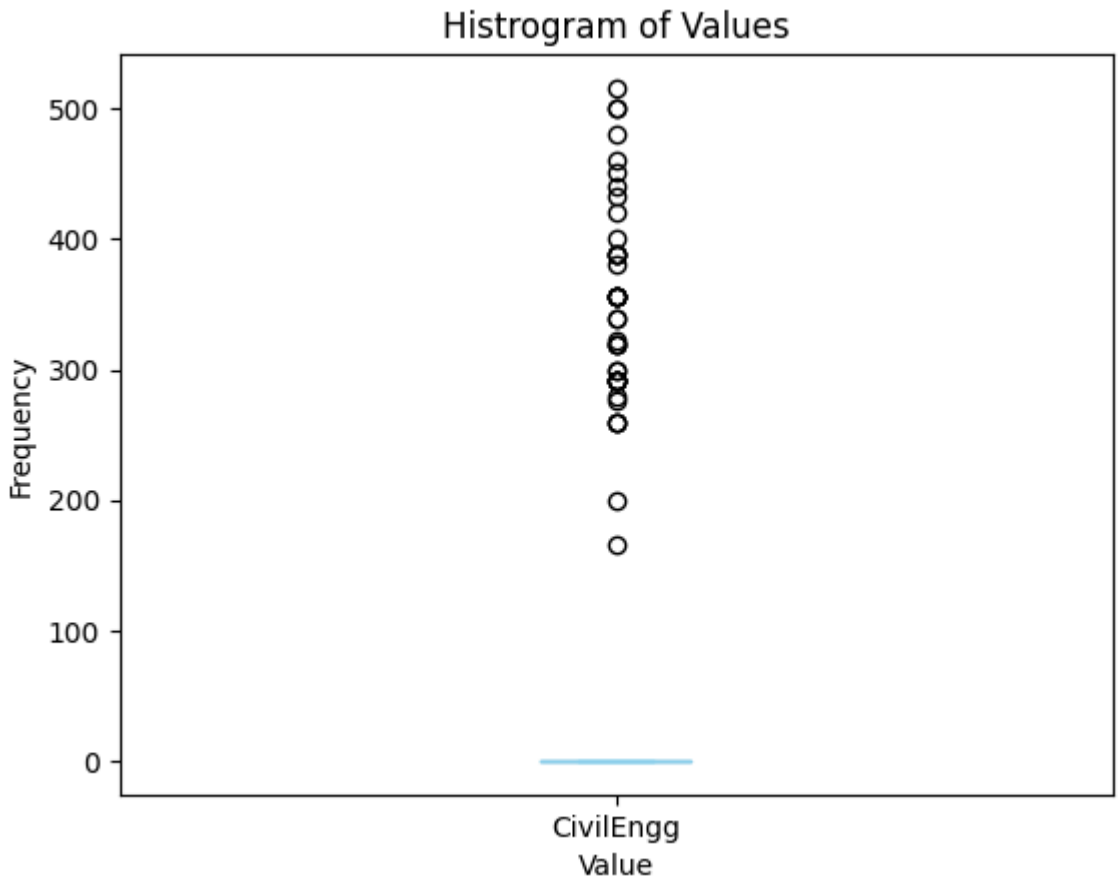
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
```

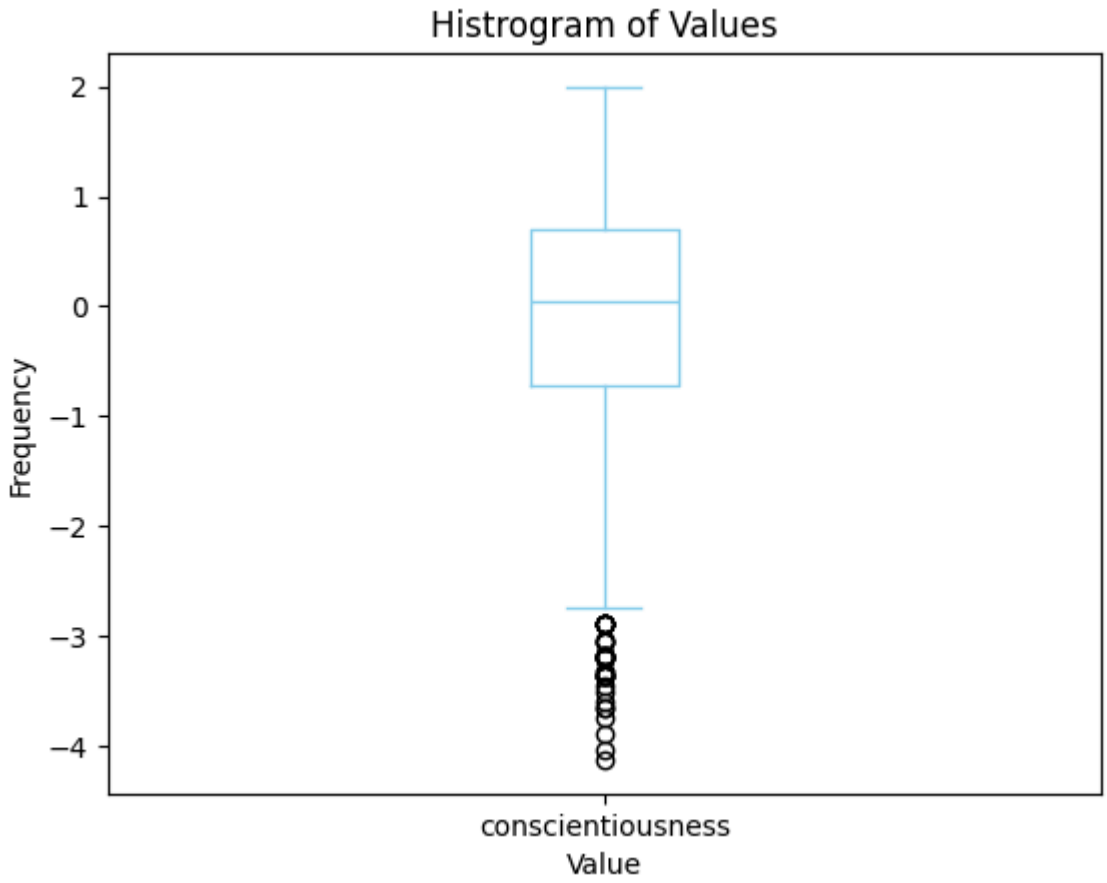
```
agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** TelecomEngg **********
min         0.000000
max       548.000000
mean       32.757629
median      0.000000
std       104.568796
skew        3.042584
kurt        7.821100
Name: TelecomEngg, dtype: float64

