amcat-data-analysis-in1242074

October 5, 2024

[1]:
```
# Import numpy,pandas and sns libraries
import numpy as np
import pandas as pd
import seaborn as sns
```

[2]:
```
# Read excel into the Dataframe using pandas
df_data=pd.read_excel('C:/Users/madhu/Desktop/kanav bansal/data.xlsx')
```

[3]:
```
# Display first 5 rows using head
df_data.head()
```

[3]:

	Unnamed: 0	ID	Salary	DOJ	DOL	\
0	train	203097	420000	2012-06-01	present	
1	train	579905	500000	2013-09-01	present	
2	train	810601	325000	2014-06-01	present	
3	train	267447	1100000	2011-07-01	present	
4	train	343523	200000	2014-03-01	2015-03-01 00:00:00	

	Designation	JobCity	Gender	DOB	10percentage	…	\
0	senior quality engineer	Bangalore	f	1990-02-19	84.3	…	
1	assistant manager	Indore	m	1989-10-04	85.4	…	
2	systems engineer	Chennai	f	1992-08-03	85.0	…	
3	senior software engineer	Gurgaon	m	1989-12-05	85.6	…	
4	get	Manesar	m	1991-02-27	78.0	…	

	ComputerScience	MechanicalEngg	ElectricalEngg	TelecomEngg	CivilEngg	\
0	-1	-1	-1	-1	-1	
1	-1	-1	-1	-1	-1	
2	-1	-1	-1	-1	-1	
3	-1	-1	-1	-1	-1	
4	-1	-1	-1	-1	-1	

	conscientiousness	agreeableness	extraversion	nueroticism	\
0	0.9737	0.8128	0.5269	1.35490	
1	-0.7335	0.3789	1.2396	-0.10760	
2	0.2718	1.7109	0.1637	-0.86820	
3	0.0464	0.3448	-0.3440	-0.40780	

```
4            -0.8810       -0.2793       -1.0697       0.09163

   openess_to_experience
0                -0.4455
1                 0.8637
2                 0.6721
3                -0.9194
4                -0.1295

[5 rows x 39 columns]
```

[4]:
```
# find the size of the dataframe using size variable
df_data.size
```

[4]:
```
155922
```

[5]:
```
# Use describe function on the Dataframe displays the Statistical parameters␣
  ↪like mean,min etc
df_data.describe()
```

[5]:
```
                 ID        Salary                            DOJ  \
count  3.998000e+03  3.998000e+03                           3998
mean   6.637945e+05  3.076998e+05  2013-07-02 11:04:10.325162496
min    1.124400e+04  3.500000e+04            1991-06-01 00:00:00
25%    3.342842e+05  1.800000e+05            2012-10-01 00:00:00
50%    6.396000e+05  3.000000e+05            2013-11-01 00:00:00
75%    9.904800e+05  3.700000e+05            2014-07-01 00:00:00
max    1.298275e+06  4.000000e+06            2015-12-01 00:00:00
std    3.632182e+05  2.127375e+05                            NaN

                                DOB  10percentage  12graduation  \
count                          3998   3998.000000   3998.000000
mean   1990-12-06 06:01:15.637819008     77.925443   2008.087544
min              1977-10-30 00:00:00     43.000000   1995.000000
25%              1989-11-16 06:00:00     71.680000   2007.000000
50%              1991-03-07 12:00:00     79.150000   2008.000000
75%              1992-03-13 18:00:00     85.670000   2009.000000
max              1997-05-27 00:00:00     97.760000   2013.000000
std                              NaN      9.850162      1.653599

       12percentage     CollegeID  CollegeTier   collegeGPA  …  \
count   3998.000000   3998.000000  3998.000000  3998.000000  …
mean      74.466366   5156.851426     1.925713    71.486171  …
min       40.000000      2.000000     1.000000     6.450000  …
25%       66.000000    494.000000     2.000000    66.407500  …
50%       74.400000   3879.000000     2.000000    71.720000  …
75%       82.600000   8818.000000     2.000000    76.327500  …
```

```
max       98.700000  18409.000000     2.000000    99.930000  …
std       10.999933   4802.261482     0.262270     8.167338  …

       ComputerScience  MechanicalEngg  ElectricalEngg  TelecomEngg  \
count      3998.000000     3998.000000     3998.000000  3998.000000
mean         90.742371       22.974737       16.478739    31.851176
min          -1.000000       -1.000000       -1.000000    -1.000000
25%          -1.000000       -1.000000       -1.000000    -1.000000
50%          -1.000000       -1.000000       -1.000000    -1.000000
75%          -1.000000       -1.000000       -1.000000    -1.000000
max         715.000000      623.000000      676.000000   548.000000
std         175.273083       98.123311       87.585634   104.852845

         CivilEngg  conscientiousness  agreeableness  extraversion  \
count  3998.000000        3998.000000    3998.000000   3998.000000
mean      2.683842          -0.037831       0.146496      0.002763
min      -1.000000          -4.126700      -5.781600     -4.600900
25%      -1.000000          -0.713525      -0.287100     -0.604800
50%      -1.000000           0.046400       0.212400      0.091400
75%      -1.000000           0.702700       0.812800      0.672000
max     516.000000           1.995300       1.904800      2.535400
std      36.658505           1.028666       0.941782      0.951471

       nueroticism  openess_to_experience
count  3998.000000            3998.000000
mean     -0.169033              -0.138110
min      -2.643000              -7.375700
25%      -0.868200              -0.669200
50%      -0.234400              -0.094300
75%       0.526200               0.502400
max       3.352500               1.822400
std       1.007580               1.008075

[8 rows x 29 columns]
```

[6]:
```
# info() method lists the columns in the Dataframe and its type object,int,float
df_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
```

[7]:
```
# count method displays the row count
df_data['Unnamed: 0'].count()
```

[7]: 3998

[8]:
```
# nunique method displays the number of unique values in a dataframe column␣
 ↪unnamed:  0
df_data['Unnamed: 0'].nunique()
```

[8]:
```
1
```

[9]:
```
# displays the unique string in the column unnamed:  0
df_data['Unnamed: 0'].unique()
```

[9]:
```
array(['train'], dtype=object)
```

[10]:
```
# displays the value counts in the dataframe of column Unnamed:  0
df_data['Unnamed: 0'].value_counts()
```

[10]:
```
Unnamed: 0
train    3998
Name: count, dtype: int64
```

[11]:
```
# count method displays the row count
df_data['DOL'].count()
```

[11]:
```
3998
```

[12]:
```
# nunique method displays the number of unique values in a dataframe column DOL
df_data['DOL'].nunique()
```

[12]:
```
67
```

[13]:
```
# displays the unique string in the column DOL
df_data['DOL'].unique()
```

[13]:
```
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)
```

[14]:
```
# displays the value counts in the dataframe of column DOL
df_data['DOL'].value_counts()
```

[14]:
```
DOL
present                 1875
2015-04-01 00:00:00      573
2015-03-01 00:00:00      124
2015-05-01 00:00:00      112
2015-01-01 00:00:00       99
                         …
2005-03-01 00:00:00        1
2015-10-01 00:00:00        1
2010-02-01 00:00:00        1
2011-02-01 00:00:00        1
2010-10-01 00:00:00        1
Name: count, Length: 67, dtype: int64
```

[15]:
```
# count method displays the row count
df_data['Designation'].count()
```

[15]:
```
3998
```

[16]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪Designation
df_data['Designation'].nunique()
```

[16]:
```
419
```

[17]:
```
# displays the unique string in the column Designation
df_data['Designation'].unique()
```

[17]:
```
array(['senior quality engineer', 'assistant manager', 'systems engineer',
       'senior software engineer', 'get', 'system engineer',
       'java software engineer', 'mechanical engineer',
       'electrical engineer', 'project engineer', 'senior php developer',
       'senior systems engineer', 'quality assurance engineer',
       'qa analyst', 'network engineer', 'product development engineer',
       'associate software developer', 'data entry operator',
       'software engineer', 'developer', 'electrical project engineer',
       'programmer analyst', 'systems analyst', 'ase',
       'telecommunication engineer', 'application developer',
       'ios developer', 'executive assistant', 'online marketing manager',
       'documentation specialist', 'associate software engineer',
       'management trainee', 'site manager', 'software developer',
       '.net developer', 'production engineer', 'jr. software engineer',
       'trainee software developer', 'ui developer',
       'assistant system engineer', 'android developer',
```

```
'customer service', 'test engineer', 'java developer', 'engineer',
'recruitment coordinator', 'technical support engineer',
'data analyst', 'assistant software engineer', 'faculty',
'entry level management trainee',
'customer service representative', 'software test engineer',
'firmware engineer', 'php developer', 'research associate',
'research analyst', 'quality engineer', 'programmer',
'technical support executive', 'business analyst', 'web developer',
'application engineer', 'project coordinator', 'engineer trainee',
'sap consultant', 'quality analyst', 'marketing coordinator',
'system administrator', 'senior engineer',
'business development managerde', 'network administrator',
'technical support specialist', 'business development executive',
'junior software engineer', 'asp.net developer',
'graduate engineer trainee', 'field engineer',
'assistant professor', 'trainee software engineer',
'senior software developer',
'quality assurance automation engineer', 'design engineer',
'telecom engineer', 'quality control engineer',
'hardware engineer', 'hr recruiter', 'sales associate',
'junior engineer', 'associate engineer', 'maintenance engineer',
'sales engineer', 'human resources associate',
'mobile application developer',
'electronic field service engineer', 'process associate',
'field service engineer', 'it support specialist',
'software development engineer', 'business process analyst',
'operation engineer', 'electrical designer', 'marketing assistant',
'sales executive', 'admin assistant', 'senior java developer',
'account executive', 'oracle dba', 'rf engineer',
'embedded software engineer', 'programmer analyst trainee',
'technical engineer', 'operations executive', 'trainee engineer',
'recruiter', 'lecturer', '.net web developer',
'marketing executive', 'operations assistant', 'associate manager',
'electrical design engineer', 'systems administrator',
'client services associate', 'it analyst', 'senior developer',
'cad designer', 'business technology analyst', 'asst. manager',
'service engineer', 'executive recruiter', 'planning engineer',
'associate technical operations', 'web designer',
'software architect', 'software quality assurance tester',
'seo trainee', 'process engineer',
'software quality assurance analyst', 'designer',
'business systems consultant', 'business development manager',
'junior research fellow', 'technical recruiter',
'operations analyst', 'quality assurance test engineer',
'linux systems administrator', 'software trainee',
'entry level sales and marketing', 'electrical field engineer',
'windows systems administrator', 'junior software developer',
```

```
'python developer', 'web application developer',
'assistant systems engineer', 'javascript developer',
'operation executive', 'performance engineer', 'technical writer',
'operations engineer and jetty handling', 'lead engineer',
'portfolio analyst', 'associate system engineer',
'mechanical design engineer', 'product engineer',
'network security engineer', 'operations manager',
'technical lead', 'operations', 'quality assurance tester',
'automation engineer', 'data scientist', 'quality associate',
'manual tester', 'sr. engineer', 'embedded engineer',
'service and sales engineer', 'telecom support engineer',
'engineer- customer support', 'cloud engineer', 'branch manager',
'business analyst consultant', 'technology lead',
'software trainee engineer', 'dcs engineer', 'junior manager',
'ux designer', 'clerical', 'hr generalist',
'database administrator', 'senior design engineer', 'seo',
'assistant engineer', 'marketing analyst', 'it executive',
'salesforce developer', 'software tester', 'sql dba',
'junior engineer product support', 'manager',
'senior business analyst', 'c# developer',
'implementation engineer', 'executive hr', 'executive engineer',
'sharepoint developer', 'system analyst',
'sales management trainee', 'senior project engineer',
'it recruiter', 'software engineer analyst',
'desktop support technician', 'continuous improvement engineer',
'process advisor', 'etl developer', 'sales and service engineer',
'project manager', 'training specialist', 'product manager',
'staffing recruiter', 'assistant programmer', 'quality controller',
'mis executive', 'game developer', 'digital marketing specialist',
'principal software engineer', 'software devloper',
'senior mechanical engineer', 'technical operations analyst',
'service coordinator', 'testing engineer', 'technical assistant',
'sap abap consultant', 'seo engineer', 'project assistant',
'talent acquisition specialist', 'sales account manager',
'software engineer trainee', 'customer service manager',
'help desk analyst', 'general manager', 'engineering manager',
'senior network engineer',
'field based employee relations manager', 'phone banking officer',
'support engineer', 'associate test engineer',
'technology analyst', 'network support engineer',
'it business analyst', 'junior system analyst',
'senior .net developer', 'secretary', 'research engineer',
'quality assurance auditor', 'process executive',
'lecturer & electrical maintenance', 'office coordinator',
'hr manager', 'html developer', 'sales support',
'front end web developer', 'administrative support',
'territory sales manager', 'project administrator',
```

```
'environmental engineer', 'web designer and seo',
'information security analyst',
'field business development associate', 'operational executive',
'administrative coordinator', 'senior risk consultant',
'desktop support engineer', 'cad drafter', 'noc engineer',
'industrial engineer', 'it engineer', 'human resources intern',
'senior quality assurance engineer', 'clerical assistant',
'software enginner', 'quality assurance',
'delivery software engineer', 'graphic designer',
'sales development manager', 'visiting faculty',
'business intelligence analyst', 'team lead',
'operational excellence manager', 'sales & service engineer',
'web intern', 'full stack developer', 'database developer',
'sr. database engineer', 'graduate apprentice trainee',
'software engineer associate', 'technical analyst',
'executive engg', 'it technician', 'business system analyst',
'process control engineer', 'technical consultant',
'business office manager', 'quality control inspector',
'product design engineer', 'manufacturing engineer',
'seo executive', 'sap analyst', 'software engineere',
'financial service consultant', 'co faculty', 'software analyst',
'desktop support analyst', 'graduate engineer',
'engineering technician', 'it assistant', 'marketing manager',
'human resource assistant', 'hr assistant', 'product developer',
'customer support engineer',
'quality control inspection technician', 'gis/cad engineer',
'senior web developer', 'sql developer', 'research staff member',
'sap abap associate consultant', 'associate qa',
'corporate recruiter', 'project management officer',
'business systems analyst', 'software programmer',
'help desk technician', 'sales manager', 'catalog associate',
'assistant store manager', 'software engg', 'it developer',
'apprentice', 'business consultant', 'controls engineer',
'ruby on rails developer', 'risk consultant', 'account manager',
'professor', 'assistant administrator', 'civil engineer',
'educator', 'service manager', 'teradata dba',
'full-time loss prevention associate', 'junior recruiter',
'associate developer', 'assistant electrical engineer',
'shift engineer', 'dotnet developer', 'rf/dt engineer',
'human resources analyst', 'software test engineerte',
'junior .net developer', 'java trainee', 'maintenance supervisor',
'r&d engineer', 'front end developer', 'engineer-hws',
'operations engineer', 'senior research fellow',
'web designer and joomla administrator',
'enterprise solutions developer',
'information technology specialist', 'site engineer',
'graduate trainee engineer', 'quality assurance analyst',
```

```
'cnc programmer', 'financial analyst', 'system engineer trainee',
'sap mm consultant', 'assistant system engineer trainee',
'qa trainee', 'teradata developer', 'hr executive',
'senior programmer', 'software test engineer (etl)',
'associate software engg', 'supply chain analyst', 'sales trainer',
'software executive', 'team leader',
'assistant system engineer - trainee', 'seo analyst',
'risk investigator', 'executive administrative assistant',
'program manager', 'r & d', 'sap functional consultant',
'website developer/tester', 'software designer',
'sales coordinator', 'qa engineer', 'aircraft technician',
'customer care executive', 'senior test engineer',
'program analyst trainee', 'electrical controls engineer',
'trainee decision scientist', 'editor', 'bss engineer', 'dba',
'software eng', 'computer faculty', 'recruitment associate',
'logistics executive', 'quality consultant',
'senior sales executive', 'db2 dba', 'test technician',
'it operations associate', 'software engineering associate',
'research scientist', 'jr. software developer'], dtype=object)
```

[18]:
```
# displays the value counts in the dataframe of column Designation
df_data['Designation'].value_counts()
```

[18]:
```
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
```

[19]:
```
# count method displays the row count
df_data['JobCity'].count()
```

[19]:
```
3998
```

[20]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪JobCity
df_data['JobCity'].nunique()
```

[20]:
```
339
```

[21]:
```
# displays the unique string in the column JobCity
df_data['JobCity'].unique()
```

[21]:
```
array(['Bangalore', 'Indore', 'Chennai', 'Gurgaon', 'Manesar',
       'Hyderabad', 'Banglore', 'Noida', 'Kolkata', 'Pune', -1, 'mohali',
       'Jhansi', 'Delhi', 'Hyderabad ', 'Bangalore ', 'noida', 'delhi',
       'Bhubaneswar', 'Navi Mumbai', 'Mumbai', 'New Delhi', 'Mangalore',
       'Rewari', 'Gaziabaad', 'Bhiwadi', 'Mysore', 'Rajkot',
       'Greater Noida', 'Jaipur', 'noida ', 'HYDERABAD', 'mysore',
       'THANE', 'Maharajganj', 'Thiruvananthapuram', 'Punchkula',
       'Bhubaneshwar', 'Pune ', 'coimbatore', 'Dhanbad', 'Lucknow',
       'Trivandrum', 'kolkata', 'mumbai', 'Gandhi Nagar', 'Una',
       'Daman and Diu', 'chennai', 'GURGOAN', 'vsakhapttnam', 'pune',
       'Nagpur', 'Bhagalpur', 'new delhi - jaisalmer', 'Coimbatore',
       'Ahmedabad', 'Kochi/Cochin', 'Bankura', 'Bengaluru', 'Mysore ',
       'Kanpur ', 'jaipur', 'Gurgaon ', 'bangalore', 'CHENNAI',
       'Vijayawada', 'Kochi', 'Beawar', 'Alwar', 'NOIDA', 'Greater noida',
       'Siliguri ', 'raipur', 'gurgaon', 'Bhopal', 'Faridabad', 'Jodhpur',
       'udaipur', 'Muzaffarpur', 'Kolkata`', 'Bulandshahar', 'Haridwar',
       'Raigarh', 'Visakhapatnam', 'Jabalpur', 'hyderabad', 'Unnao',
       'KOLKATA', 'Thane', 'Aurangabad', 'Belgaum', 'gurgoan', 'Dehradun',
       'Rudrapur', 'Jamshedpur', 'vizag', 'Nouda', 'Dharamshala',
       'Banagalore', 'Hissar', 'Ranchi', 'BANGALORE', 'Madurai', 'Gurga',
       'Chandigarh', 'Australia', ' Chennai', 'CHEYYAR', 'Mumbai ',
       'sonepat', 'Ghaziabad', 'Pantnagar', 'Siliguri', 'mumbai ',
       'Jagdalpur', 'Chennai ', 'angul', 'Baroda', ' ariyalur', 'Jowai',
       'Kochi/Cochin, Chennai and Coimbatore', 'bhubaneswar', 'Neemrana',
       'VIZAG', 'Tirupathi', 'Lucknow ', 'Ahmedabad ', 'Bhubneshwar',
       'Noida ', 'pune ', 'Calicut', 'Gandhinagar', 'LUCKNOW', 'Dubai',
       'bengaluru', 'MUMBAI', 'Ahmednagar', 'Nashik', 'New delhi',
       'Bellary', 'Ludhiana', 'New Delhi ', 'Muzaffarnagar', 'BHOPAL',
       'Gurgoan', 'Gagret', 'Indirapuram, Ghaziabad', 'Gwalior',
       'new delhi', 'TRIVANDRUM', 'Chennai & Mumbai', 'Rajasthan',
       'Sonipat', 'Bareli', 'Kanpur', 'Hospete', 'Miryalaguda', ' mumbai',
       'Dharuhera', 'lucknow', 'meerut', 'dehradun', 'Ganjam', 'Hubli',
       'bangalore ', 'NAVI MUMBAI', 'ncr', 'Agra', 'Trichy',
       'kudankulam ,tarapur', 'Ongole', 'Sambalpur', 'Pondicherry',
       'Bundi', 'SADULPUR,RAJGARH,DISTT-CHURU,RAJASTHAN', 'AM', 'Bikaner',
       'Vadodara', 'BAngalore', 'india', 'Asansol', 'Tirunelvelli',
       'Ernakulam', 'DELHI', 'Bilaspur', 'Chandrapur', 'Nanded',
       'Dharmapuri', 'Vandavasi', 'Rohtak', 'trivandrum', 'Nagpur ',
       'Udaipur', 'Patna', 'banglore', 'indore', 'Salem', 'Nasikcity',
       'Gandhinagar ', 'Technopark, Trivandrum', 'Bharuch', 'Tornagallu',
       'Raipur', 'Kolkata ', 'Jaspur', 'Burdwan', 'Bhubaneswar ',
       'Shimla', 'ahmedabad', 'Gajiabaad', 'Jammu', 'Shahdol',
       'Muvattupuzha', 'Al Jubail,Saudi Arabia', 'Kalmar, Sweden',
       'Secunderabad', 'A-64,sec-64,noida', 'Ratnagiri', 'Jhajjar',
```

```
       'Gulbarga', 'hyderabad(bhadurpally)', 'Nalagarh', 'Chandigarh ',
       'Jaipur ', 'Jeddah Saudi Arabia', ' Delhi', 'PATNA', 'SHAHDOL',
       'Chennai, Bangalore', 'Bhopal ', 'Jamnagar', 'PUNE', 'Tirupati',
       'Gonda', 'jamnagar', 'chennai ', 'orissa', 'kharagpur',
       'Trivandrum ', 'Navi Mumbai , Hyderabad', 'Joshimath',
       'chandigarh', 'Bathinda', 'Johannesburg', 'kala amb ', 'Karnal',
       'LONDON', 'Kota', 'Panchkula', 'Baddi HP', 'Nagari',
       'Mettur, Tamil Nadu ', 'Durgapur', 'pondi', 'Surat', 'Kurnool',
       'kolhapur', 'Madurai ', 'GREATER NOIDA', 'Bhilai', ' Pune',
       'hderabad', 'KOTA', 'thane', 'Vizag', 'Bahadurgarh',
       'Rayagada, Odisha', 'kakinada', 'GURGAON', 'Varanasi', 'punr',
       'Nellore', 'patna', 'Meerut', 'hyderabad ', 'Sahibabad', 'Howrah',
       'BHUBANESWAR', 'Trichur', 'Ambala', 'Khopoli', 'keral', 'Roorkee',
       'Greater NOIDA', 'Navi mumbai', 'ghaziabad', 'Allahabad',
       'Delhi/NCR', 'Panchkula ', 'Ranchi ', 'Jalandhar', 'manesar',
       'vapi', 'PILANI', 'muzzafarpur', 'RAS AL KHAIMAH', 'bihar',
       'singaruli', 'KANPUR', 'Banglore ', 'pondy', 'Mohali', 'Phagwara',
       ' Mumbai', ' bangalore', 'GURAGAON', 'Baripada', 'MEERUT',
       'Yamuna Nagar', 'shahibabad', 'sampla', 'Guwahati', 'Rourkela',
       'Banaglore', 'Vellore', 'Dausa', 'latur (Maharashtra )',
       'NEW DELHI', 'kanpur', 'Mainpuri', 'karnal', 'Dammam', 'Haldia',
       'sambalpur', 'RAE BARELI', 'ranchi', 'jAipur', 'BANGLORE',
       'Patiala', 'Gorakhpur', 'new dehli', 'BANGALORE ', 'Ambala City',
       'Karad', 'Rajpura', 'Pilani', 'haryana', 'Asifabadbanglore'],
      dtype=object)
```

[22]:
```
# displays the value counts in the dataframe of column JobCity
df_data['JobCity'].value_counts()
```

[22]:
```
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
```

[23]:
```
# count method displays the row count
df_data['Gender'].count()
```

[23]:
```
3998
```

[24]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪Gender
df_data['Gender'].nunique()
```

[24]:
```
2
```

[25]:
```
# displays the unique string in the column Gender
df_data['Gender'].unique()
```

[25]:
```
array(['f', 'm'], dtype=object)
```

[26]:
```
# displays the value counts in the dataframe of column Gender
df_data['Gender'].value_counts()
```

[26]:
```
Gender
m    3041
f     957
Name: count, dtype: int64
```

[27]:
```
# count method displays the row count
df_data['10board'].count()
```

[27]:
```
3998
```

[28]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪10board
df_data['10board'].nunique()
```

[28]:
```
275
```

[29]:
```
# displays the value counts in the dataframe of column 10board
df_data['10board'].value_counts()
```

[29]:
```
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
```

[30]:
```
# count method displays the row count
df_data['12board'].count()
```

[30]: 3998

[31]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪12board
df_data['12board'].nunique()
```

[31]: 340

[32]:
```
# displays the unique string in the column 12board
df_data['12board'].unique()
```

[32]:
```
array(['board of intermediate education,ap', 'cbse', 'state board',
       'mp board', 'isc', 'icse', 'karnataka pre university board', 'up',
       'p u board, karnataka', 'dept of pre-university education', 'bie',
       'kerala state hse board', 'up board', 0, 'bseb', 'chse', 'puc',
       ' upboard',
       'state  board of intermediate education, andhra pradesh',
       'karnataka state board',
       'west bengal state council of technical education', 'wbchse',
       'maharashtra state board', 'ssc', 'isc board',
       'sda matric higher secondary school', 'uttar pradesh board', 'ibe',
       'chsc', 'board of intermediate', 'isce', 'upboard', 'sbtet',
       'hisher seconadry examination(state board)', 'pre university',
       'borad of intermediate', 'j & k board',
       'intermediate board of andhra pardesh', 'rbse',
       'central board of secondary education', 'jkbose', 'hbse',
       'board of intermediate education', 'state', 'ms board', 'pue',
       'intermediate state board', 'stateboard', 'hsc',
       'electonincs and communication(dote)', 'karnataka pu board',
       'government polytechnic mumbai , mumbai board', 'pu board',
       'baord of intermediate education', 'apbie', 'andhra board',
       'tamilnadu stateboard',
       'west bengal council of higher secondary education',
       'cbse,new delhi', 'u p board', 'intermediate', 'biec,patna',
       'diploma in engg (e &tc) tilak maharashtra vidayapeeth',
       'hsc pune', 'pu board karnataka', 'kerala', 'gsheb',
       'up(allahabad)', 'nagpur', 'st joseph hr sec school',
       'pre university board', 'ipe', 'maharashtra', 'kea', 'apsb',
       'himachal pradesh board of school education', 'staae board',
       'international baccalaureate (ib) diploma', 'nios',
       'karnataka board of university',
       'board of secondary education rajasthan', 'uttarakhand board',
       'ua', 'scte vt orissa', 'matriculation',
       'department of pre-university education', 'wbscte',
```

'preuniversity board(karnataka)', 'jharkhand accademic council',
'bieap', 'msbte (diploma in computer technology)',
'jharkhand acamedic council (ranchi)',
'department of pre-university eduction', 'biec', 'all india board',
'sjrcw', ' board of intermediate', 'msbte',
'sri sankara vidyalaya', 'chse, odisha', 'bihar board',
'maharashtra state(latur board)', 'rajasthan board', 'mpboard',
'state board of technical eduction panchkula', 'upbhsie', 'apbsc',
'state board of technical education and training',
'secondary board of rajasthan',
'tamilnadu higher secondary education board',
'jharkhand academic council',
'board of intermediate education,hyderabad', 'up baord', 'pu',
'dte', 'board of secondary education', 'pre-university',
'board of intermediate education,andhra pradesh',
'up board , allahabad', 'srv girls higher sec school,rasipuram',
'intermediate board of education,andhra pradesh',
'intermediate board examination',
'department of pre-university education, bangalore',
'stmiras college for girls', 'mbose',
'department of pre-university education(government of karnataka)',
'dpue', 'msbte pune', 'board of school education harayana',
'sbte, jharkhand', 'bihar intermediate education council, patna',
'higher secondary', 's j polytechnic', 'latur',
'board of secondary education, rajasthan', 'jyoti nivas', 'pseb',
'biec-patna', 'board of intermediate education,andra pradesh',
'chse,orissa', 'pre-university board', 'mp', 'intermediate board',
'govt of karnataka department of pre-university education',
'karnataka education board',
'board of secondary school of education', 'pu board ,karnataka',
'karnataka secondary education board', 'karnataka sslc',
'board of intermediate ap', 'u p', 'state board of karnataka',
'directorate of technical education,banglore', 'matric board',
'andhpradesh board of intermediate education',
'stjoseph of cluny matrhrsecschool,neyveli,cuddalore district',
'bte up', 'scte and vt ,orissa', 'hbsc',
'jawahar higher secondary school', 'nagpur board', 'bsemp',
'board of intermediate education, andhra pradesh',
'board of higher secondary orissa',
'board of secondary education,rajasthan(rbse)',
'board of intermediate education:ap,hyderabad', 'science college',
'karnatak pu board', 'aissce', 'pre university board of karnataka',
'bihar', 'kerala state board', 'uo board', 'cicse',
'karnataka board', 'tn state board',
'kolhapur divisional board, maharashtra',
'jaycee matriculation school',
'board of higher secondary examination, kerala',

'uttaranchal state board', 'intermidiate', 'bciec,patna', 'bice',
'karnataka state', 'state broad', 'wbbhse', 'gseb',
'uttar pradesh', 'ghseb', 'board of school education uttarakhand',
'gseb/technical education board', 'msbshse,pune',
'tamilnadu state board', 'board of technical education',
'kerala university', 'uttaranchal shiksha avam pariksha parishad',
'chse(concil of higher secondary education)',
'bright way college, (up board)', 'board of intermidiate',
'higher secondary state certificate', 'karanataka secondary board',
'maharashtra board', 'andhra pradesh state board', 'cgbse',
'diploma in computers', 'bte,delhi', 'rajasthan board ajmer',
'mpbse', 'pune board', 'state board of technical education',
'gshseb', 'amravati divisional board',
'dote (diploma - computer engg)', 'up bord',
'karnataka pre-university board', 'jharkhand board',
'punjab state board of technical education & industrial training',
'department of technical education',
'sri chaitanya junior kalasala', 'state board (jac, ranchi)',
'gujarat board', 'aligarh muslim university',
'tamil nadu state board', 'hse', 'karnataka secondary education',
'state board ', 'karnataka pre unversity board',
'ks rangasamy institute of technology',
'karnataka board secondary education', 'narayana junior college',
'bteup', 'board of intermediate(bie)', 'hsc maharashtra board',
'tamil nadu state', 'uttrakhand board', 'psbte',
'stateboard/tamil nadu', 'intermediate council patna',
'technical board, punchkula', 'board of intermidiate examination',
'sri kannika parameswari highier secondary school, udumalpet',
'ap board', 'nashik board', 'himachal pradesh board',
'maharashtra satate board',
'andhra pradesh board of secondary education',
'tamil nadu polytechnic',
'maharashtra state board mumbai divisional board',
'department of pre university education',
'dav public school,hehal', 'board of intermediate education, ap',
'rajasthan board of secondary education',
'department of technical education, bangalore', 'chse,odisha',
'maharashtra nasik board',
'west bengal council of higher secondary examination (wbchse)',
'holy cross matriculation hr sec school', 'cbsc',
'pu board karnataka', 'biec patna', 'kolhapur', 'bseb, patna',
'up board allahabad', 'intermideate', 'nagpur board,nagpur',
'diploma(msbte)', 'dav public school',
'pre university board, karnataka', 'ssm srsecschool', 'state bord',
'jstb,jharkhand', 'intermediate board of education',
'mp board bhopal', 'pub', 'madhya pradesh board',
'bihar intermediate education council',

```
       'west bengal council of higher secondary eucation',
       'isc board , new delhi', 'mpc',
       'certificate for higher secondary education (chse)orissa',
       'maharashtra state board for hsc',
       'board of intermeadiate education', 'latur board',
       'andhra pradesh', 'karnataka pre-university',
       'lucknow public college', 'nagpur divisional board',
       'ap intermediate board', 'cgbse raipur', 'uttranchal board',
       'jiec', 'central board of secondary education, new delhi',
       'bihar school examination board patna',
       'state board of technical education harayana', 'mp-bse',
       'up bourd', 'dav public school sec 14',
       'haryana state board of technical education chandigarh',
       'council for indian school certificate examination',
       'jaswant modern school', 'madhya pradesh open school',
       'aurangabad board', 'j&k state board of school education',
       'diploma ( maharashtra state board of technical education)',
       'board of technicaleducation ,delhi',
       'maharashtra state boar of secondary and higher secondary education',
       'hslc (tamil nadu state board)',
       'karnataka state examination board', 'puboard', 'nasik',
       'west bengal board of higher secondary education',
       'up board,allahabad', 'board of intrmediate education,ap', 'cbese',
       'karnataka state pre- university board',
       'state board - west bengal council of higher secondary education :
wbchse',
       'maharashtra state board of secondary & higher secondary education',
       'biec, patna', 'state syllabus', 'cbse board', 'scte&vt',
       'board of intermediate,ap',
       'secnior secondary education board of rajasthan',
       'maharashtra board, pune', 'rbse (state board)',
       'board of intermidiate education,ap',
       'board of high school and intermediate education uttarpradesh',
       'higher secondary education',
       'board fo intermediate education, ap', 'intermedite',
       'ap board for intermediate education', 'ahsec',
       'punjab state board of technical education & industrial training,
chandigarh',
       'state board - tamilnadu', 'jharkhand acedemic council',
       'scte & vt (diploma)', 'karnataka pu',
       'board of intmediate education ap', 'up-board',
       'boardofintermediate'], dtype=object)
```

[33]:
```
# displays the value counts in the dataframe of column 12board
df_data['12board'].value_counts()
```

[33]:
```
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
```

[34]:
```
# count method displays the row count
df_data['Degree'].count()
```

[34]:
```
3998
```

[35]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪Degree
df_data['Degree'].nunique()
```

[35]:
```
4
```

[36]:
```
# displays the unique string in the column Degree
df_data['Degree'].unique()
```

[36]:
```
array(['B.Tech/B.E.', 'MCA', 'M.Tech./M.E.', 'M.Sc. (Tech.)'],
      dtype=object)
```

[37]:
```
# displays the value counts in the dataframe of column Degree
df_data['Degree'].value_counts()
```

[37]:
```
Degree
B.Tech/B.E.      3700
MCA               243
M.Tech./M.E.       53
M.Sc. (Tech.)       2
Name: count, dtype: int64
```

[38]:
```
# count method displays the row count
df_data['Specialization'].count()
```