********** CivilEngg **********
min         0.000000
max       516.000000
mean        3.673337
median      0.000000
std        36.559052
skew       10.319461
kurt      109.142713
Name: CivilEngg, dtype: float64

********** conscientiousness **********
min      -4.126700
max       1.995300
mean     -0.037831
median    0.046400
std       1.028666
skew     -0.527003
kurt      0.122596
Name: conscientiousness, dtype: float64
```

In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```


Histrogram of Values
Frequency
500
400
300
200
100
0
TelecomEngg
Value


Histrogram of Values
Frequency
500
400
300
200
100
0
CivilEngg
Value

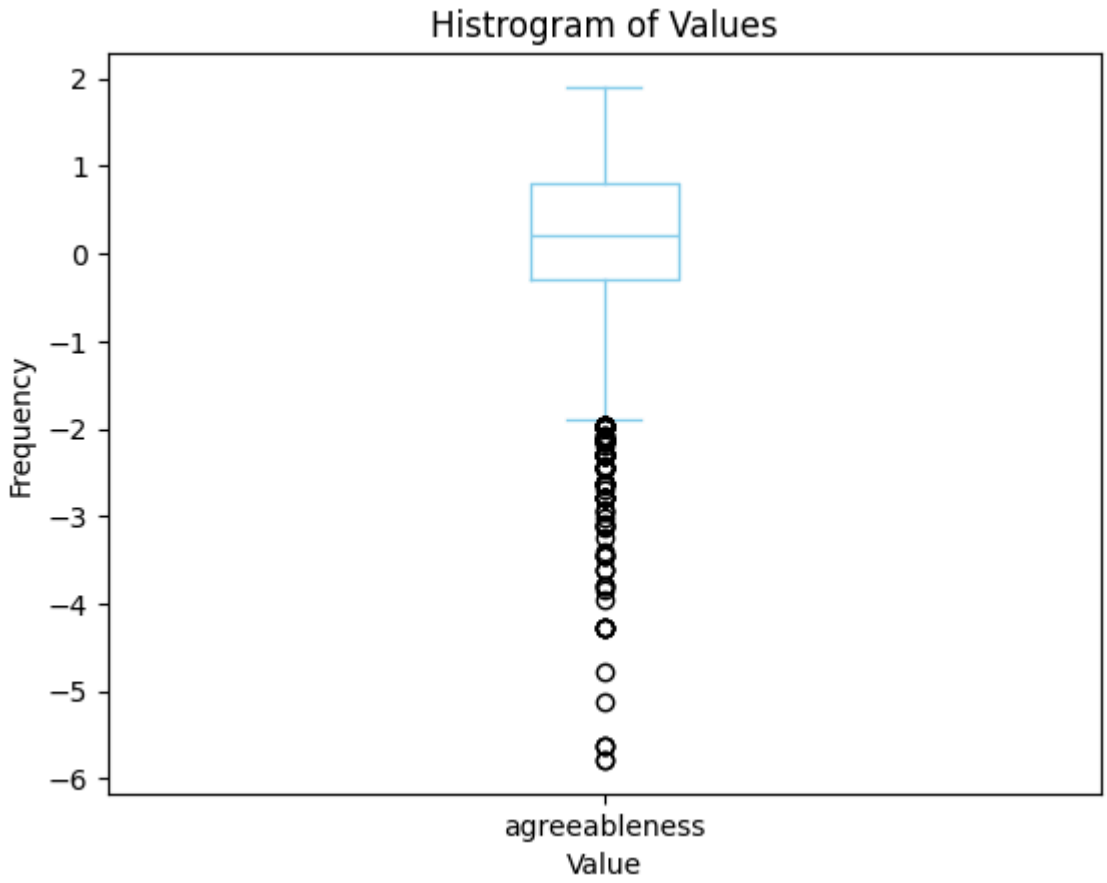


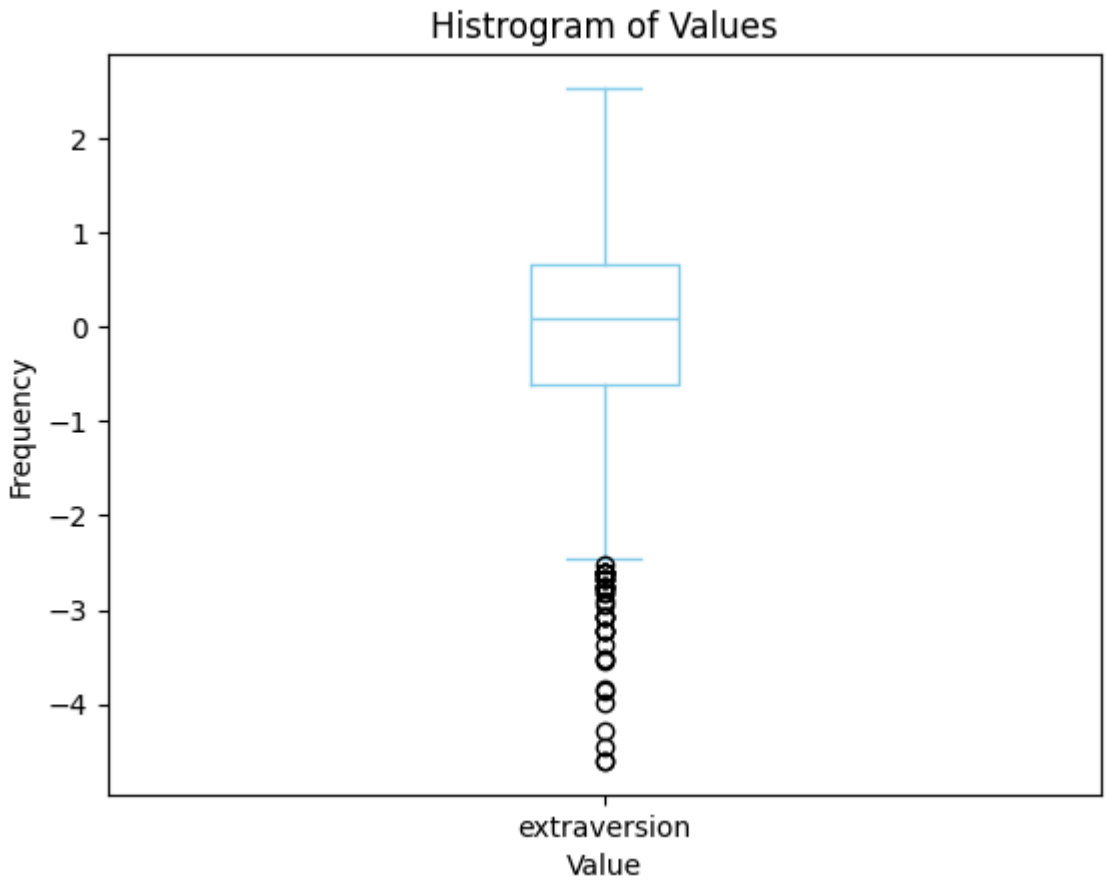
In [ ]:
```