[38]:
```
3998
```

[39]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪Specialization
df_data['Specialization'].nunique()
```

[39]:
```
46
```

[40]:
```
# displays the unique string in the column Specialization
df_data['Specialization'].unique()
```

[40]:
```
array(['computer engineering',
       'electronics and communication engineering',
       'information technology', 'computer science & engineering',
       'mechanical engineering', 'electronics and electrical engineering',
       'electronics & telecommunications',
       'instrumentation and control engineering', 'computer application',
       'electronics and computer engineering', 'electrical engineering',
       'applied electronics and instrumentation',
       'electronics & instrumentation eng',
       'information science engineering', 'civil engineering',
       'mechanical and automation', 'industrial & production engineering',
       'control and instrumentation engineering',
       'metallurgical engineering',
       'electronics and instrumentation engineering',
       'electronics engineering', 'ceramic engineering',
       'chemical engineering', 'aeronautical engineering', 'other',
       'biotechnology', 'embedded systems technology',
       'electrical and power engineering',
       'computer science and technology', 'mechatronics',
       'automobile/automotive engineering', 'polymer technology',
       'mechanical & production engineering',
       'power systems and automation', 'instrumentation engineering',
       'telecommunication engineering',
       'industrial & management engineering', 'industrial engineering',
       'computer and communication engineering',
       'information & communication technology', 'information science',
       'internal combustion engine', 'computer networking',
       'biomedical engineering', 'electronics', 'computer science'],
      dtype=object)
```

[41]:
```
# displays the value counts in the dataframe of column Specialization
df_data['Specialization'].value_counts()
```

[41]:
```
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
```

[42]:
```
# count method displays the row count
df_data['CollegeState'].count()
```

[42]:
```
3998
```

[43]:
```
# nunique method displays the number of unique values in a dataframe column␣
↪CollegeState
df_data['CollegeState'].nunique()
```

[43]:
```
26
```

[44]:
```
# displays the unique string in the column CollegeState
df_data['CollegeState'].unique()
```

[44]:
```
array(['Andhra Pradesh', 'Madhya Pradesh', 'Uttar Pradesh', 'Delhi',
       'Karnataka', 'Tamil Nadu', 'West Bengal', 'Maharashtra', 'Haryana',
       'Telangana', 'Orissa', 'Punjab', 'Kerala', 'Gujarat', 'Rajasthan',
       'Chhattisgarh', 'Uttarakhand', 'Jammu and Kashmir', 'Jharkhand',
       'Himachal Pradesh', 'Bihar', 'Assam', 'Goa', 'Sikkim',
       'Union Territory', 'Meghalaya'], dtype=object)
```

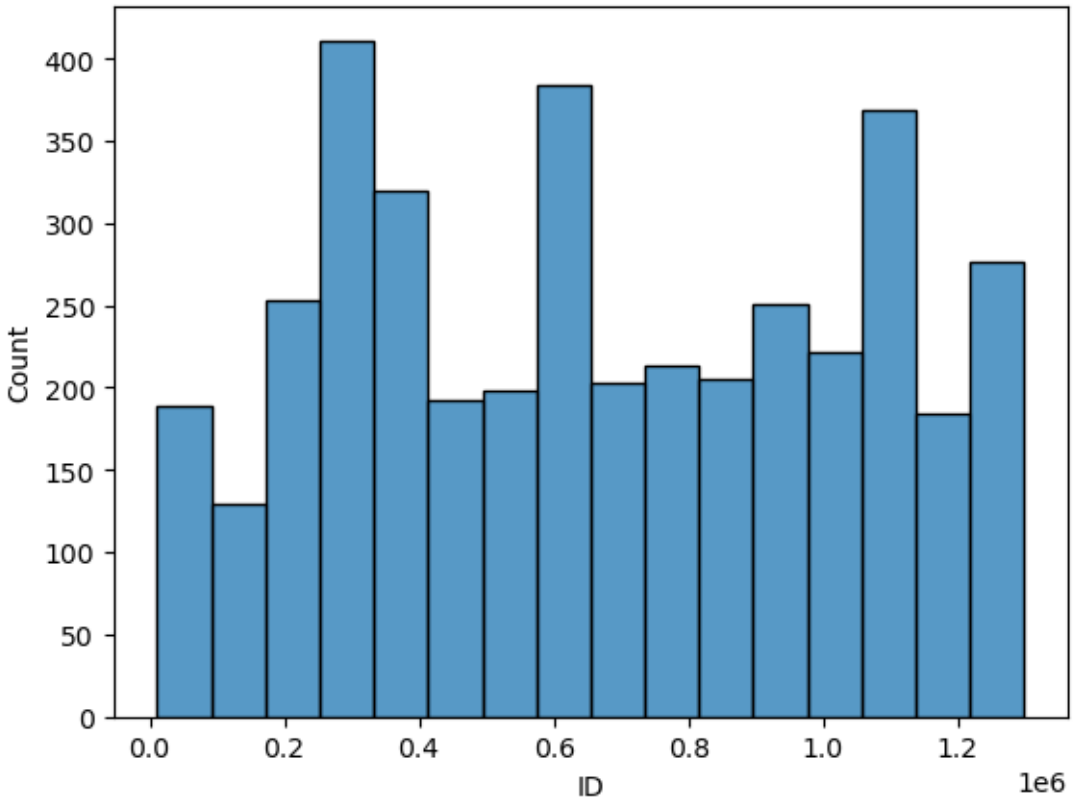
[45]:
```
# displays the value counts in the dataframe of column CollegeState
df_data['CollegeState'].value_counts()
```

[45]:
```
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
```

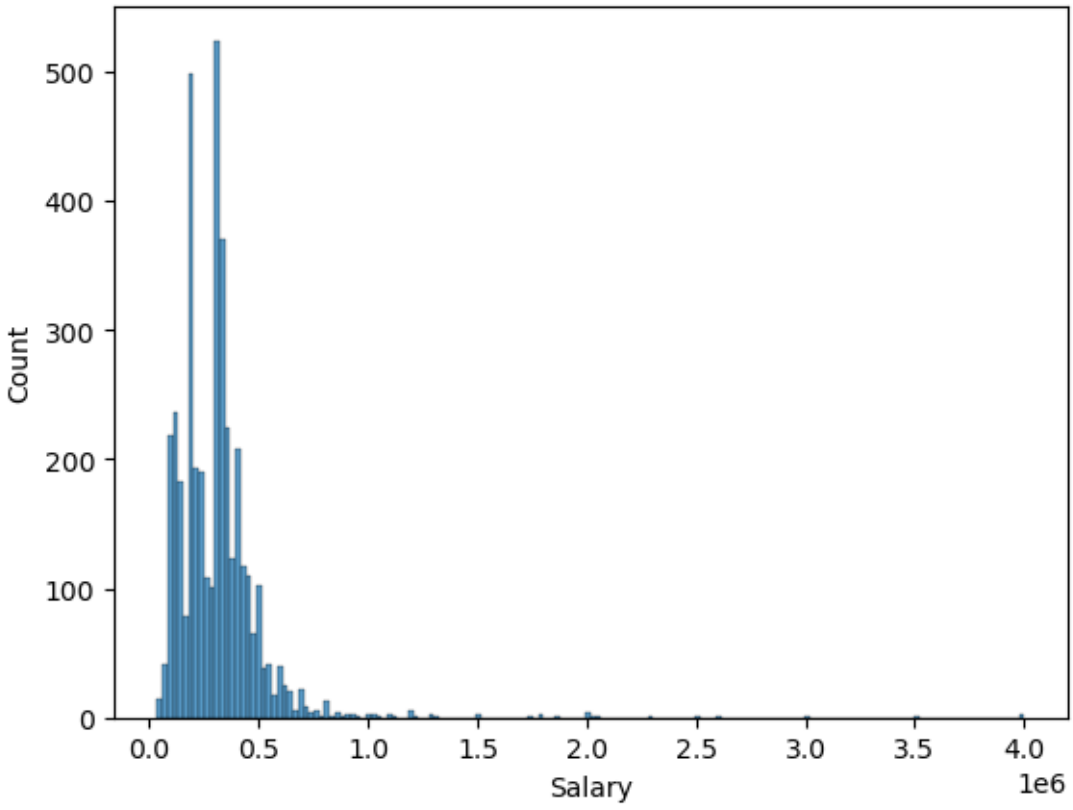
[46]:
```
# histogram plot for the numerical column ID
sns.histplot(df_data['ID'])
```

[46]:
```
<Axes: xlabel='ID', ylabel='Count'>
```



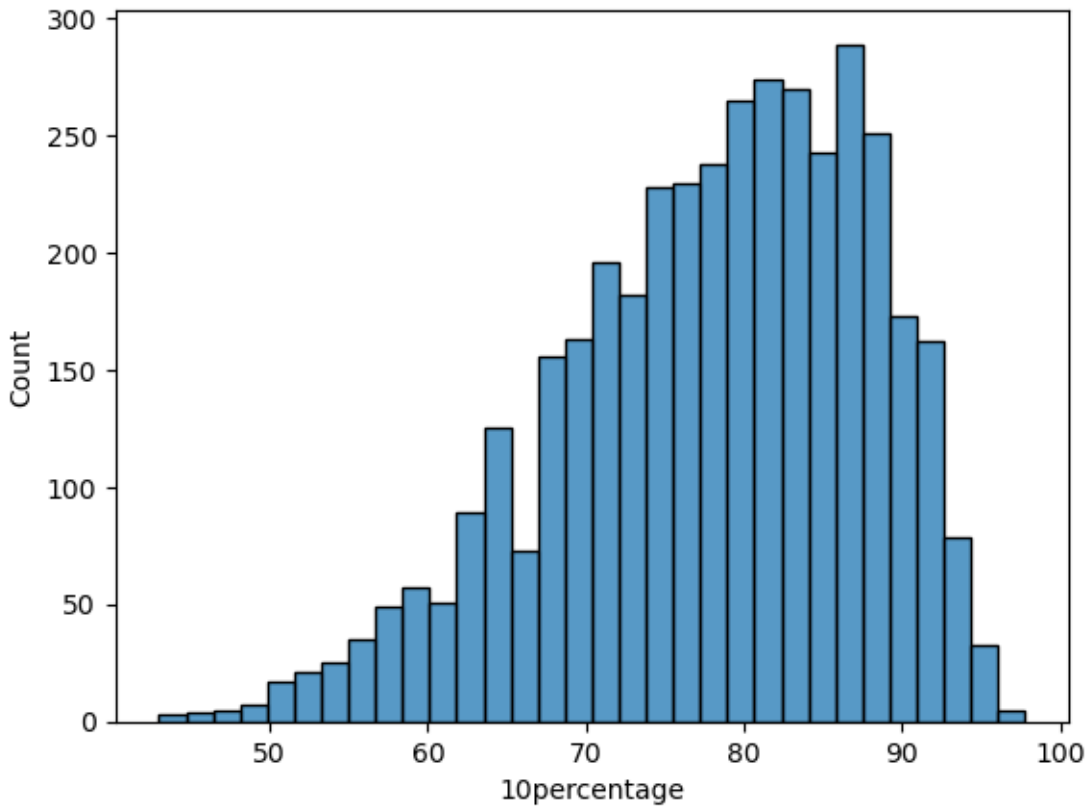
[47]:
```
# Histogram plot for the numerical column Salary
sns.histplot(df_data['Salary'])
```

[47]:
```
<Axes: xlabel='Salary', ylabel='Count'>
```



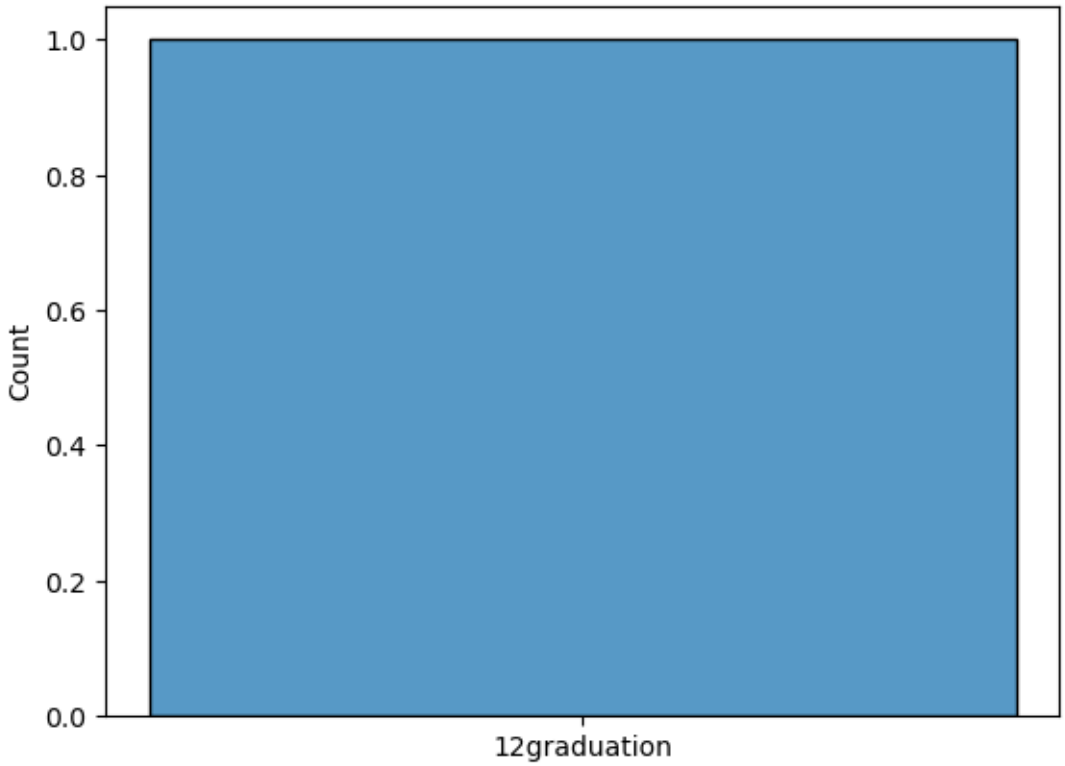
[48]:
```
# histogram plot for the column 10percentage
sns.histplot(df_data['10percentage'])
```

[48]: <Axes: xlabel='10percentage', ylabel='Count'>



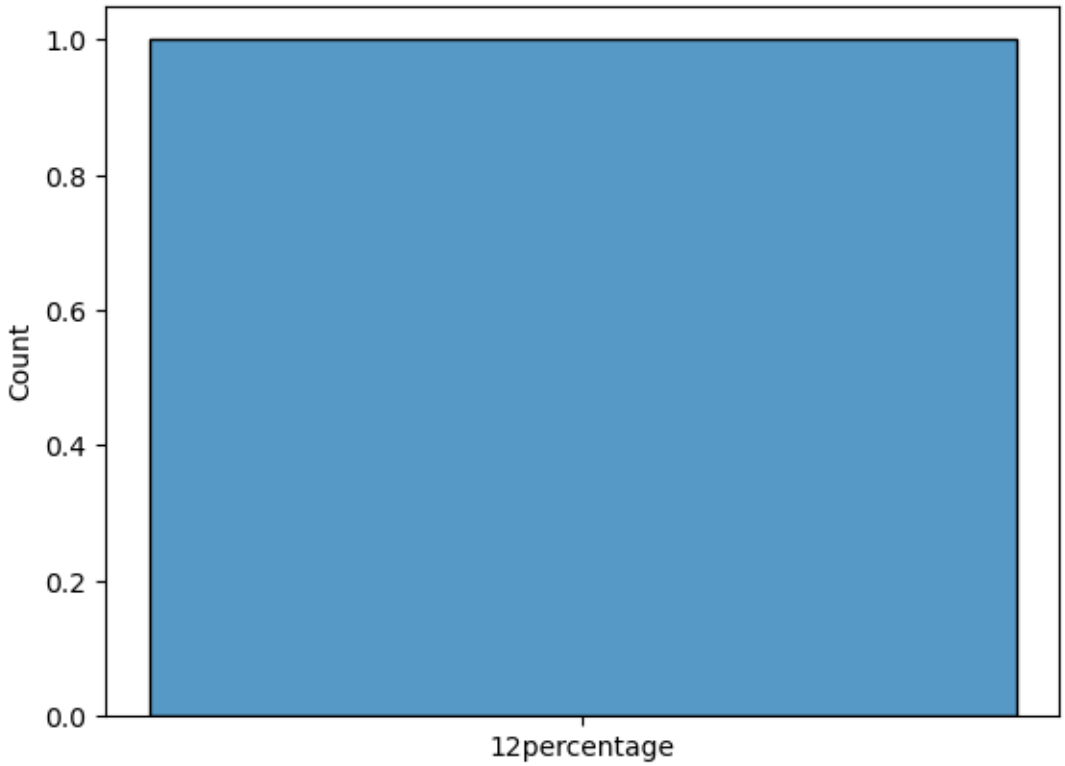
[49]:
```
# Histogram plot for the column 12graduation
sns.histplot('12graduation')
```

[49]: <Axes: ylabel='Count'>



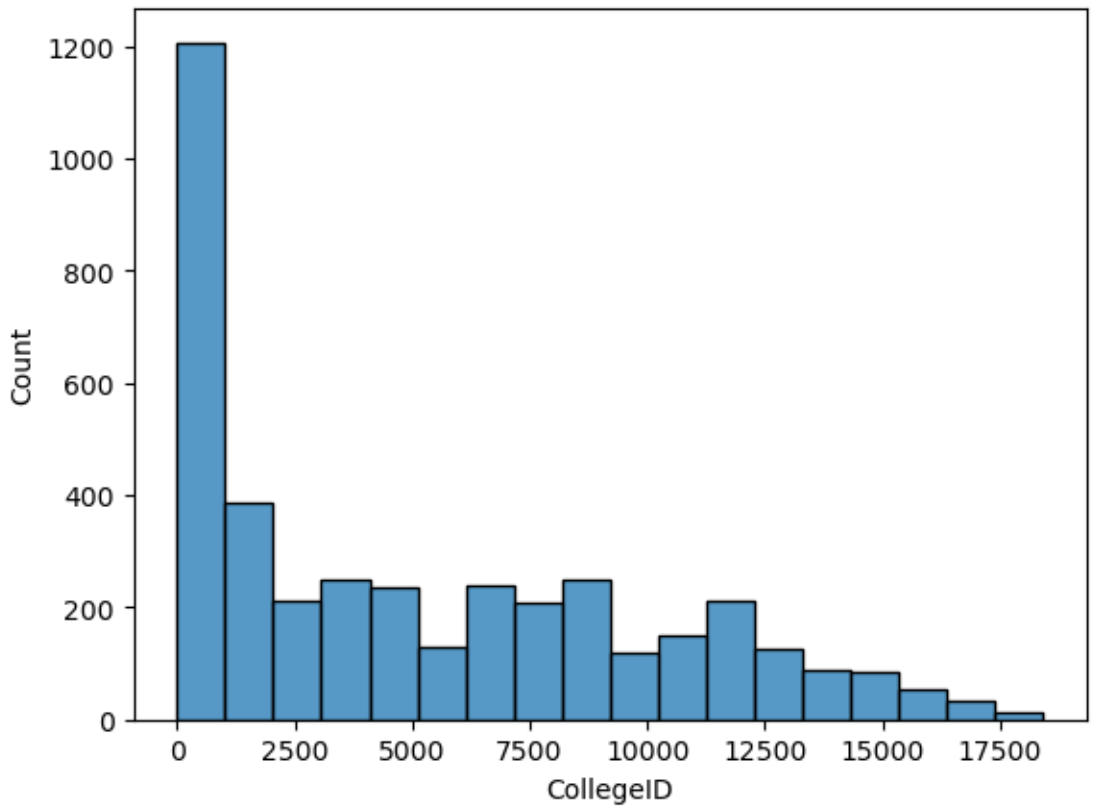
[50]:
```
# Histogram plot for the column 12percentage
sns.histplot('12percentage')
```

[50]: <Axes: ylabel='Count'>



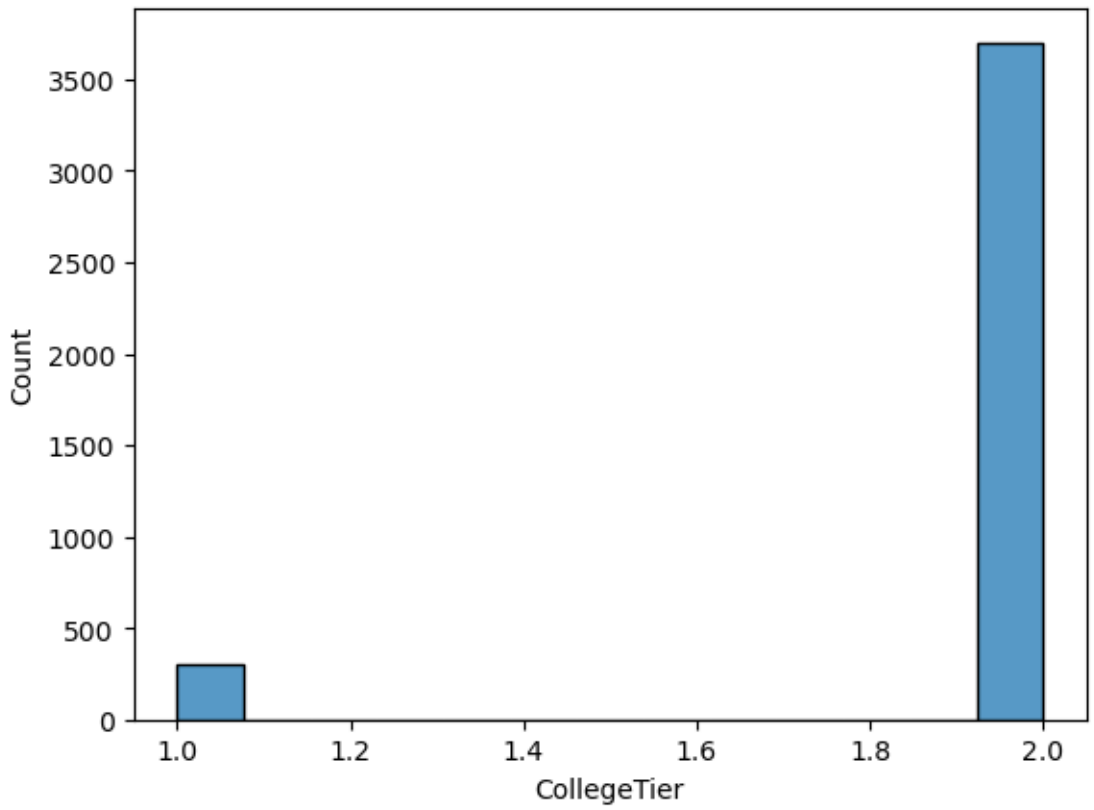
[51]:
```
# Histogram plot for the column CollegeID
sns.histplot(df_data['CollegeID'])
```

[51]:
```
<Axes: xlabel='CollegeID', ylabel='Count'>
```



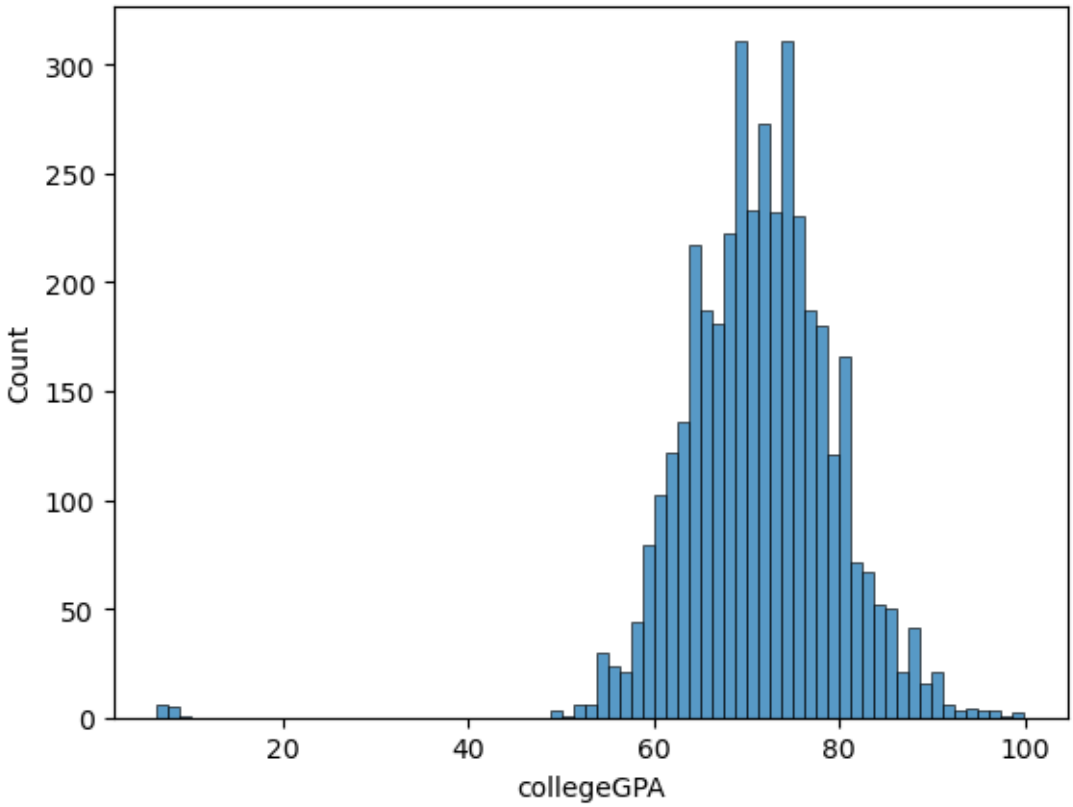
[52]:
```