num=['agreeableness','extraversion','nueroticism']
```

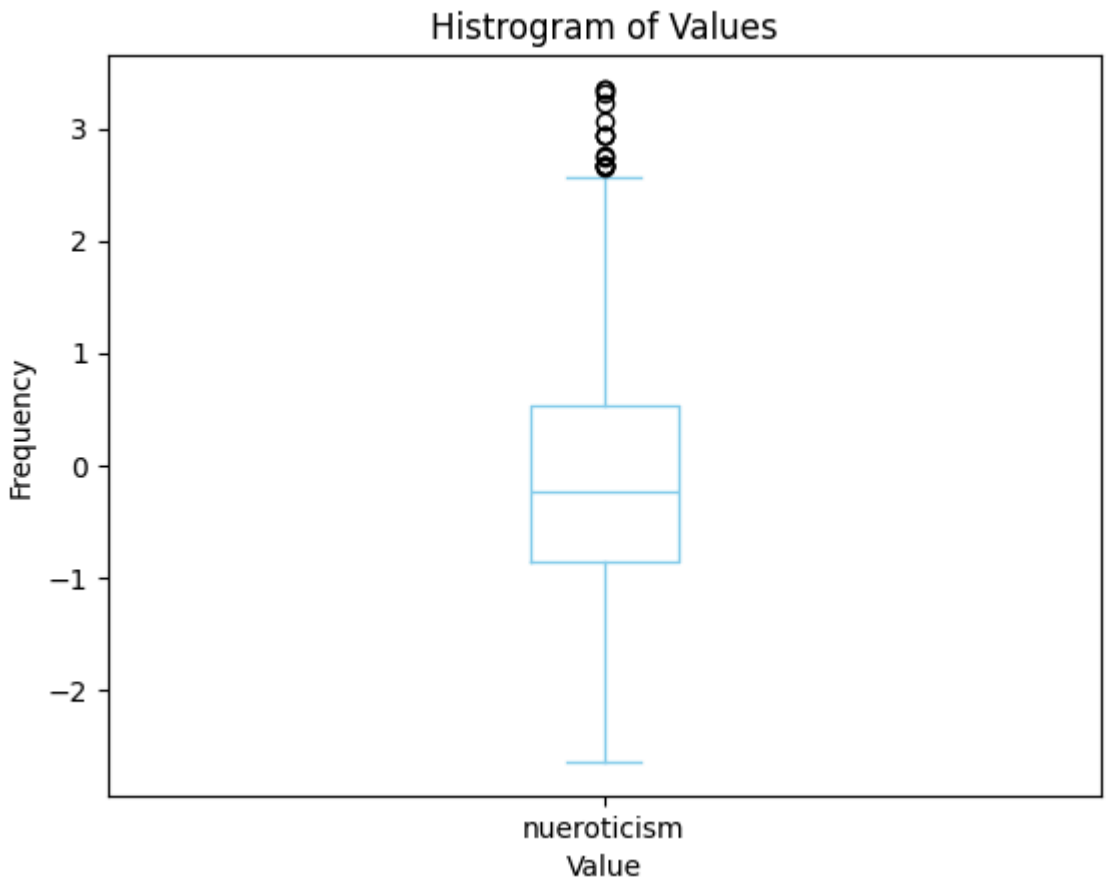
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** agreeableness **********
min      -5.781600
max       1.904800
mean      0.146496
median    0.212400
std       0.941782
skew     -1.204915
kurt      3.391242
Name: agreeableness, dtype: float64

********** extraversion **********
min      -4.600900
max       2.535400
mean      0.002763
median    0.091400
std       0.951471
skew     -0.523267
kurt      0.643969
Name: extraversion, dtype: float64

********** nueroticism **********
min      -2.643000
max       3.352500
mean     -0.169033
median   -0.234400
std       1.007580
skew      0.165710
kurt     -0.191539
Name: nueroticism, dtype: float64
```

In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```


Histrogram of Values
Frequency
2
1
0
−1
−2
−3
−4
−5
−6
agreeableness
Value


Histrogram of Values
Frequency
2
1
0
−1
−2
−3
−4
extraversion
Value

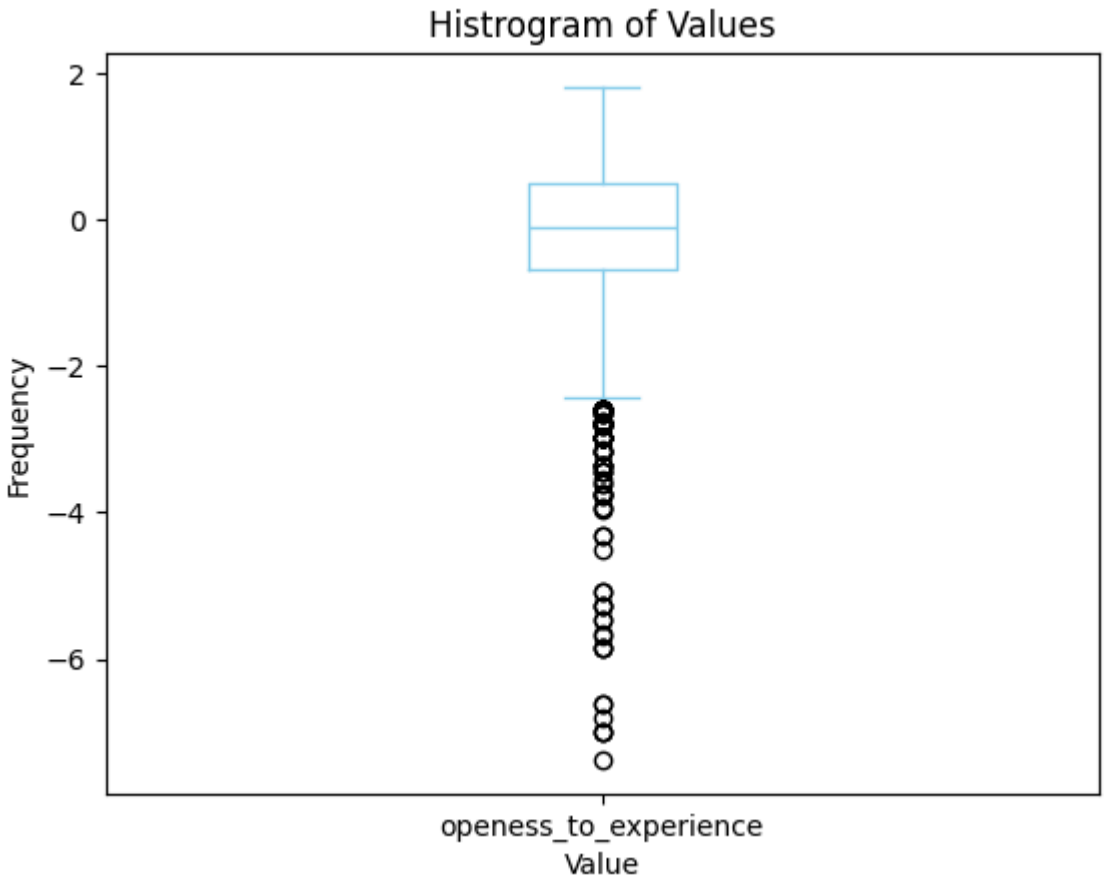


In [ ]:
```
num=['openess_to_experience']
```

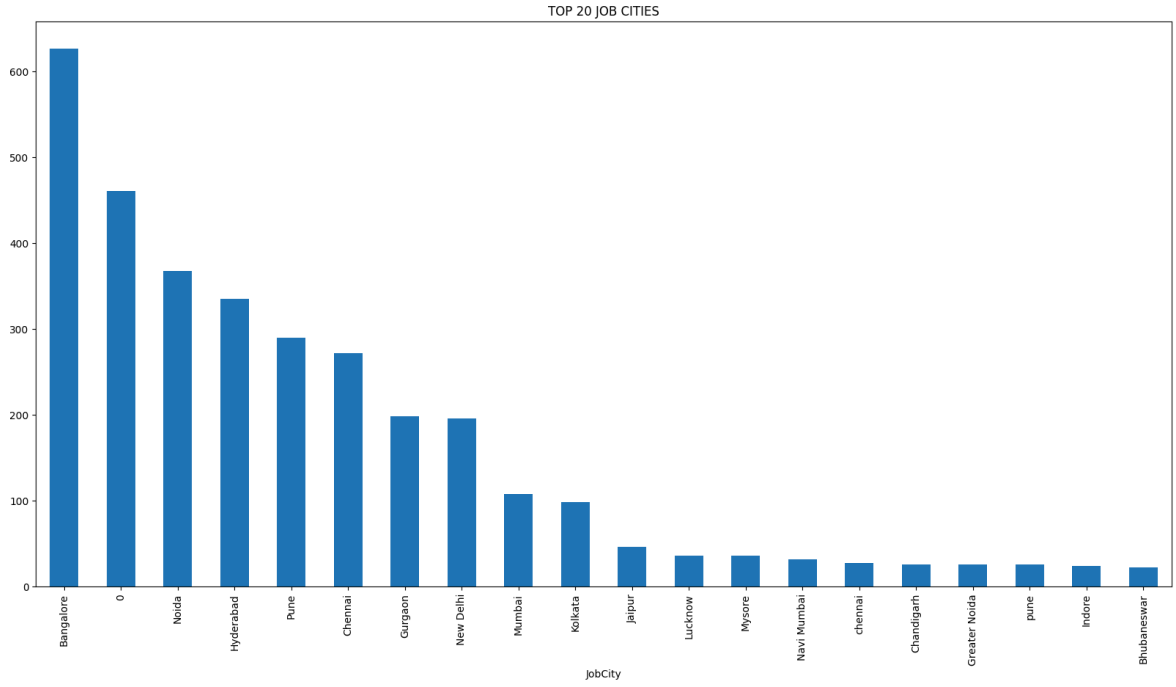
In [ ]:
```
for col_name in num:
    print('*'*10,col_name,'*'*10)
    print(data[col_name].
 agg(['min','max','mean','median','std','skew','kurt']))
    print()
```

```
********** openess_to_experience **********
min      -7.375700
max       1.822400
mean     -0.138110
median   -0.094300
std       1.008075
skew     -1.506962
kurt      5.788327
Name: openess_to_experience, dtype: float64
```