# Histogram plot for the column CollegeTier
sns.histplot(df_data['CollegeTier'])
```

[52]: <Axes: xlabel='CollegeTier', ylabel='Count'>



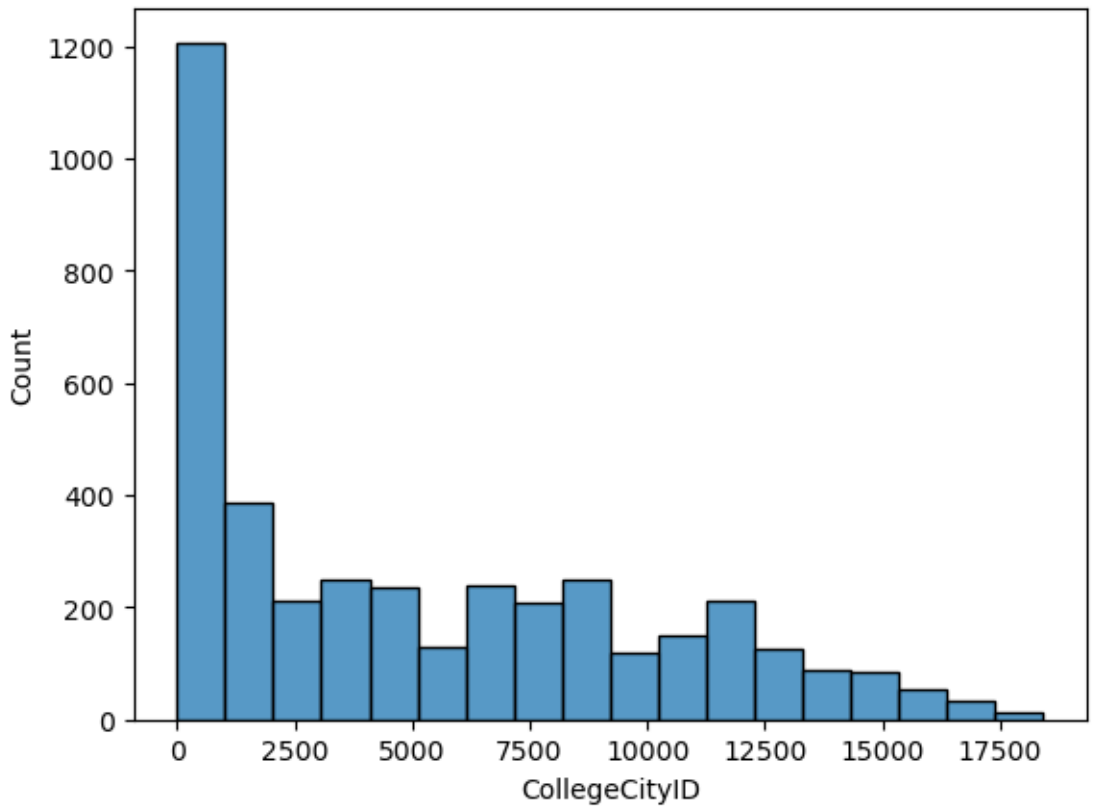
[53]:
```
# Histogram plot for the column collegeGPA
sns.histplot(df_data['collegeGPA'])
```

[53]:
```
<Axes: xlabel='collegeGPA', ylabel='Count'>
```



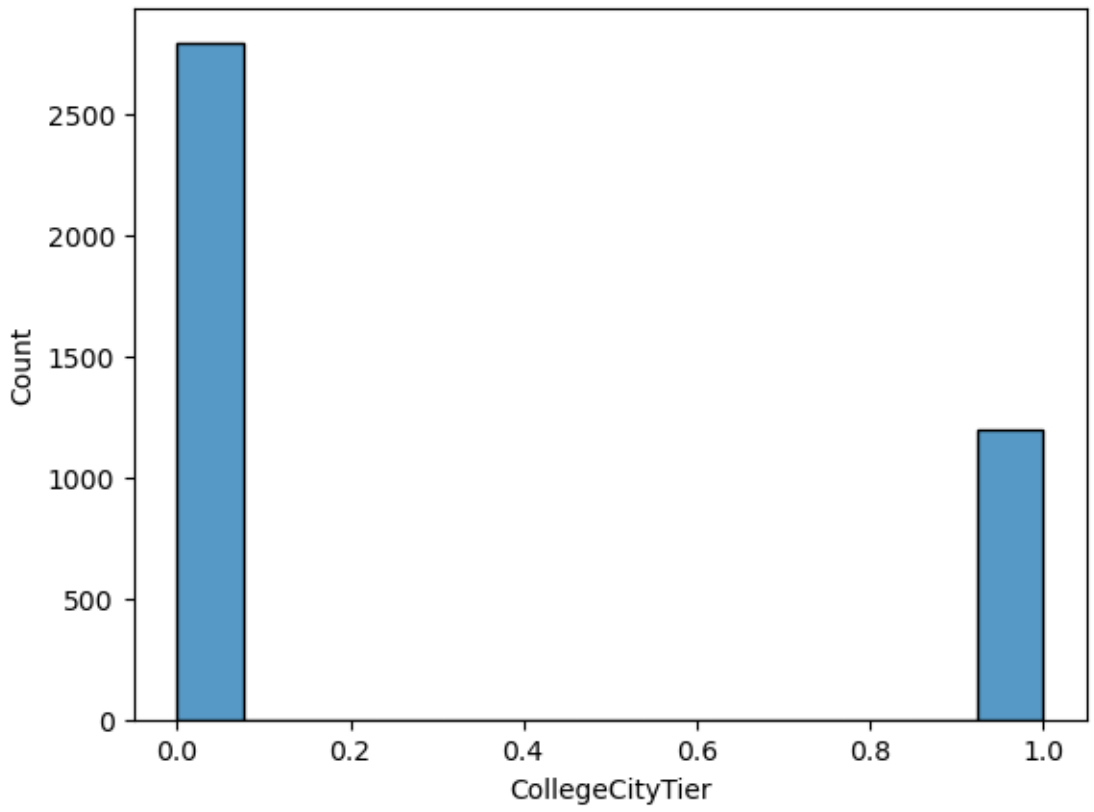
[54]:
```
# Histogram plot for the column CollegeCityID
sns.histplot(df_data['CollegeCityID'])
```

[54]: <Axes: xlabel='CollegeCityID', ylabel='Count'>



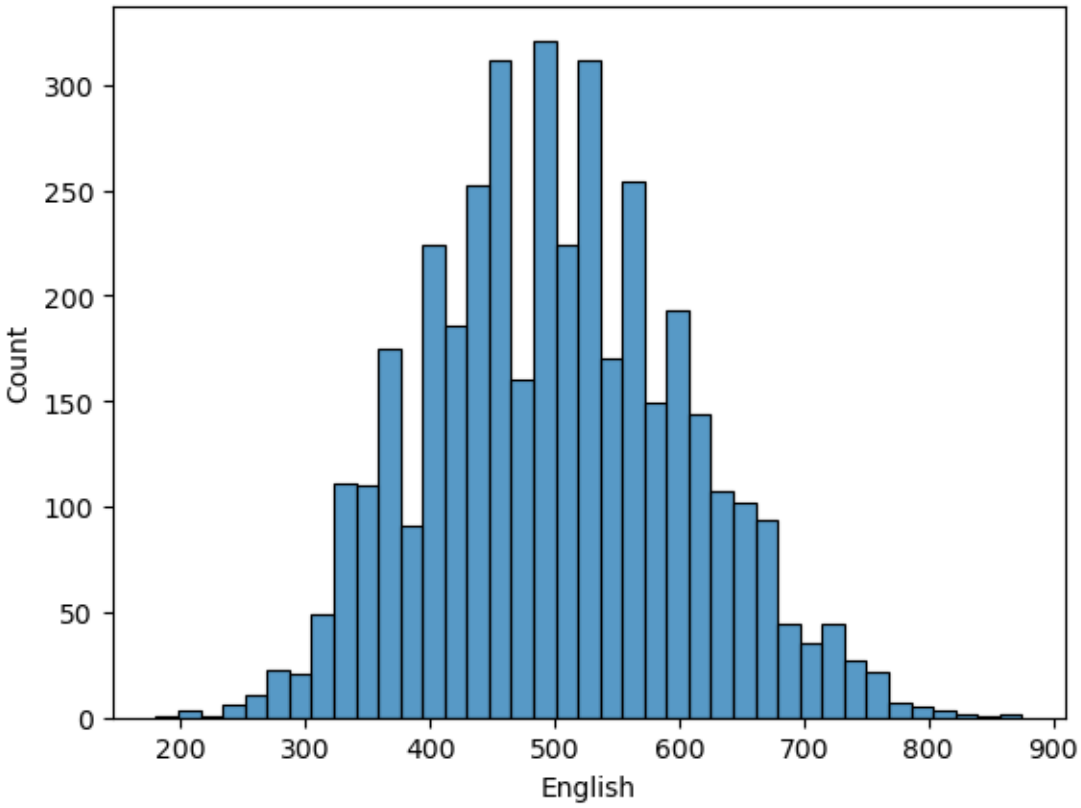
[55]:
```
# Histogram plot for the column CollegeCityTier
sns.histplot(df_data['CollegeCityTier'])
```

[55]: <Axes: xlabel='CollegeCityTier', ylabel='Count'>



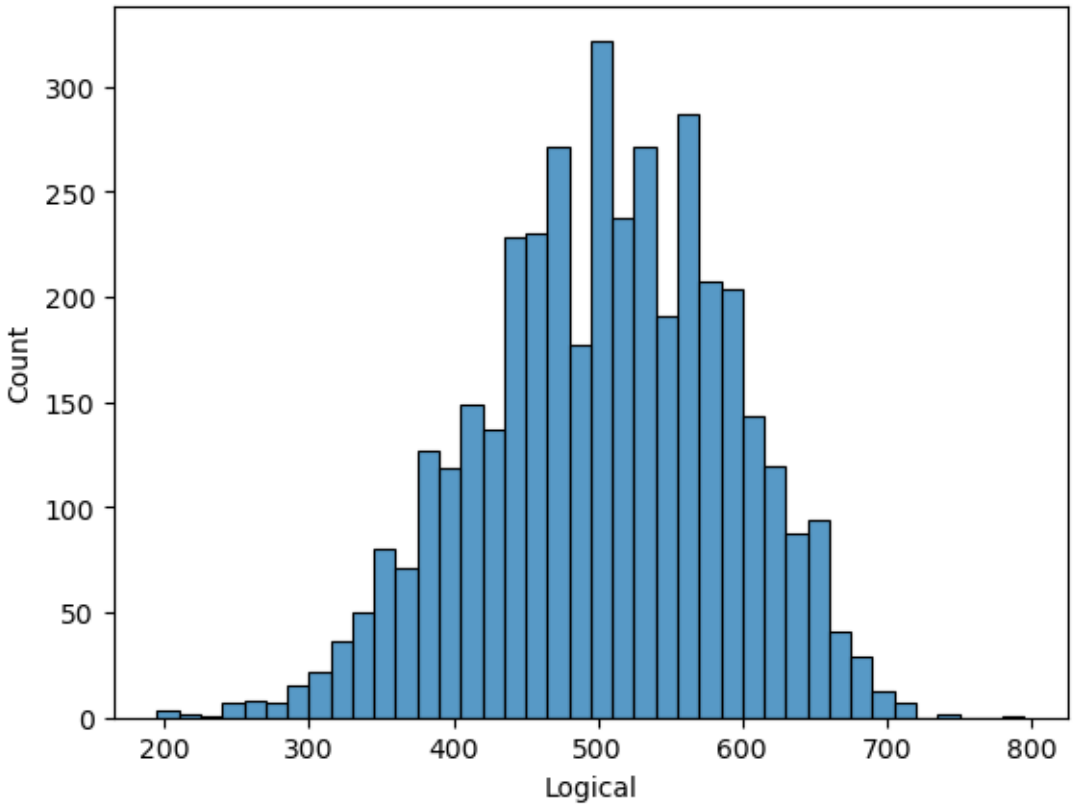
[56]:
```
# Histogram plot for the column English
sns.histplot(df_data['English'])
```

[56]: <Axes: xlabel='English', ylabel='Count'>



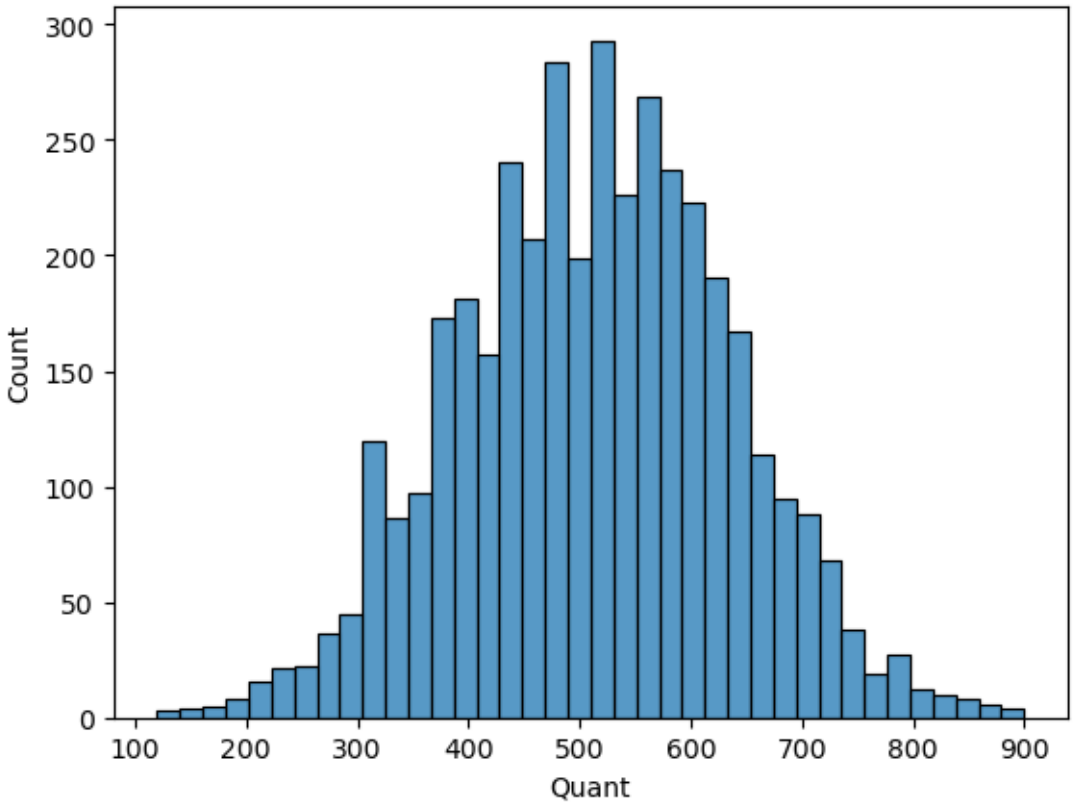
[57]:
```
# Histogram plot for the column Logical
sns.histplot(df_data['Logical'])
```

[57]: <Axes: xlabel='Logical', ylabel='Count'>



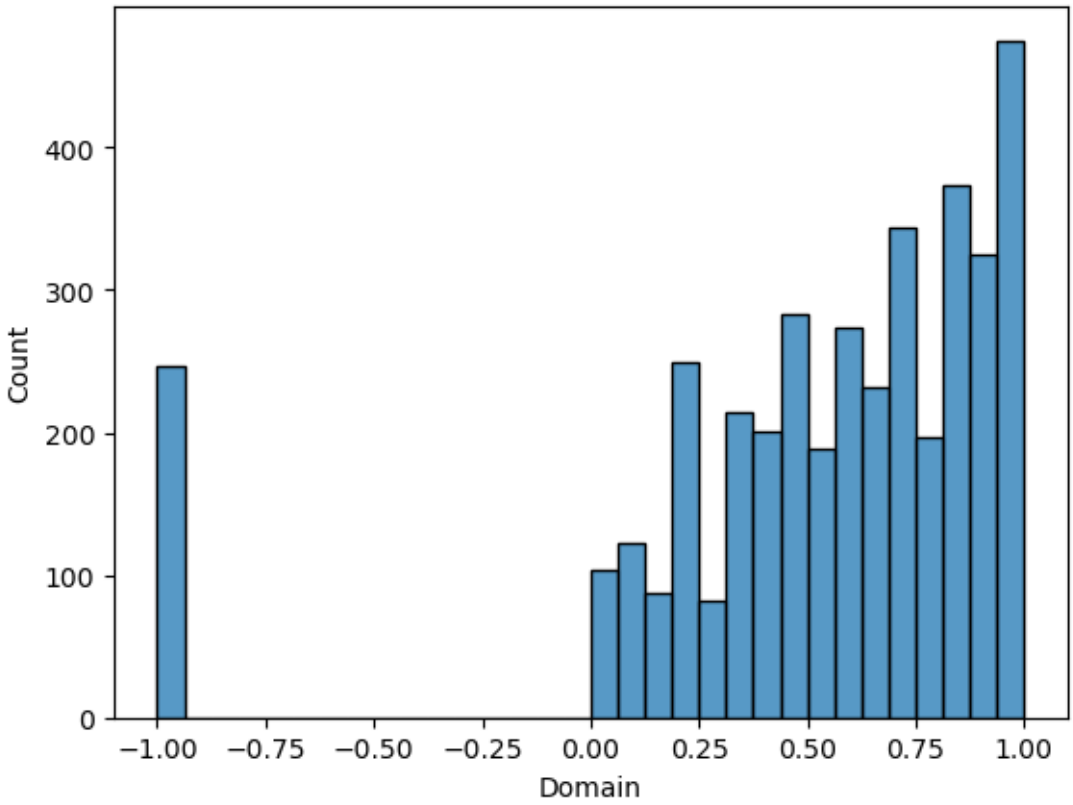
[58]:
```
# Histogram plot for the column Quant
sns.histplot(df_data['Quant'])
```

[58]: <Axes: xlabel='Quant', ylabel='Count'>



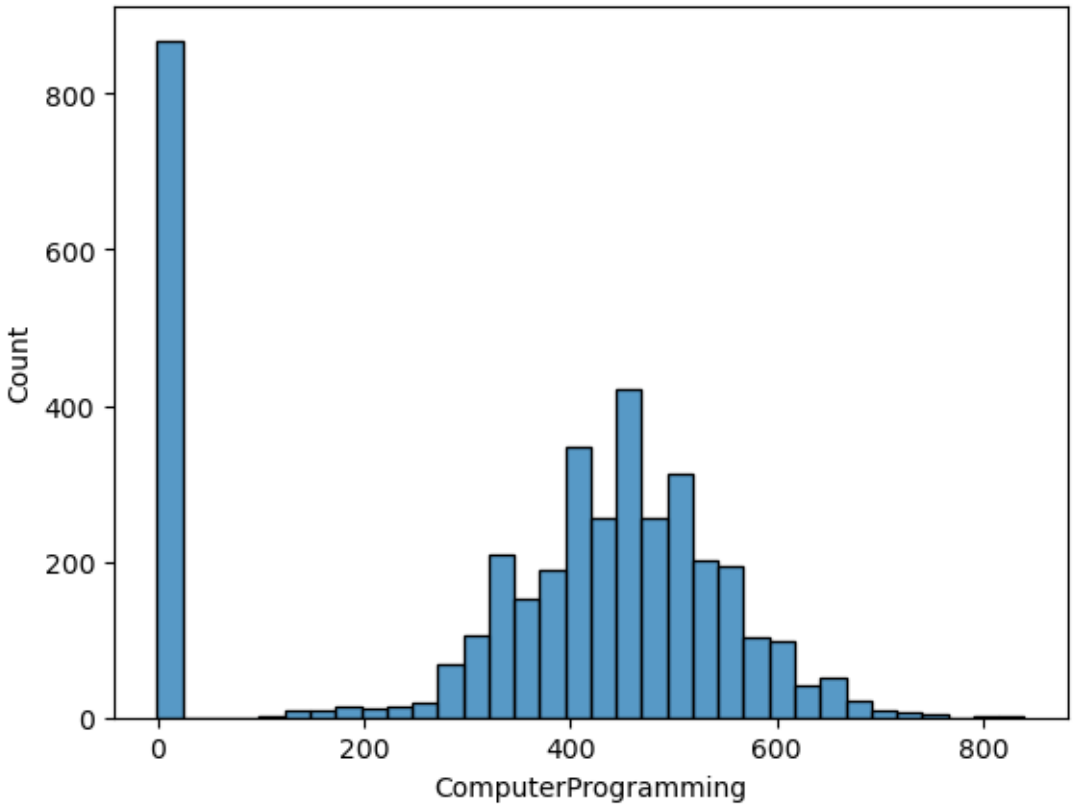
[59]:
```
# Histogram plot for the column Domain
sns.histplot(df_data['Domain'])
```

[59]: <Axes: xlabel='Domain', ylabel='Count'>



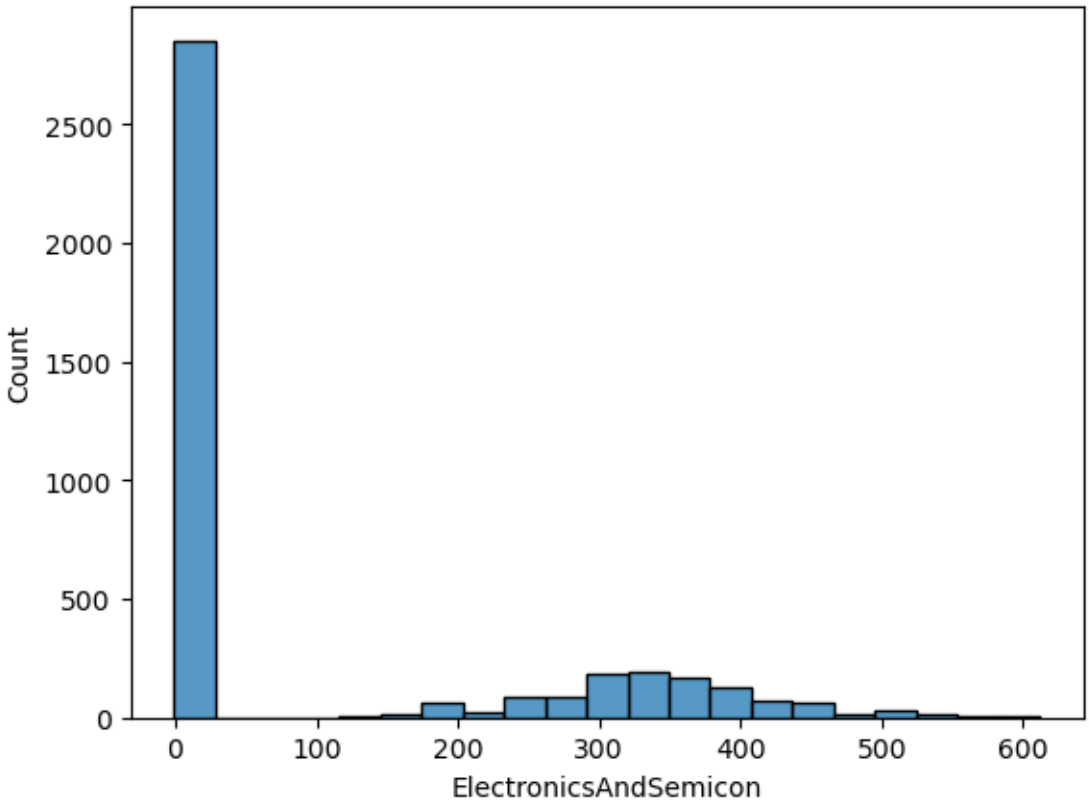
[60]:
```
# Histogram plot for the column ComputerEngineering
sns.histplot(df_data['ComputerProgramming'])
```

[60]: <Axes: xlabel='ComputerProgramming', ylabel='Count'>



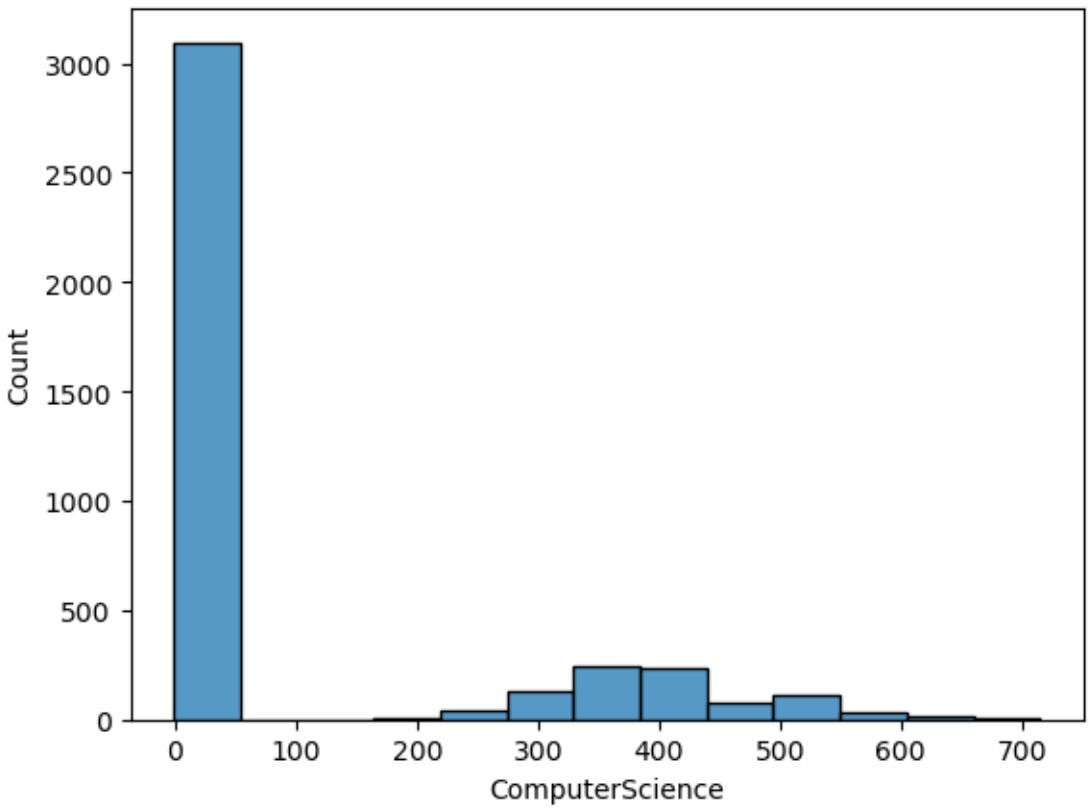
[61]:
```
# Histogram plot for the column ElectronicsAndSemiconductor
sns.histplot(df_data['ElectronicsAndSemicon'])
```

[61]: <Axes: xlabel='ElectronicsAndSemicon', ylabel='Count'>



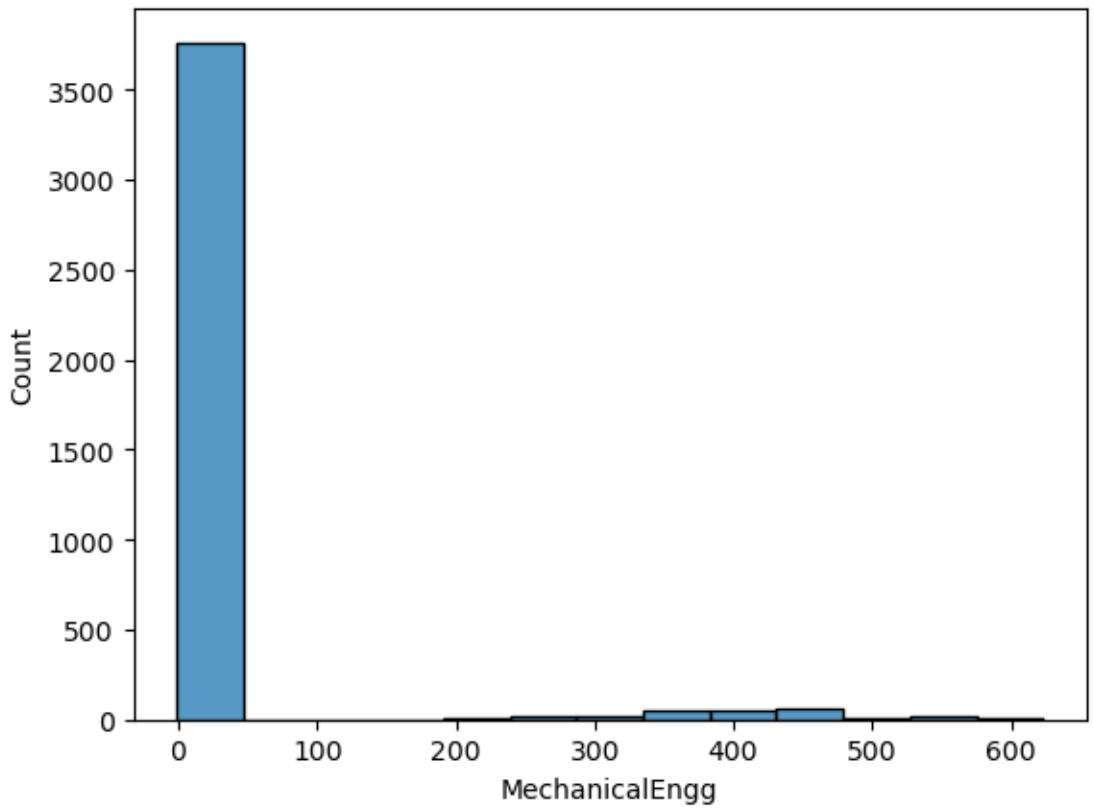
[62]:
```
# Histogram plot for the column ComputerScience
sns.histplot(df_data['ComputerScience'])
```

[62]: <Axes: xlabel='ComputerScience', ylabel='Count'>



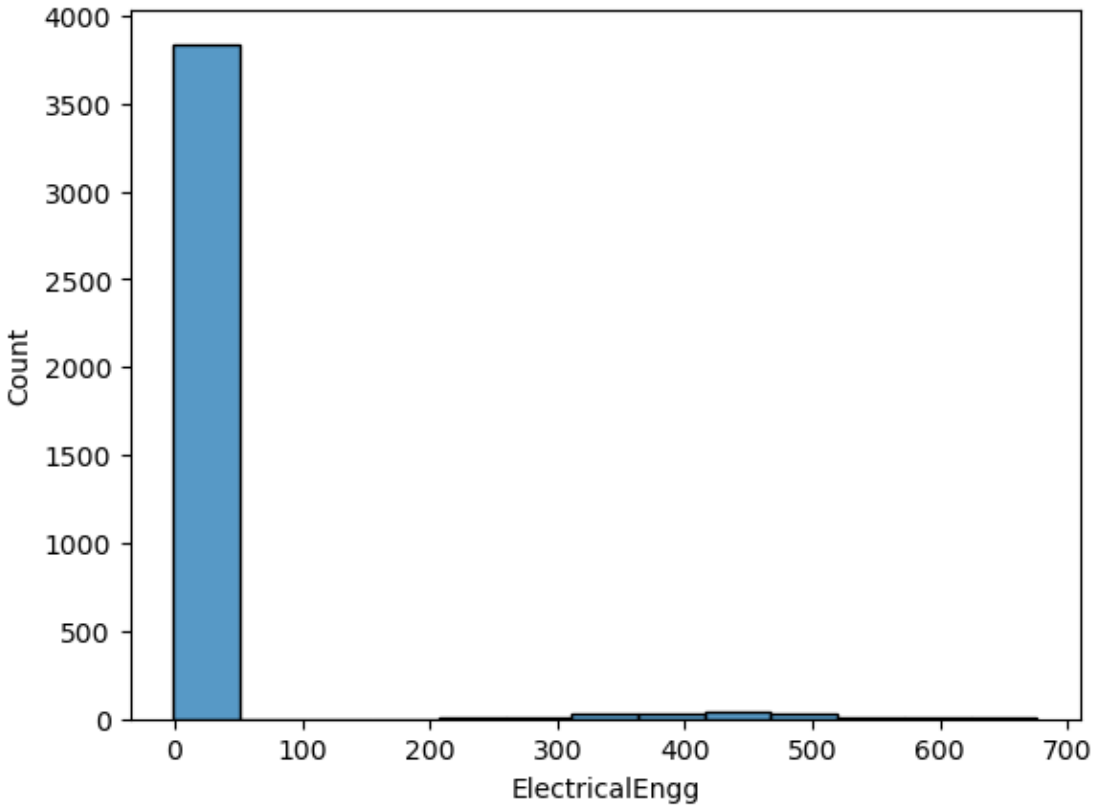