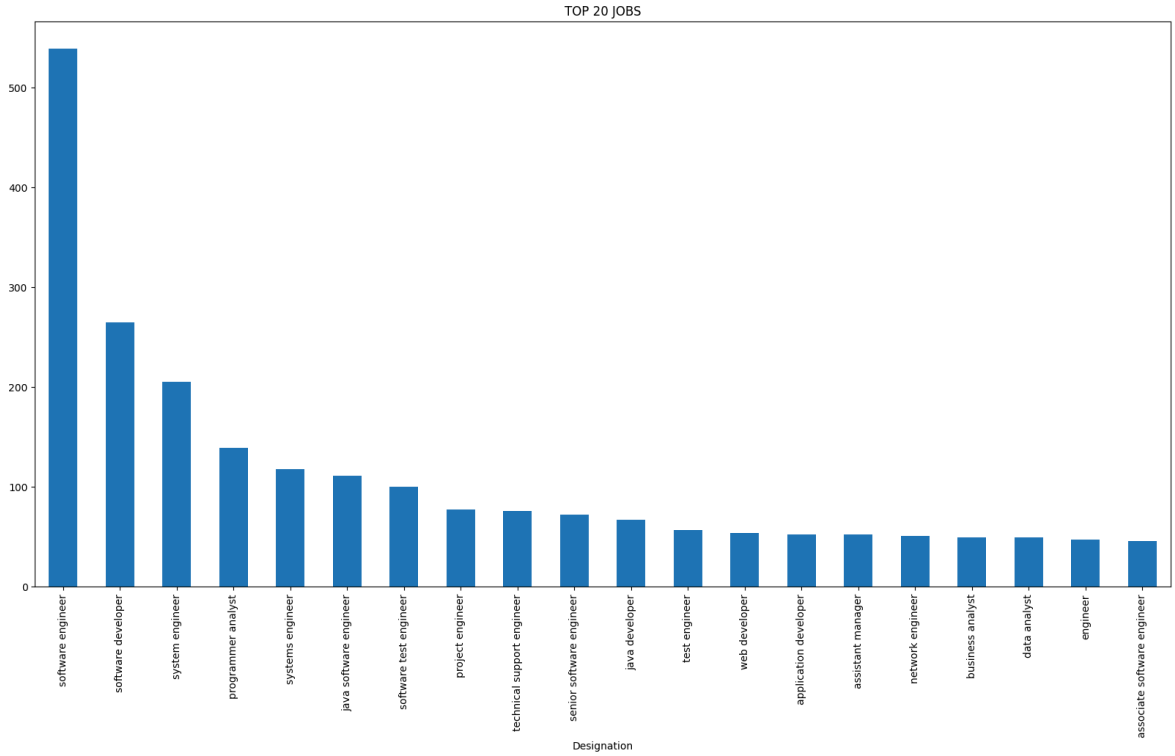
In [ ]:
```
for col_name in num:
    data[col_name].plot(kind='box',color='skyblue')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.title('Histrogram of Values')
    plt.show()
```



In [ ]:
```
plt.figure(figsize=(20,10))
city=data['JobCity'].value_counts()[:20].plot.bar()
plt.title('TOP 20 JOB CITIES')
plt.show()
```