[63]:
```
# Histogram plot for the column Mechanical Engineering
sns.histplot(df_data['MechanicalEngg'])
```

[63]: <Axes: xlabel='MechanicalEngg', ylabel='Count'>



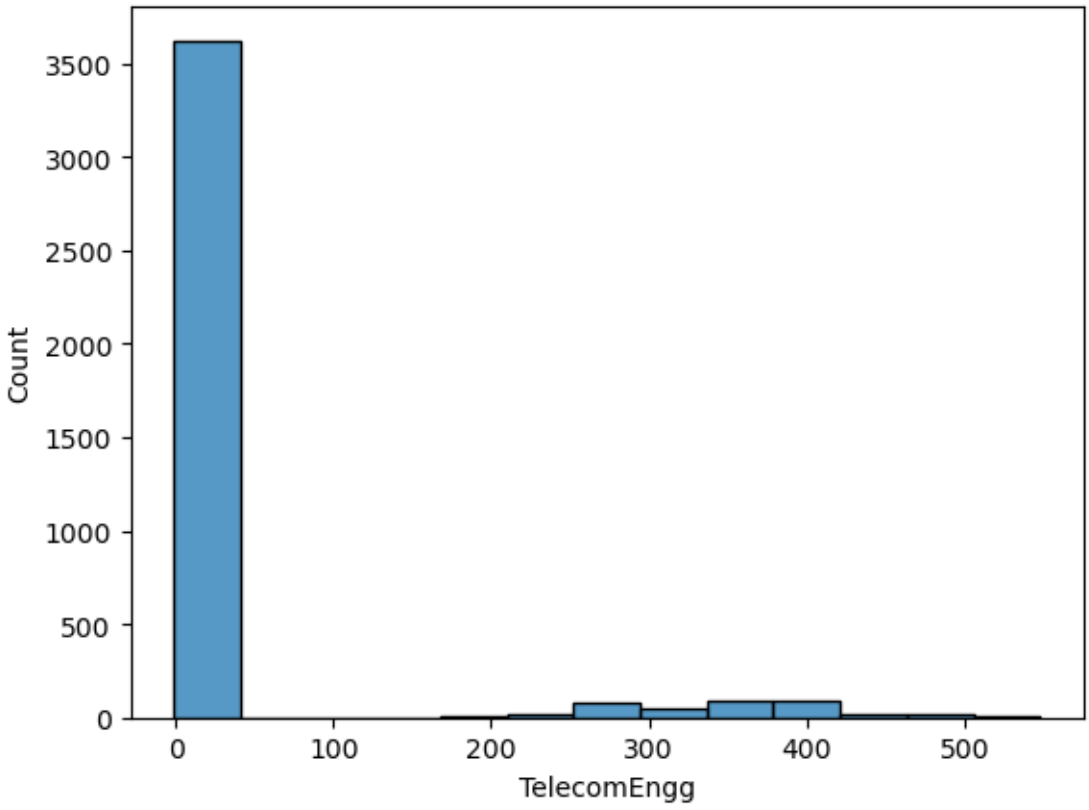
[64]:
```
# Histogram plot for the column Electrical Engineering
sns.histplot(df_data['ElectricalEngg'])
```

[64]: <Axes: xlabel='ElectricalEngg', ylabel='Count'>



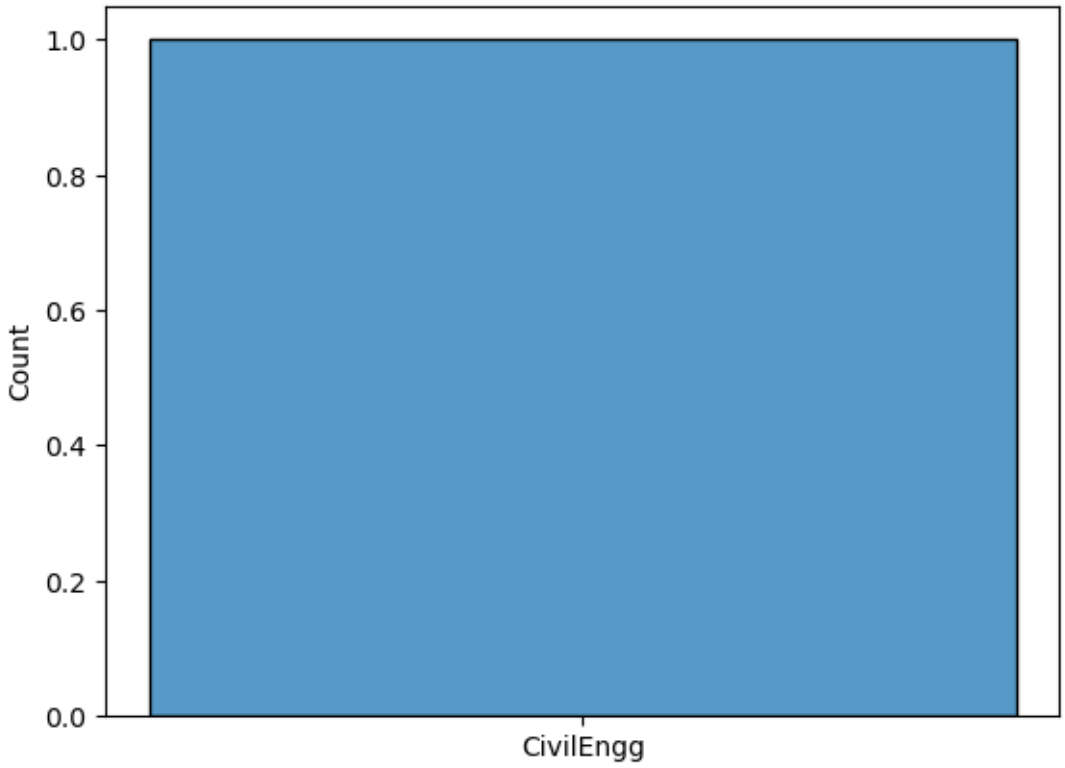
[65]:
```
# Histogram plot for the column Telecom Engineering
sns.histplot(df_data['TelecomEngg'])
```

[65]: <Axes: xlabel='TelecomEngg', ylabel='Count'>



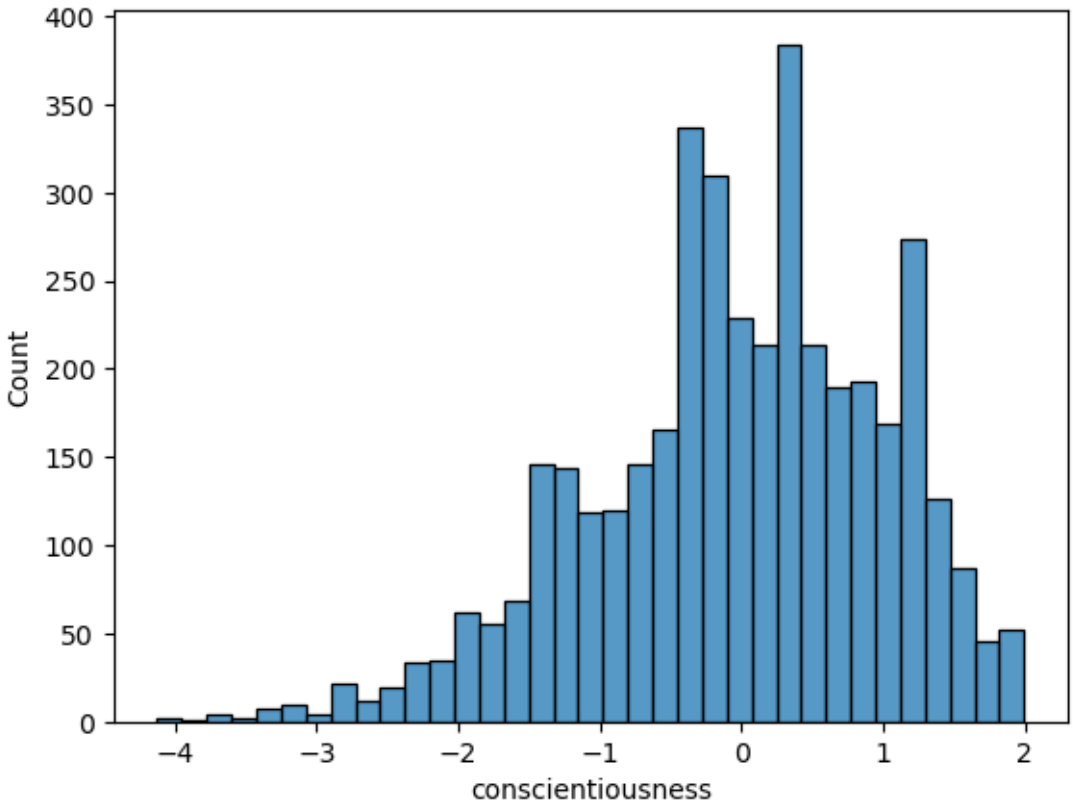
[66]:
```
# Histogram plot for the column CivilEngineering
sns.histplot('CivilEngg')
```

[66]: <Axes: ylabel='Count'>



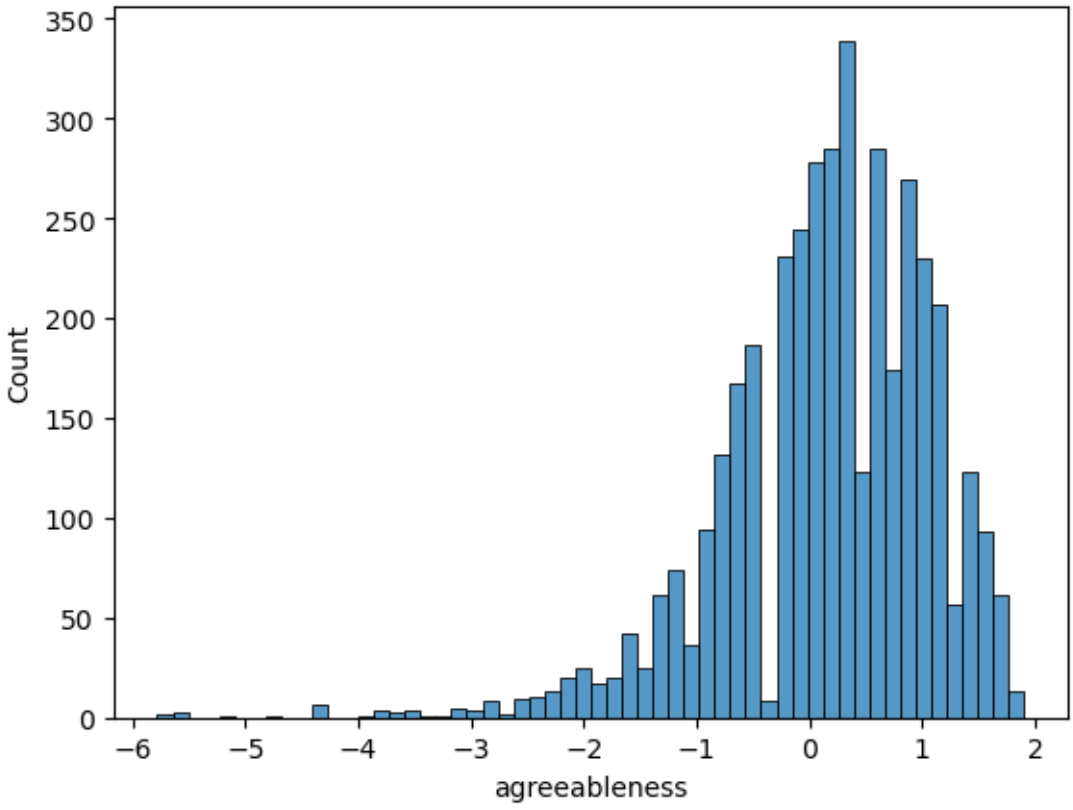
[67]:
```
# Histogram plot for the column conscientiousness
sns.histplot(df_data['conscientiousness'])
```

[67]:
```
<Axes: xlabel='conscientiousness', ylabel='Count'>
```



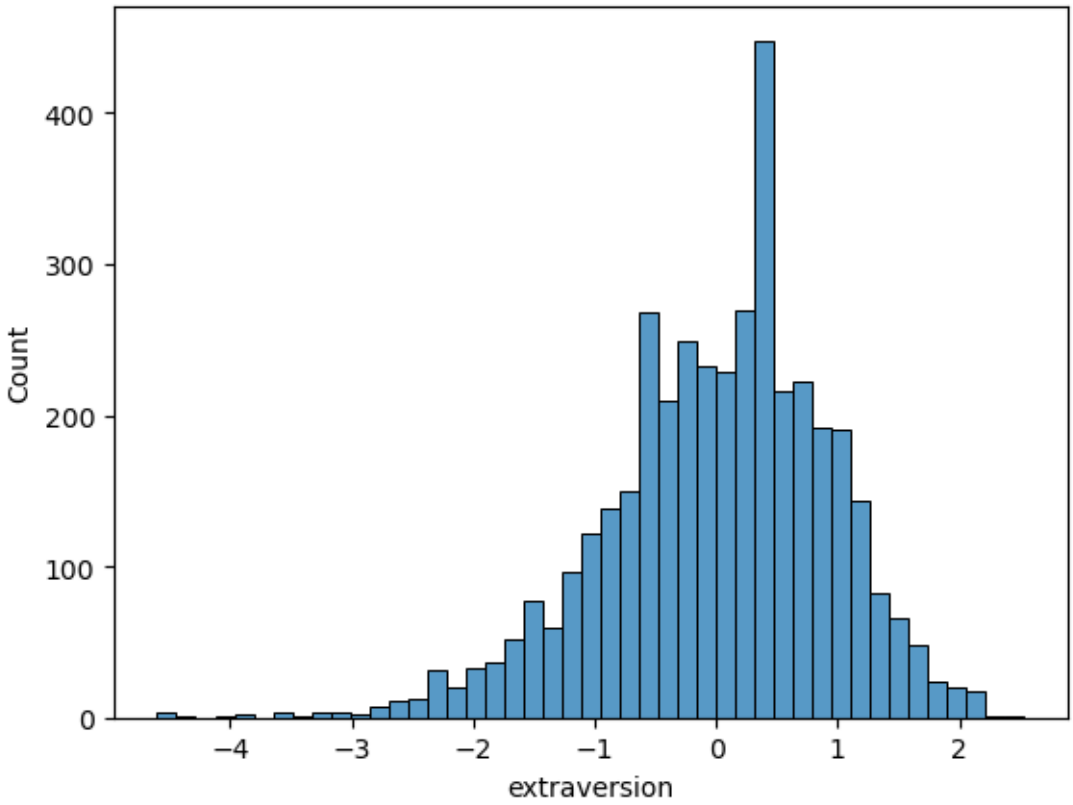
[68]:
```
# Histogram plot for the column agreeableness
sns.histplot(df_data['agreeableness'])
```

[68]: <Axes: xlabel='agreeableness', ylabel='Count'>



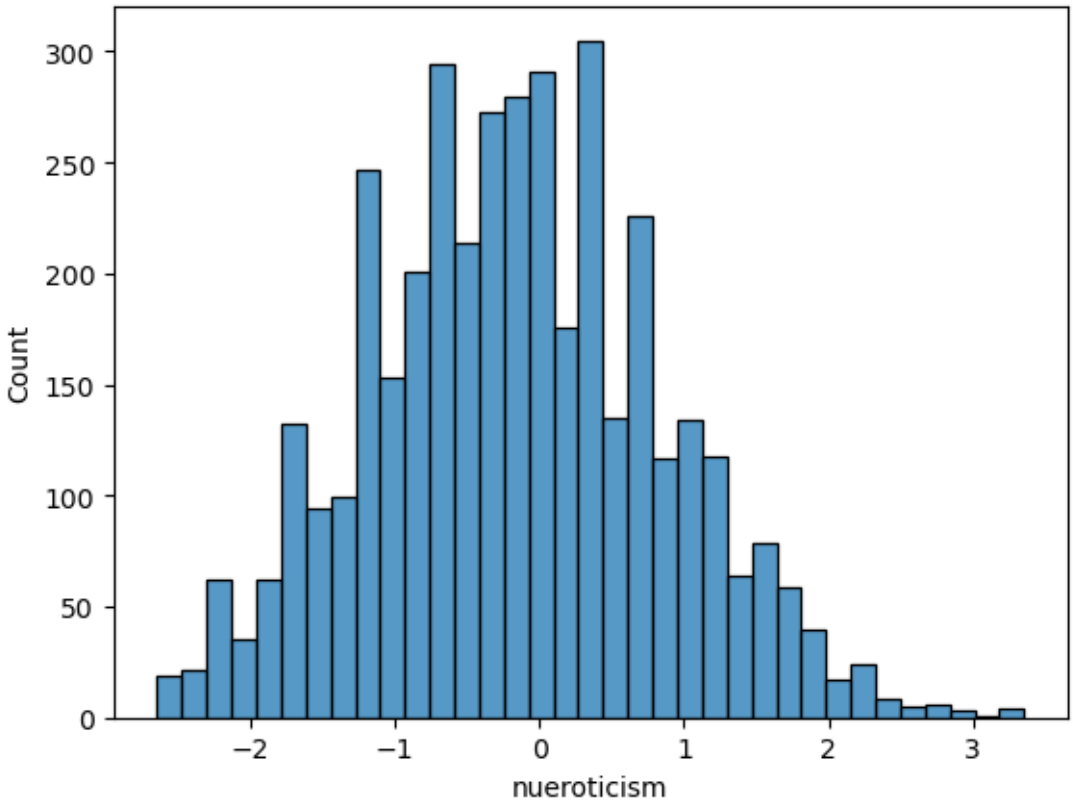
[69]:
```
# Histogram plot for the column extraversion
sns.histplot(df_data['extraversion'])
```

[69]: <Axes: xlabel='extraversion', ylabel='Count'>



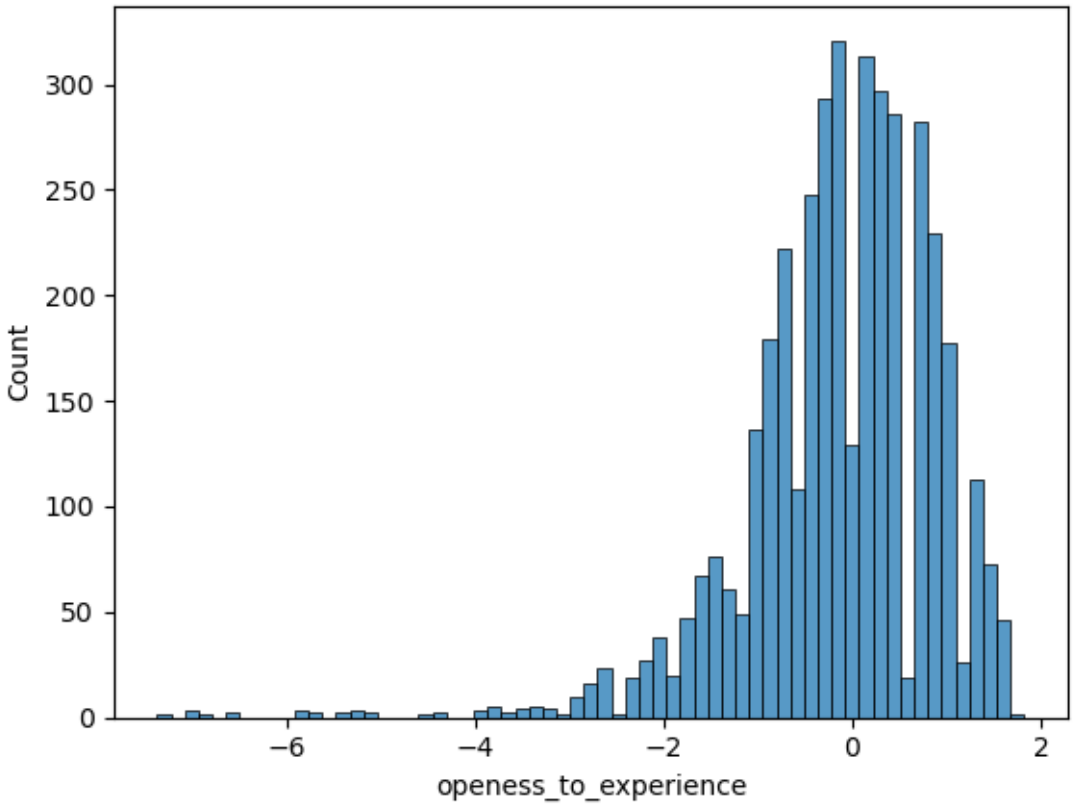
[70]:
```
# Histogram plot for the column nueroticism
sns.histplot(df_data['nueroticism'])
```

[70]: <Axes: xlabel='nueroticism', ylabel='Count'>



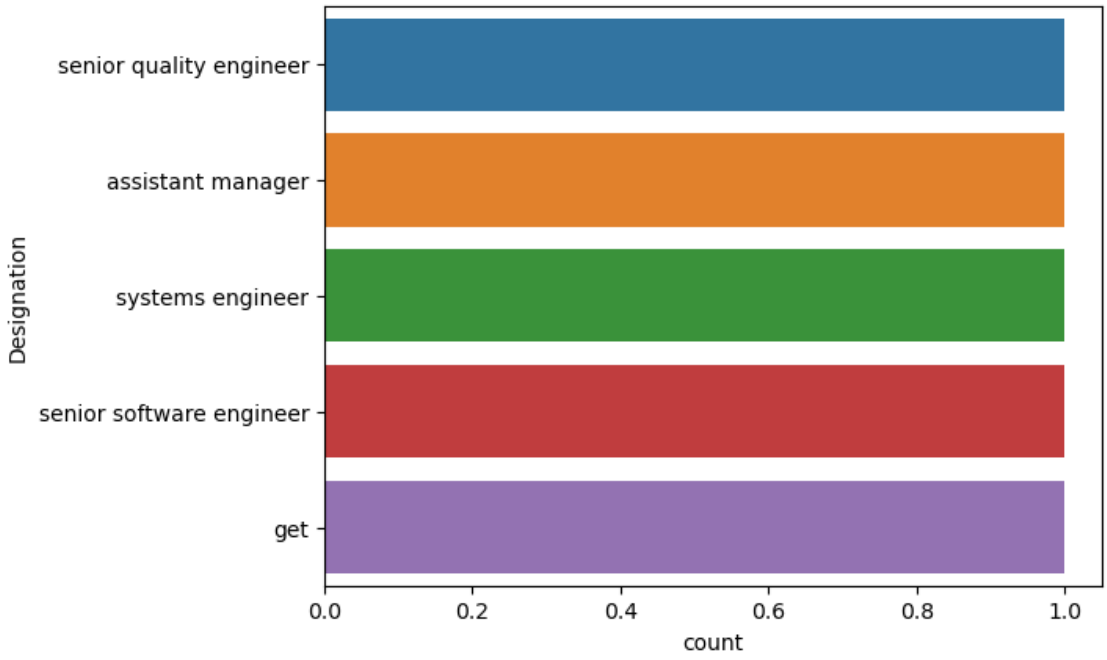
[71]:
```
# Histogram plot for the column openess_to_experience
sns.histplot(df_data['openess_to_experience'])
```

[71]: <Axes: xlabel='openess_to_experience', ylabel='Count'>



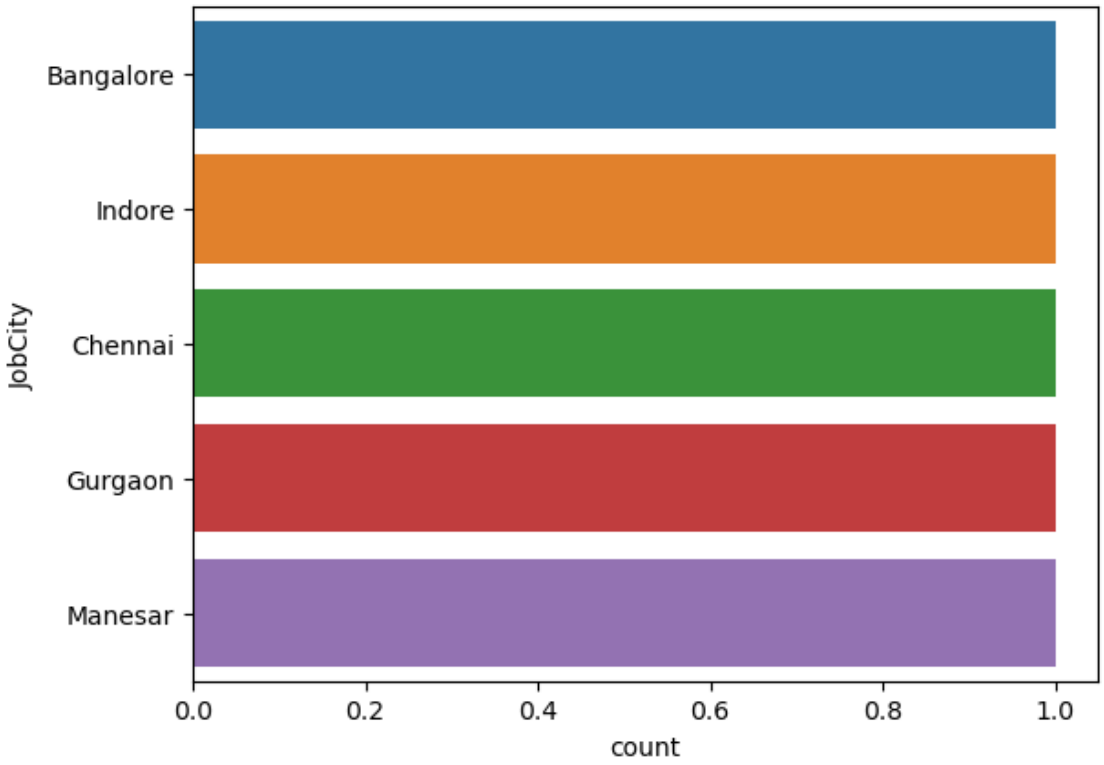
[72]:
```
# countplot for the column Designation
sns.countplot(y=df_data['Designation'].head())
```

[72]: <Axes: xlabel='count', ylabel='Designation'>



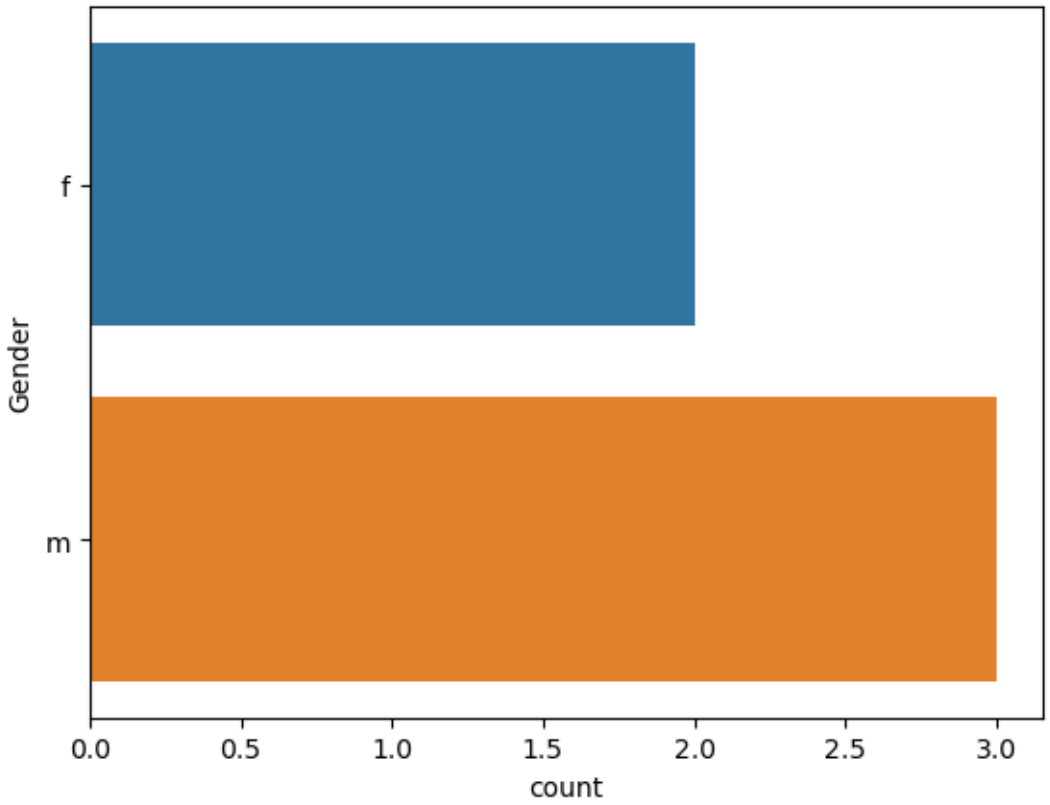
[73]:
```
# countplot for the column JobCity
sns.countplot(y=df_data['JobCity'].head())
```

[73]: <Axes: xlabel='count', ylabel='JobCity'>