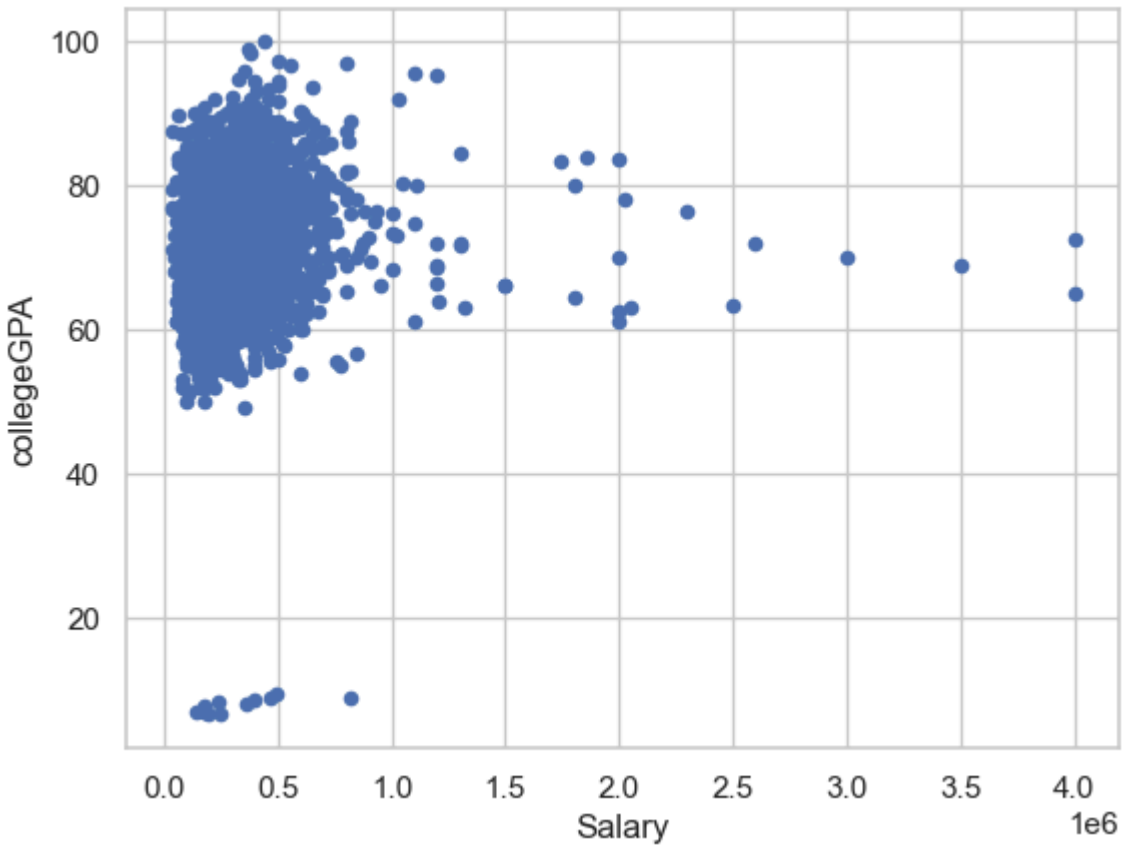


In [ ]:
```
plt.figure(figsize=(20,10))
city=data['Designation'].value_counts()[:20].plot.bar()
plt.title('TOP 20 JOBS')
plt.show()
```



In [ ]:
```
data.plot.scatter('Salary','collegeGPA')
```

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



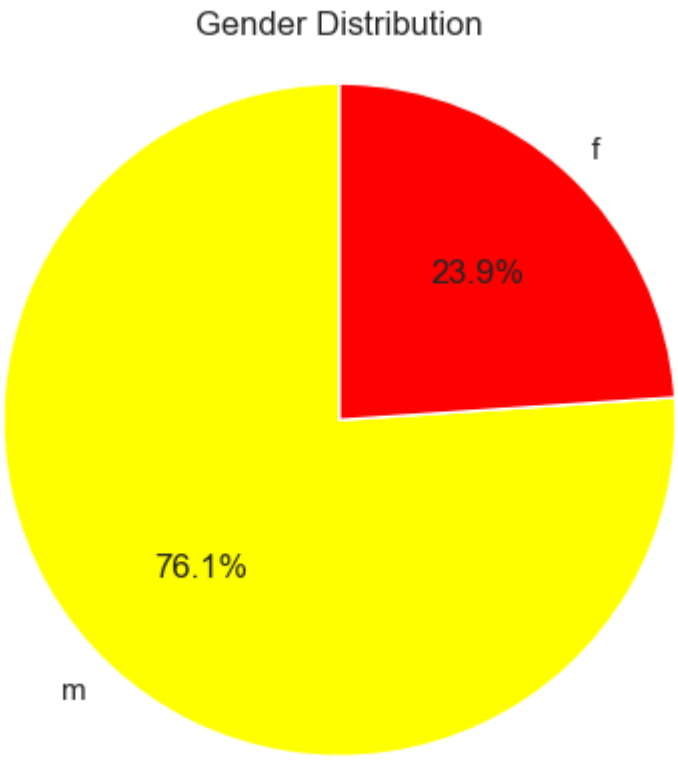
In [ ]:
```
data
```

Out[ ]:

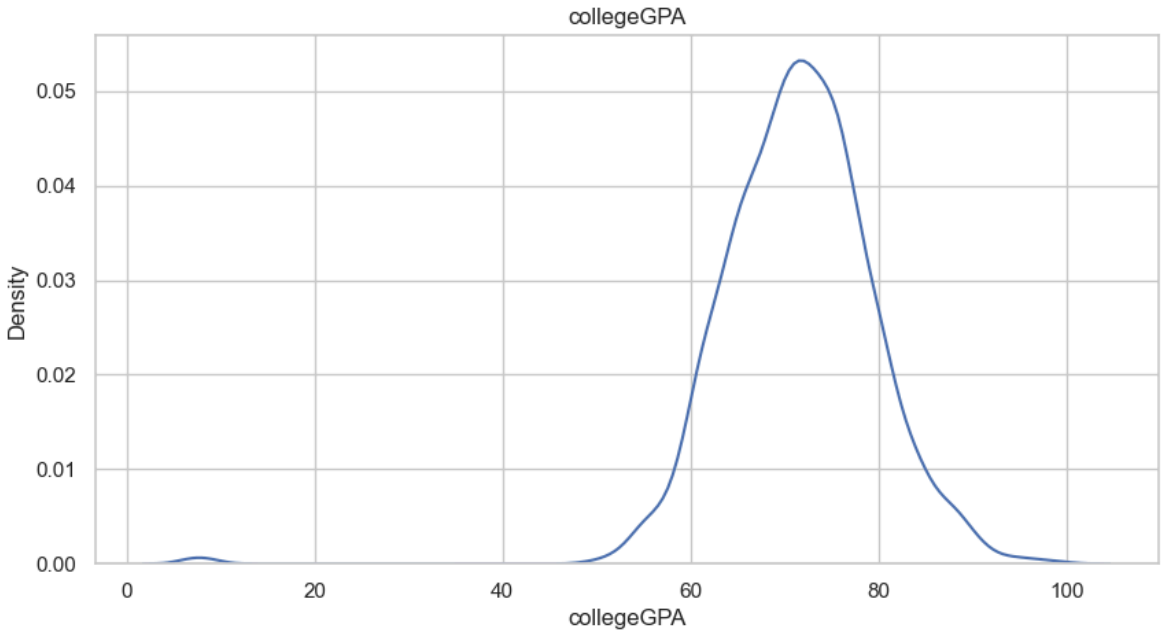
	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gen
0	train	203097	420000	2012-06-01	present	senior quality engineer	Bangalore	
1	train	579905	500000	2013-09-01	present	assistant manager	Indore	
2	train	810601	325000	2014-06-01	present	systems engineer	Chennai	
3	train	267447	1100000	2011-07-01	present	senior software engineer	Gurgaon	
4	train	343523	200000	2014-03-01	2015-03-01 00:00:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000	2011-10-01	2012-10-01 00:00:00	software engineer	New Delhi	
3994	train	752781	100000	2013-07-01	2013-07-01 00:00:00	technical writer	Hyderabad	
3995	train	355888	320000	2013-07-01	present	associate software engineer	Bangalore	
3996	train	947111	200000	2014-07-01	2015-01-01 00:00:00	software developer	Asifabadbanglore	
3997	train	324966	400000	2013-02-01	present	senior systems engineer	Chennai	

3998 rows × 39 columns

In [ ]:
```
gender_Counts=data['Gender'].value_counts()

colors=['yellow','red']

plt.pie(gender_Counts, labels=gender_Counts.index,autopct='%1.1f%%',startangle=9
plt.title('Gender Distribution')
plt.axis('equal')
plt.show()
```



In [ ]:
```
plt.figure(figsize=(10,5))
sns.kdeplot(data['collegeGPA'])
plt.title('collegeGPA')
plt.show()
```



In [ ]:
```
data
```

Out[ ]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gen
0	train	203097	420000	2012-06-01	present	senior quality engineer	Bangalore	
1	train	579905	500000	2013-09-01	present	assistant manager	Indore	
2	train	810601	325000	2014-06-01	present	systems engineer	Chennai	
3	train	267447	1100000	2011-07-01	present	senior software engineer	Gurgaon	
4	train	343523	200000	2014-03-01	2015-03-01 00:00:00	get	Manesar	
...	...	...	...	...	...	...	...	
3993	train	47916	280000	2011-10-01	2012-10-01 00:00:00	software engineer	New Delhi	
3994	train	752781	100000	2013-07-01	2013-07-01 00:00:00	technical writer	Hyderabad	
3995	train	355888	320000	2013-07-01	present	associate software engineer	Bangalore	
3996	train	947111	200000	2014-07-01	2015-01-01 00:00:00	software developer	Asifabadbanglore	
3997	train	324966	400000	2013-02-01	present	senior systems engineer	Chennai	

3998 rows × 39 columns

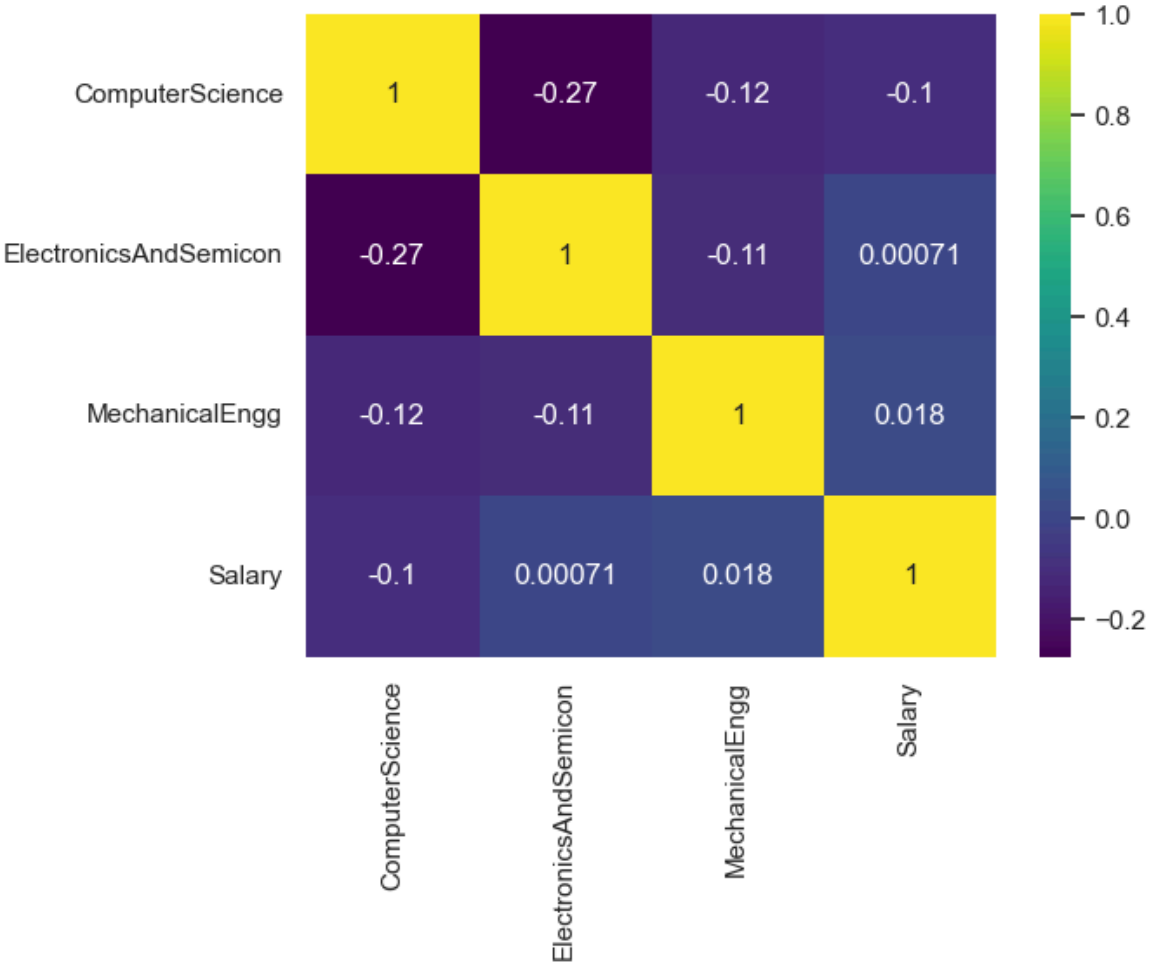
In [ ]:
```
data[['ComputerScience','ElectronicsAndSemicon','MechanicalEngg','Salary']].corr
```