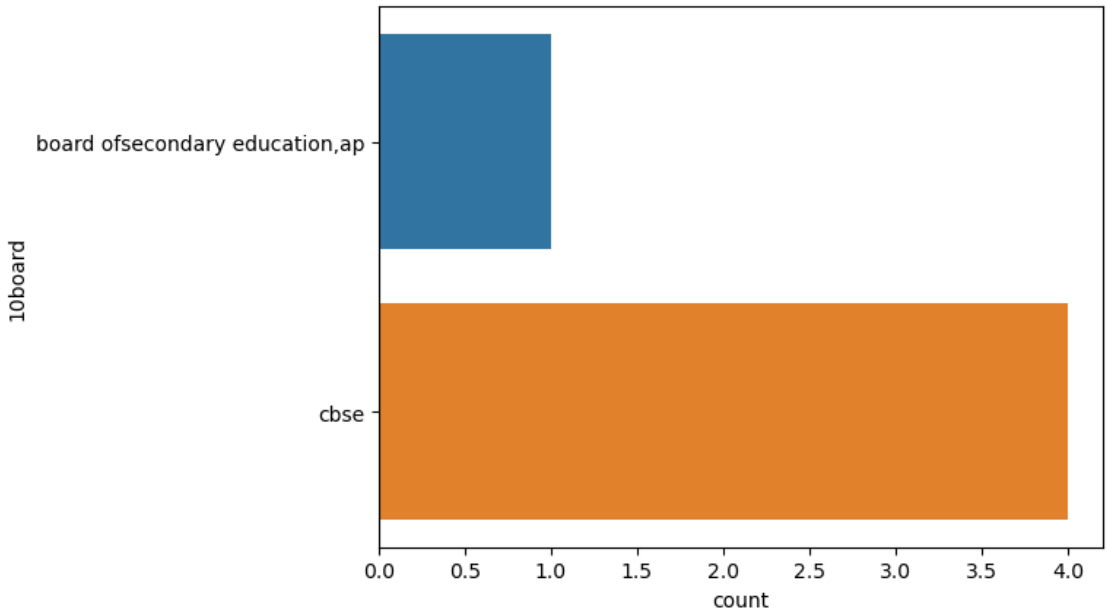
[74]:
```
# countplot for the column Gender
sns.countplot(y=df_data['Gender'].head())
```

[74]: <Axes: xlabel='count', ylabel='Gender'>



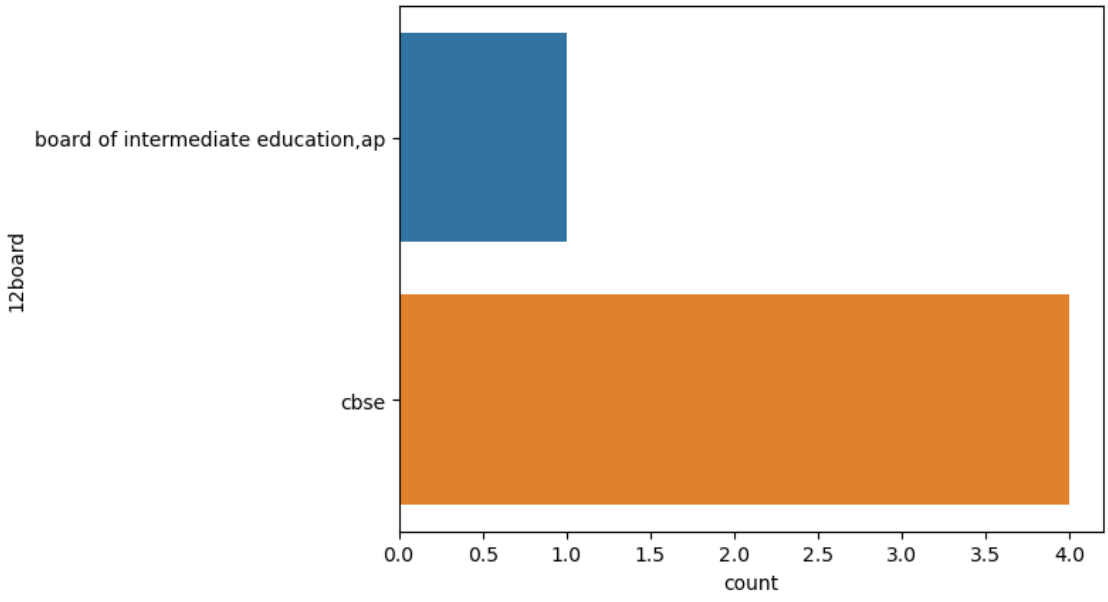
[75]:
```
# countplot for the column 10board
sns.countplot(y=df_data['10board'].head())
```

[75]: <Axes: xlabel='count', ylabel='10board'>



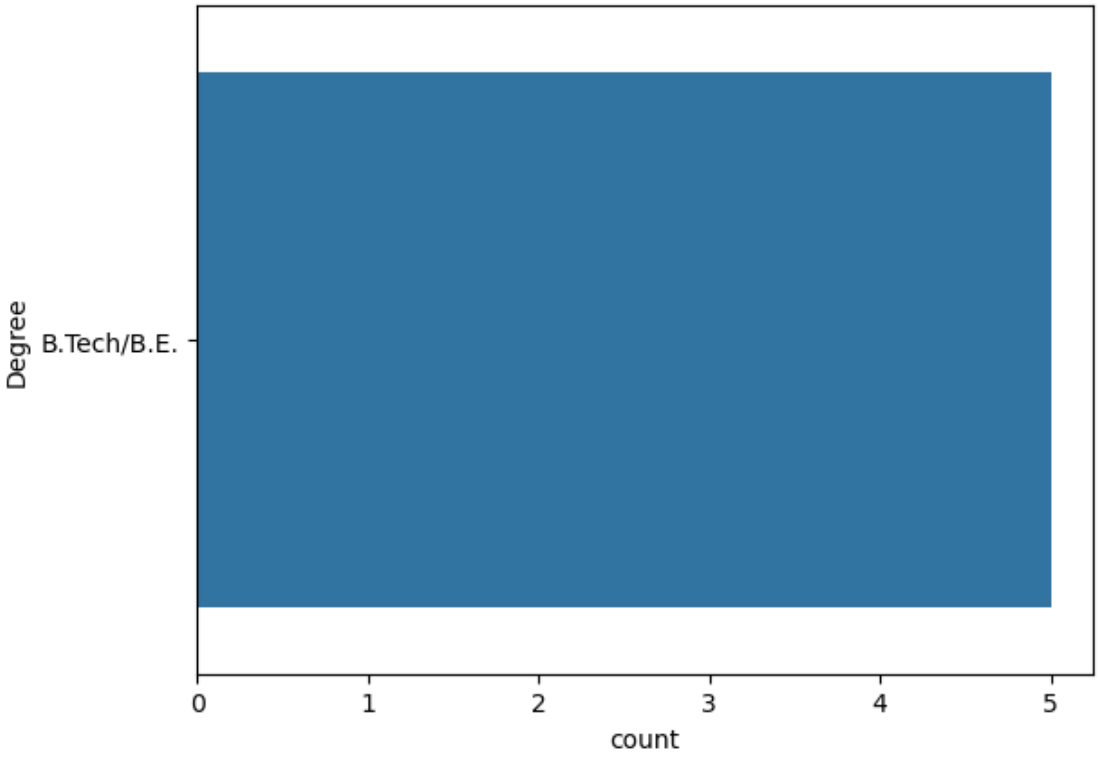
[76]:
```
# countplot for the column 12board
sns.countplot(y=df_data['12board'].head())
```

[76]: <Axes: xlabel='count', ylabel='12board'>



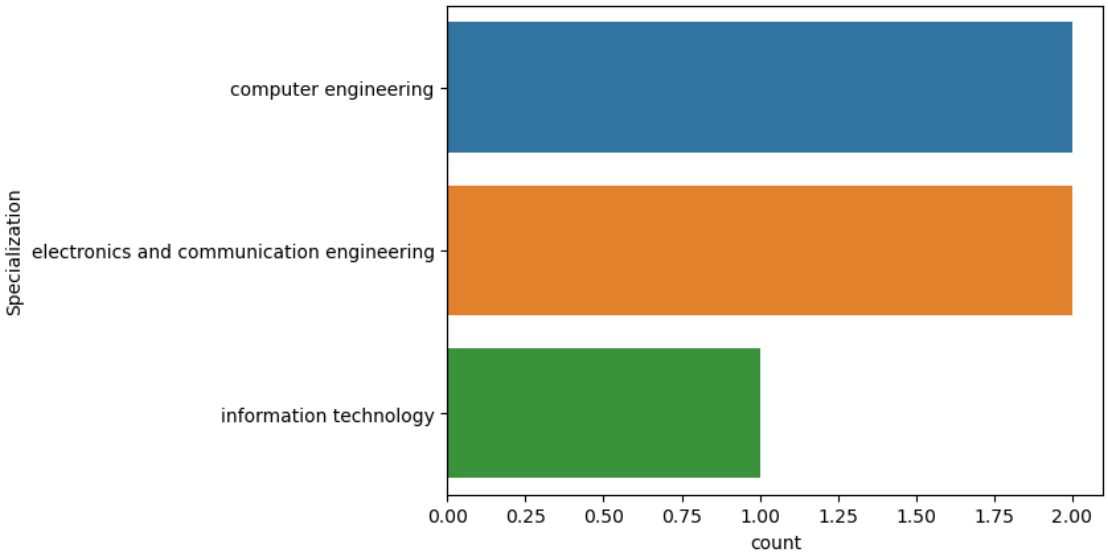
[77]:
```
# countplot for the column Degree
sns.countplot(y=df_data['Degree'].head())
```

[77]: <Axes: xlabel='count', ylabel='Degree'>



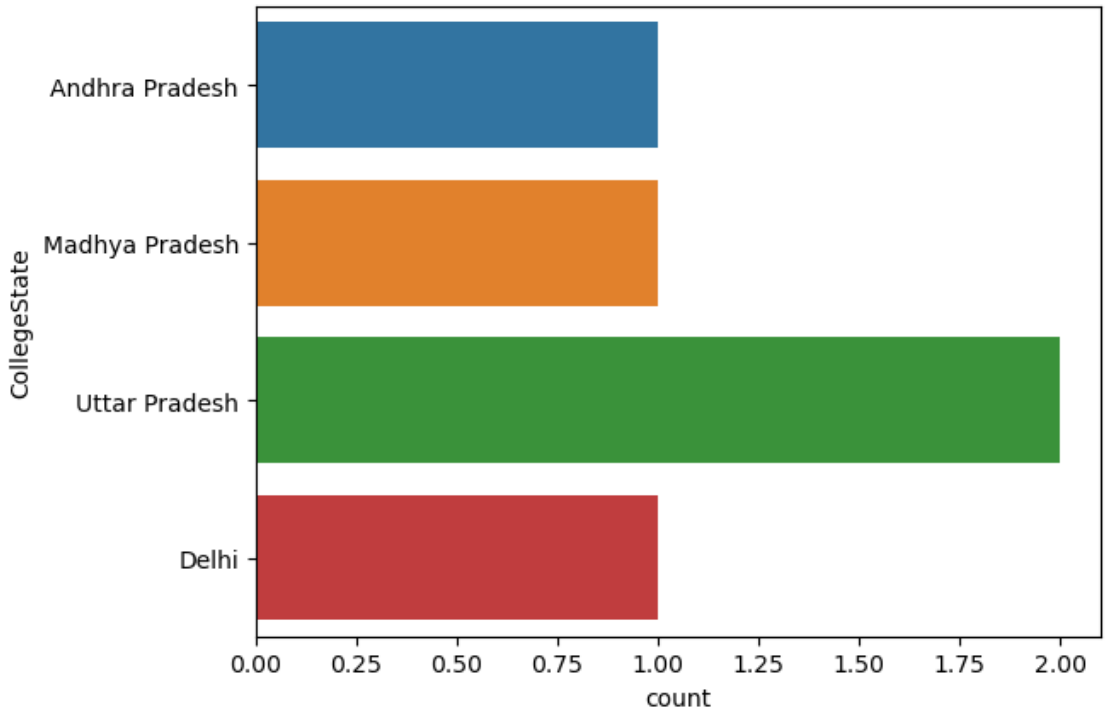
[78]:
```
# countplot for the column Specialization
sns.countplot(y=df_data['Specialization'].head())
```

[78]: <Axes: xlabel='count', ylabel='Specialization'>



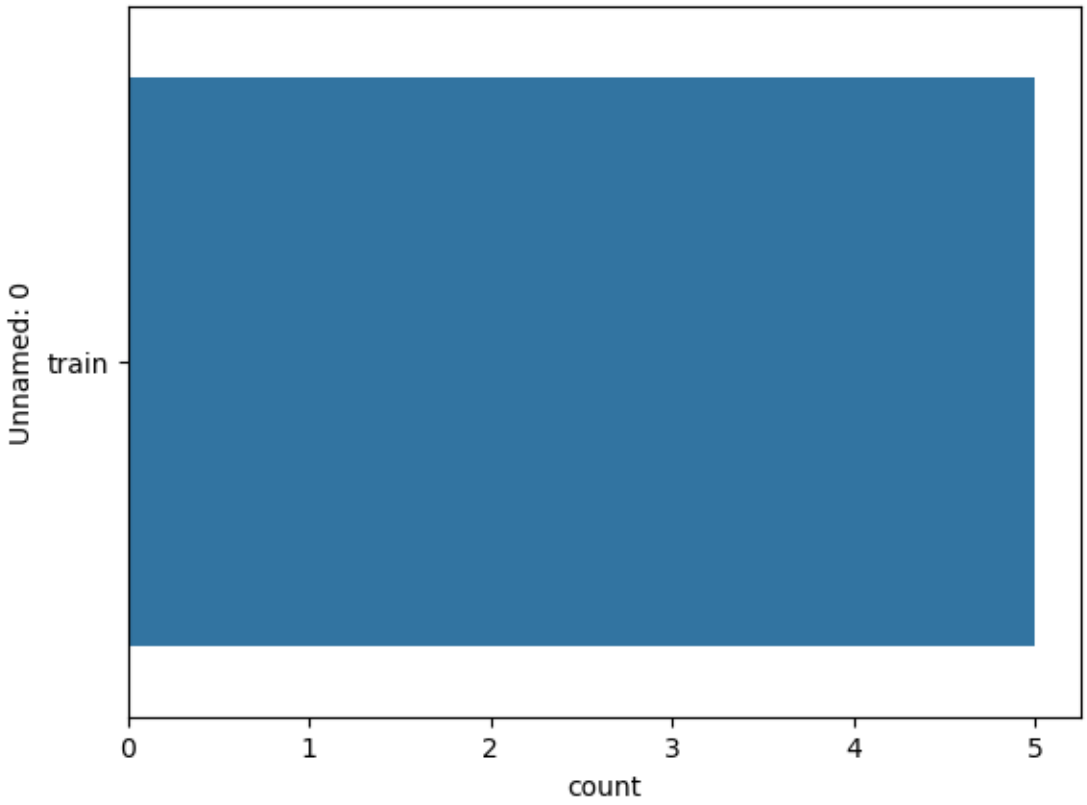
[79]:
```
# countplot for the column CollegeState
sns.countplot(y=df_data['CollegeState'].head())
```

[79]: <Axes: xlabel='count', ylabel='CollegeState'>



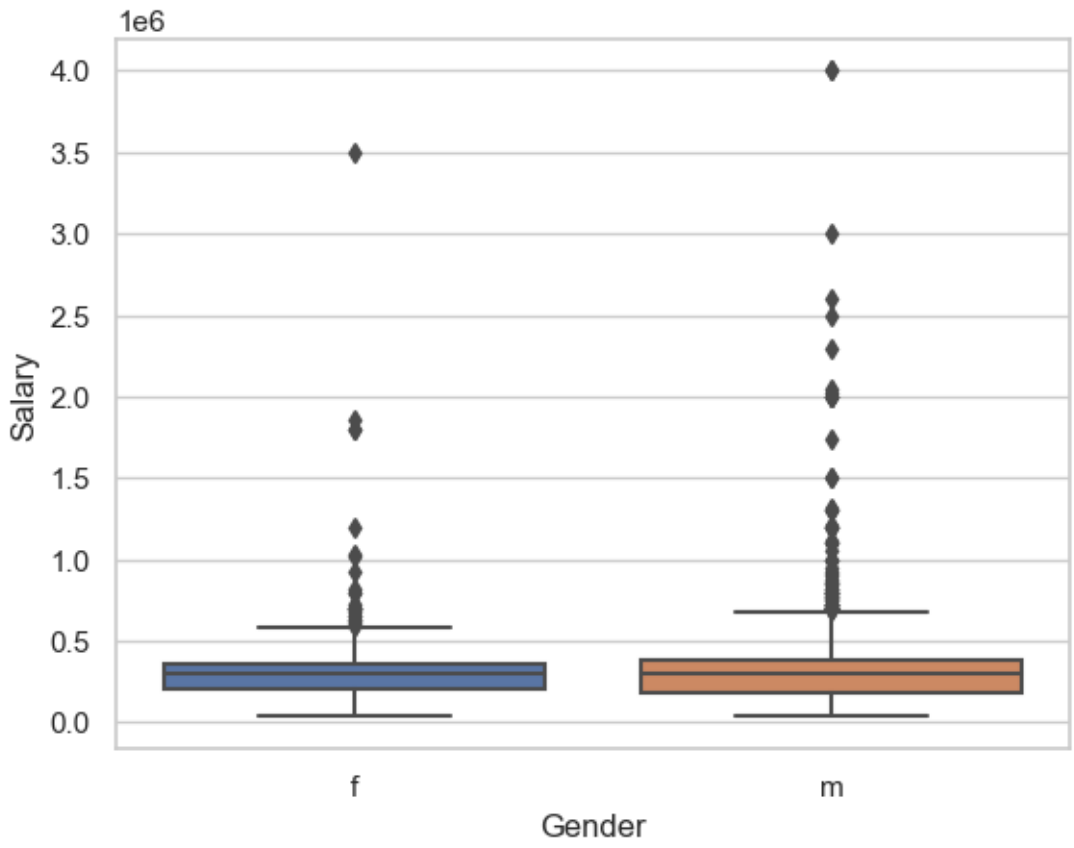
[80]:
```
# countplot for the column Unnamed:  0
sns.countplot(y=df_data['Unnamed: 0'].head())
```

[80]: <Axes: xlabel='count', ylabel='Unnamed: 0'>



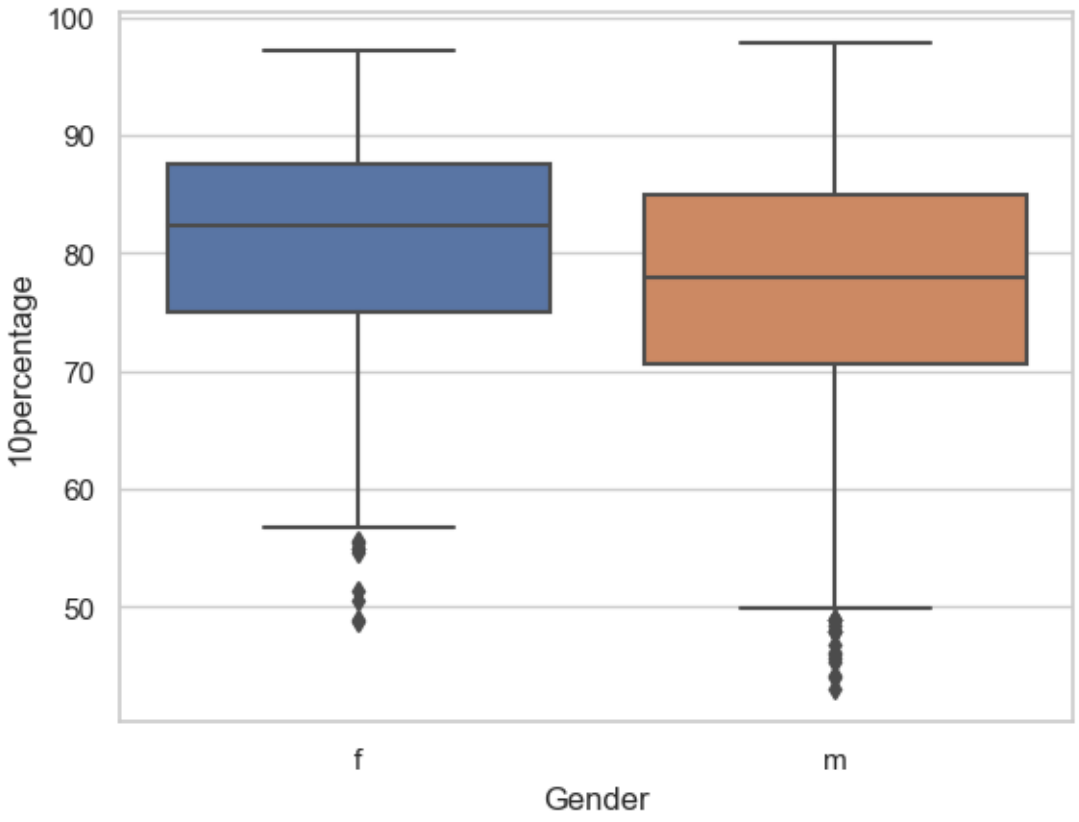
[81]:
```
# Boxplot for Gender Vs Salary
sns.set(style='whitegrid')
sns.boxplot(x=df_data["Gender"],y=df_data["Salary"])
```

[81]: <Axes: xlabel='Gender', ylabel='Salary'>



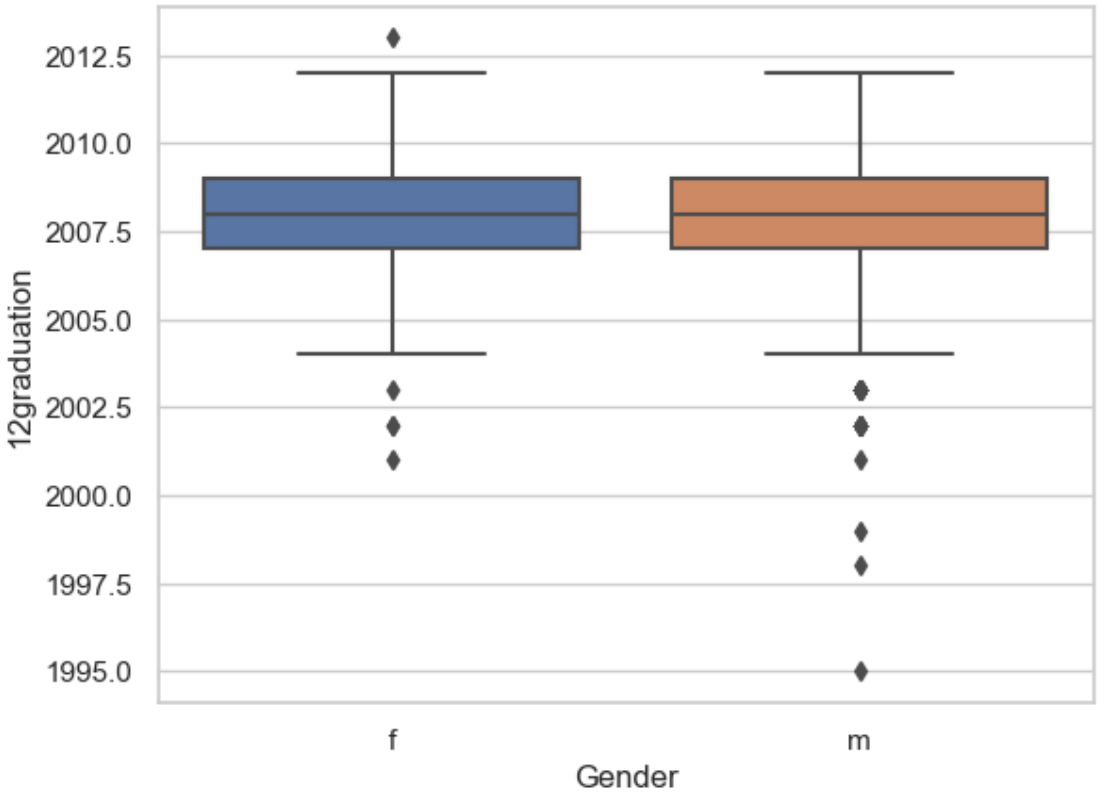
[82]:
```
# Boxplot for Gender Vs 10percentage
sns.set(style='whitegrid')
sns.boxplot(x=df_data["Gender"],y=df_data["10percentage"])
```

[82]: <Axes: xlabel='Gender', ylabel='10percentage'>



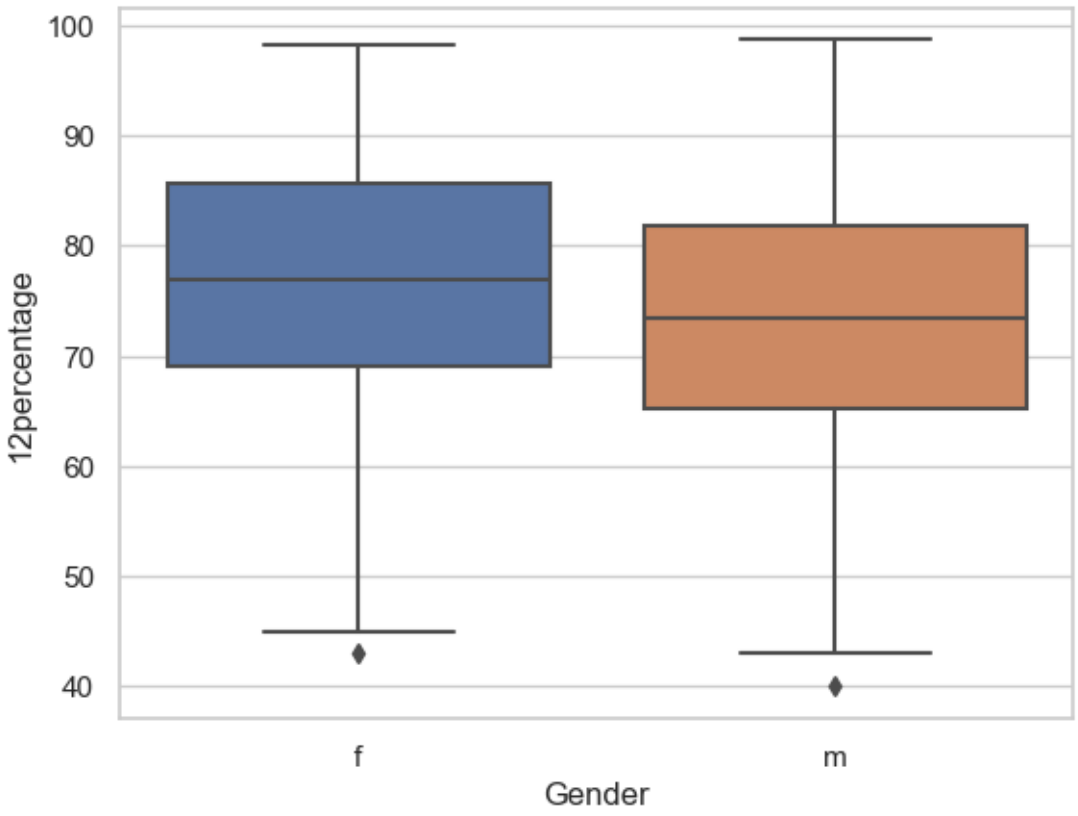
[83]:
```
# Boxplot for Gender Vs 12graduation
sns.set(style='whitegrid')
sns.boxplot(x=df_data["Gender"],y=df_data["12graduation"])
```

[83]: <Axes: xlabel='Gender', ylabel='12graduation'>



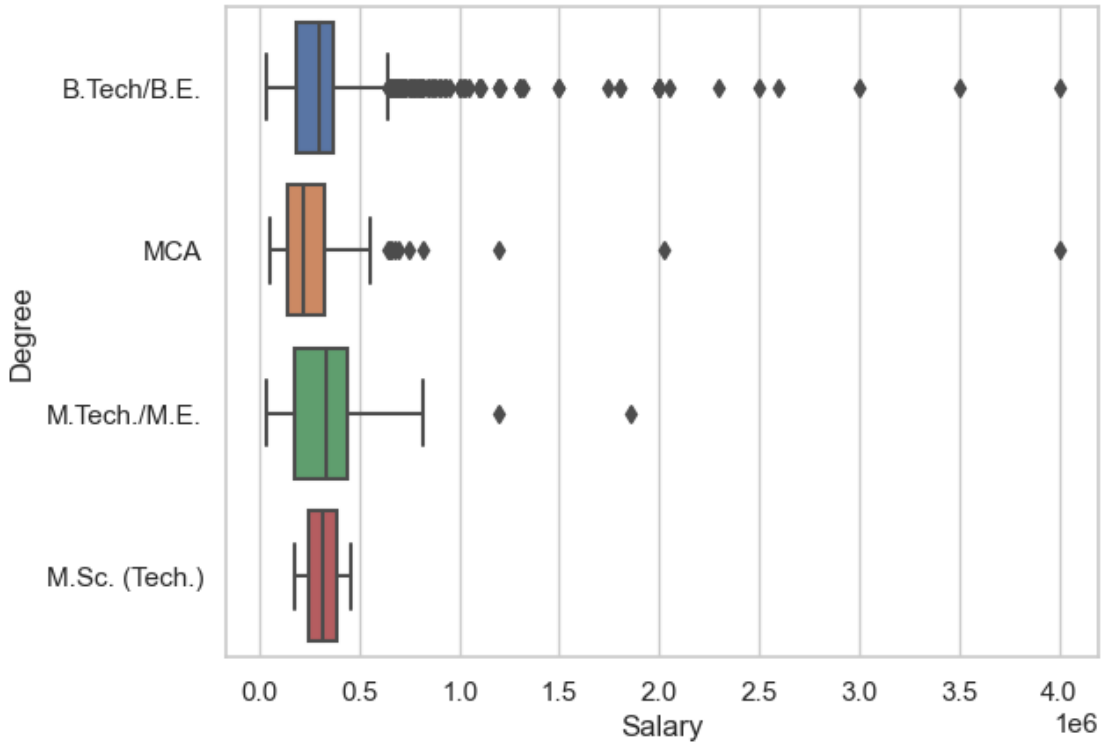
[84]:
```
# Boxplot for Gender Vs 12percentage
sns.set(style='whitegrid')