Out[ ]:

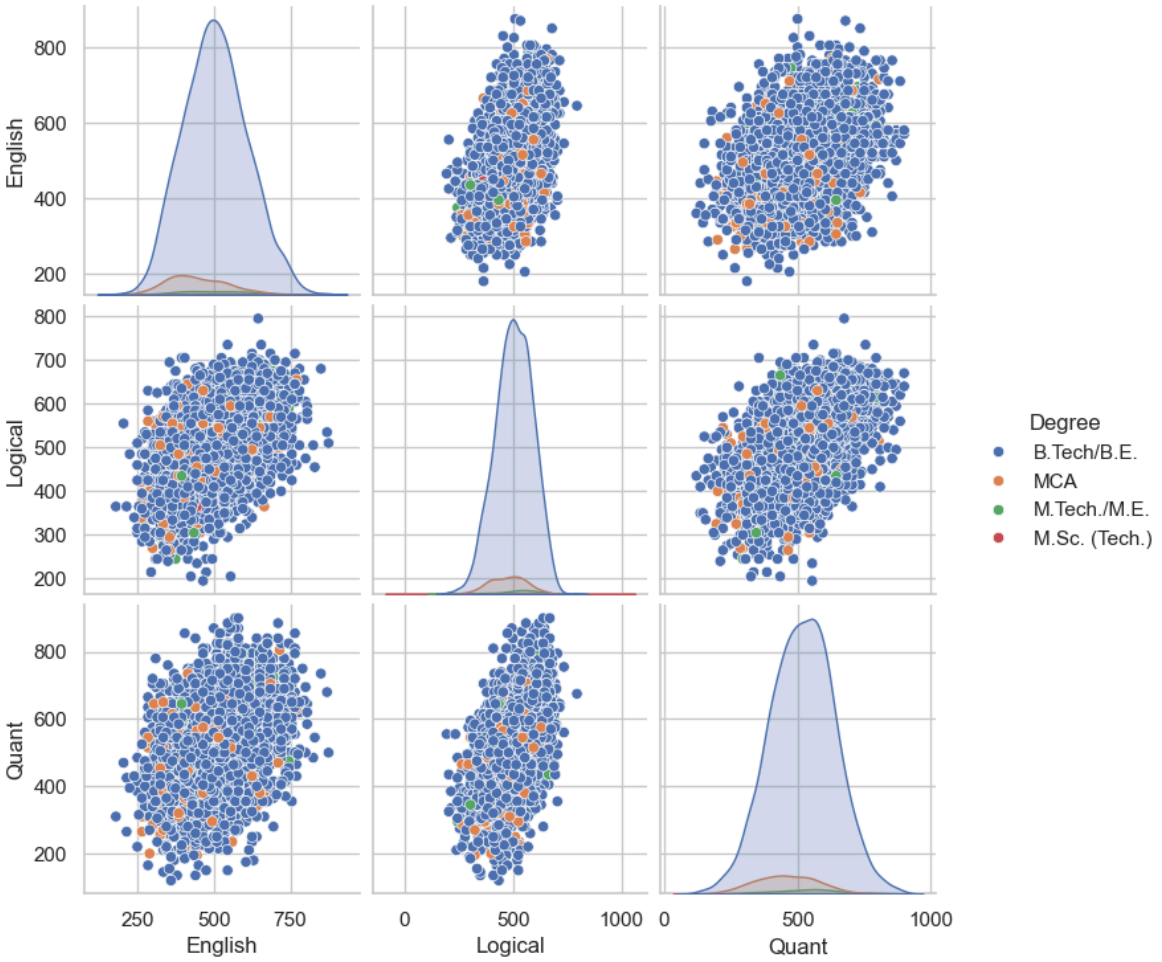
	ComputerScience	ElectronicsAndSemicon	MechanicalEngg	S
ComputerScience	1.000000	-0.273619	-0.124326	-0.10
ElectronicsAndSemicon	-0.273619	1.000000	-0.109414	0.00
MechanicalEngg	-0.124326	-0.109414	1.000000	0.01
Salary	-0.100674	0.000708	0.018493	1.00

In [ ]:
```
sns.heatmap(data[['ComputerScience','ElectronicsAndSemicon','MechanicalEngg','Sa
```

Out[ ]: <Axes: >



In [ ]:
```
sns.pairplot(data,x_vars=['English','Logical','Quant'],y_vars=['English','Logica
plt.show()
```


English
Logical
Quant
Degree
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)