sns.boxplot(x=df_data["Gender"],y=df_data["12percentage"])
```

[84]: <Axes: xlabel='Gender', ylabel='12percentage'>



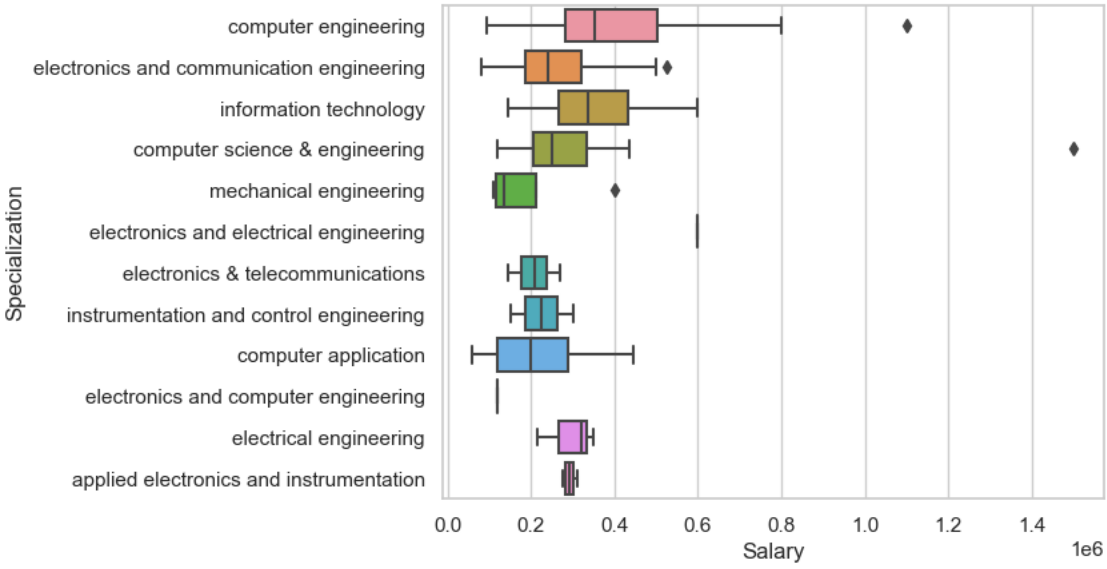
[85]:
```
# Boxplot for Degree Vs Salary
sns.set(style='whitegrid')
sns.boxplot(y=df_data["Degree"],x=df_data["Salary"])
```

[85]: <Axes: xlabel='Salary', ylabel='Degree'>



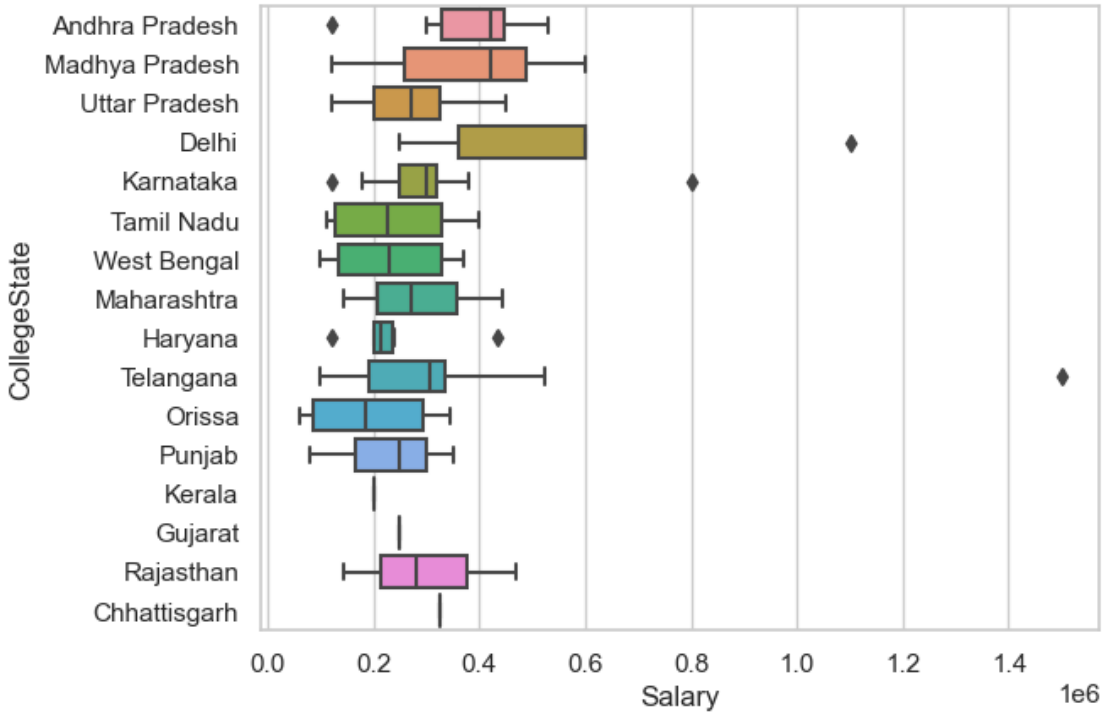
[86]:
```
# Boxplot for Specialization Vs Salary
sns.set(style='whitegrid')
sns.boxplot(y=df_data["Specialization"].head(100),x=df_data["Salary"].head(100))
```

[86]: <Axes: xlabel='Salary', ylabel='Specialization'>



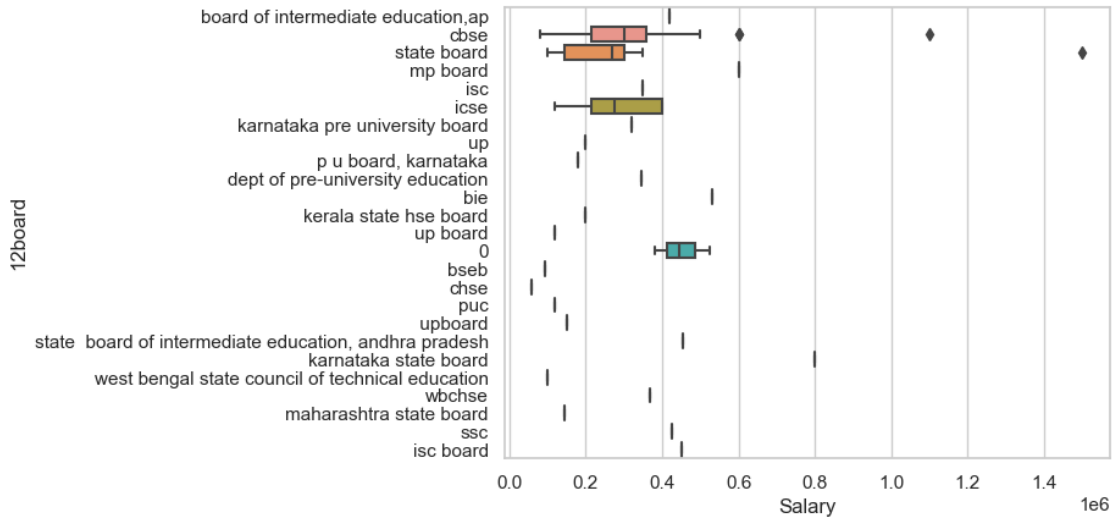
[87]:
```
# Boxplot for CollegeState Vs Salary
sns.set(style='whitegrid')
sns.boxplot(y=df_data["CollegeState"].head(100),x=df_data["Salary"].head(100))
```

[87]: <Axes: xlabel='Salary', ylabel='CollegeState'>



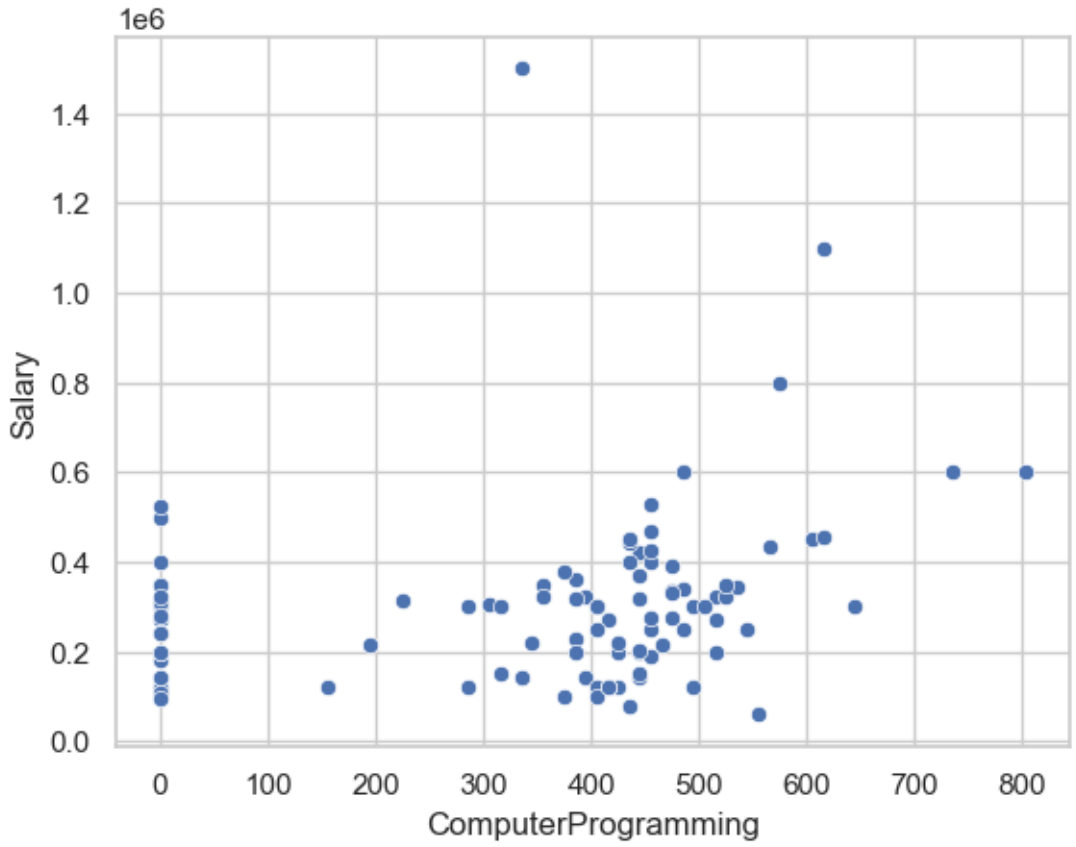
[88]:
```
# Boxplot for 12board Vs Salary
sns.set(style='whitegrid')
sns.boxplot(y=df_data["12board"].head(100),x=df_data["Salary"].head(100))
```

[88]: <Axes: xlabel='Salary', ylabel='12board'>



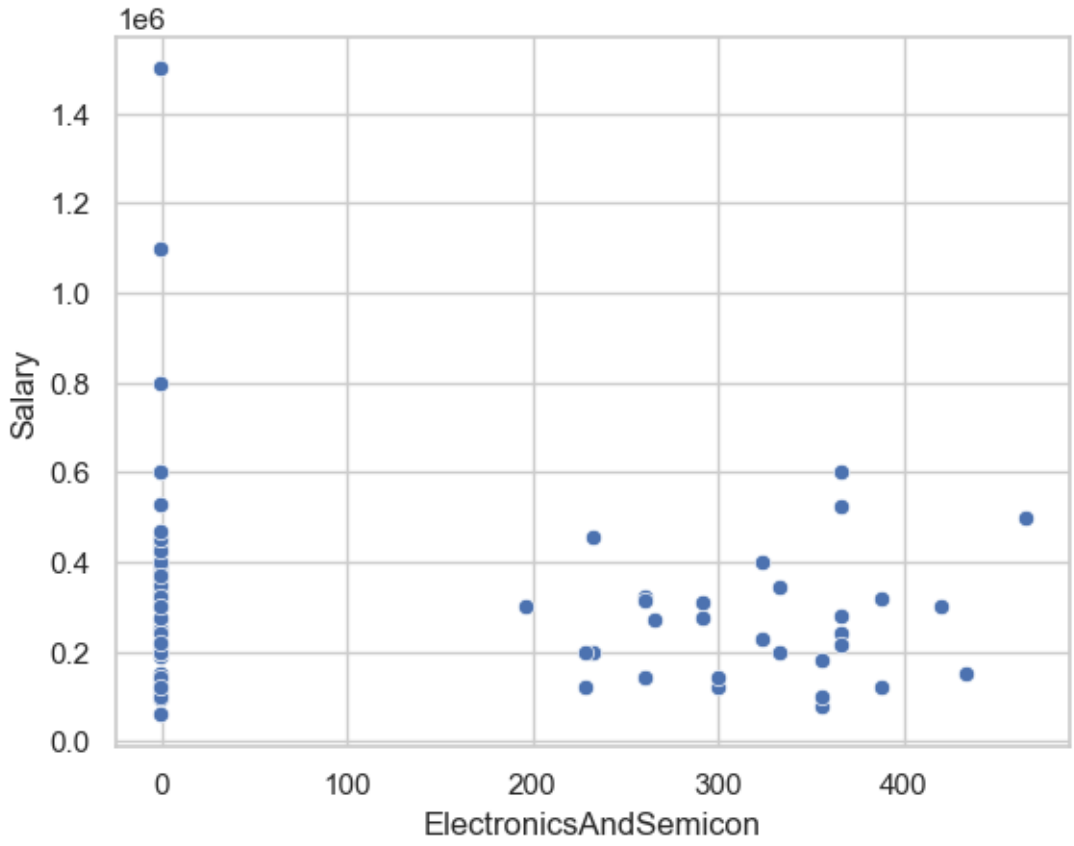
[89]:
```
# Scatterplot for ComputerProgramming Vs Salary
sns.scatterplot(x=df_data['ComputerProgramming'].head(100),y=df_data['Salary'].
↪head(100))
```

[89]: <Axes: xlabel='ComputerProgramming', ylabel='Salary'>



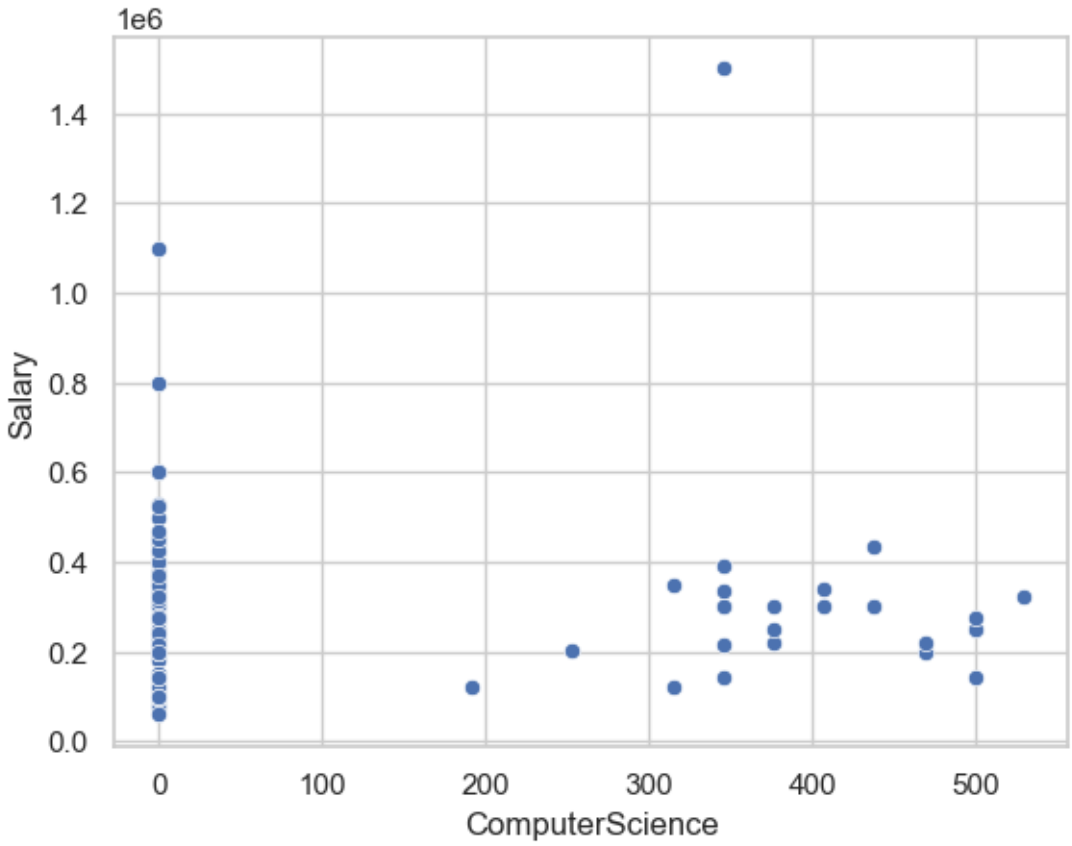
[90]:
```
# scatterplot for ElectronicsAndSemicon Vs Salary
sns.scatterplot(x=df_data['ElectronicsAndSemicon'].
 ↪head(100),y=df_data['Salary'].head(100))
```

[90]: <Axes: xlabel='ElectronicsAndSemicon', ylabel='Salary'>



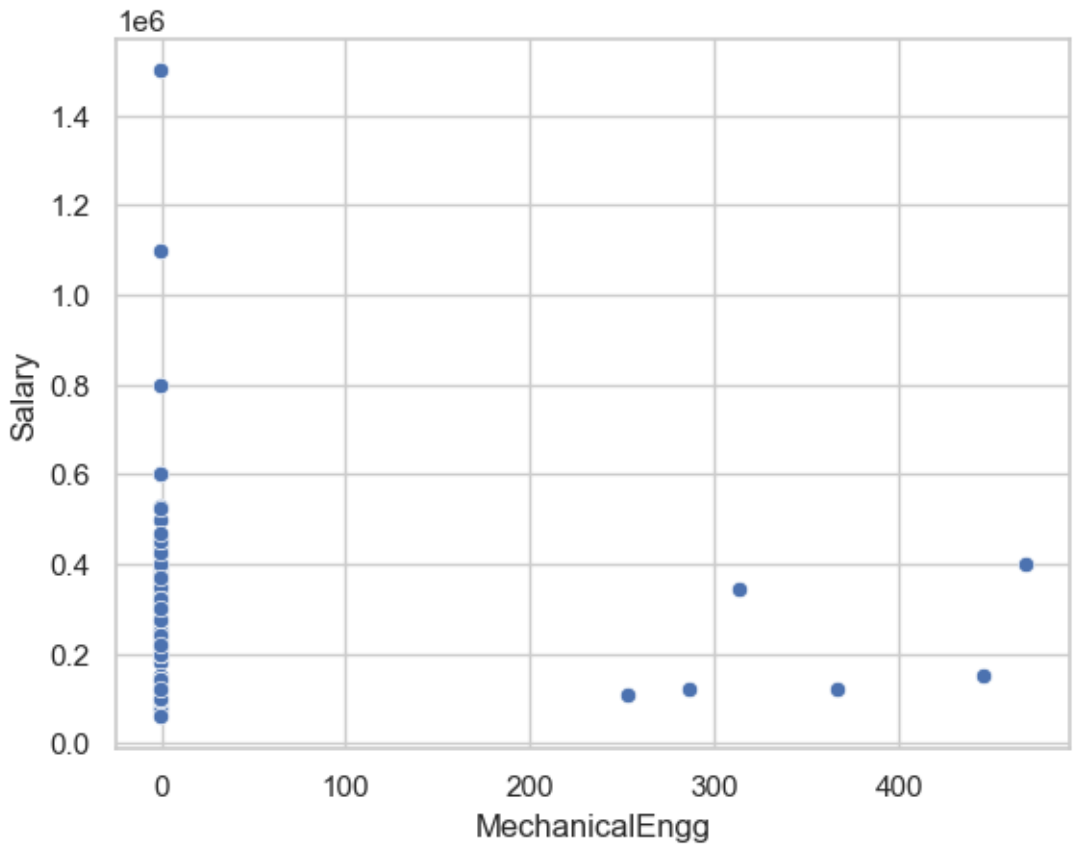
[91]:
```
# scatter plot for ComputerScience Vs Salary
sns.scatterplot(x=df_data['ComputerScience'].head(100),y=df_data['Salary'].
 ↪head(100))
```

[91]: <Axes: xlabel='ComputerScience', ylabel='Salary'>



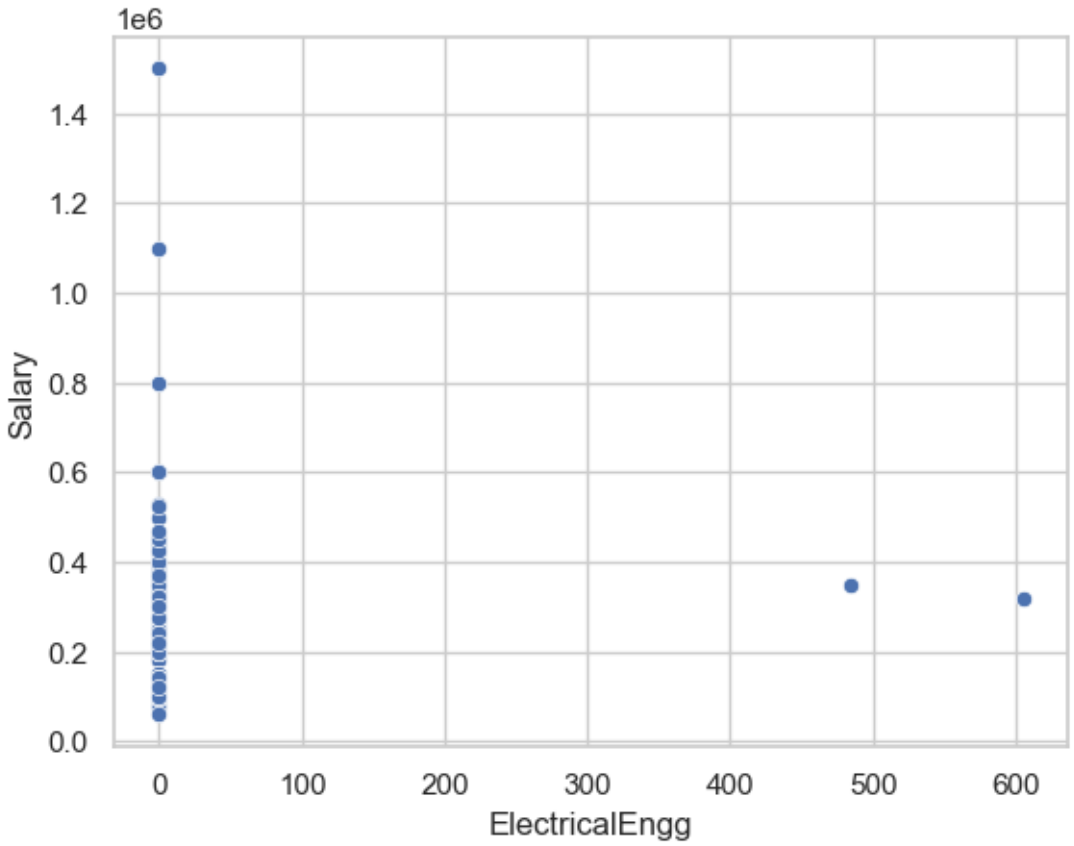
[92]:
```
# scatter plot for Mechanical Engineering Vs Salary
sns.scatterplot(x=df_data['MechanicalEngg'].head(100),y=df_data['Salary'].
↪head(100))
```

[92]: <Axes: xlabel='MechanicalEngg', ylabel='Salary'>



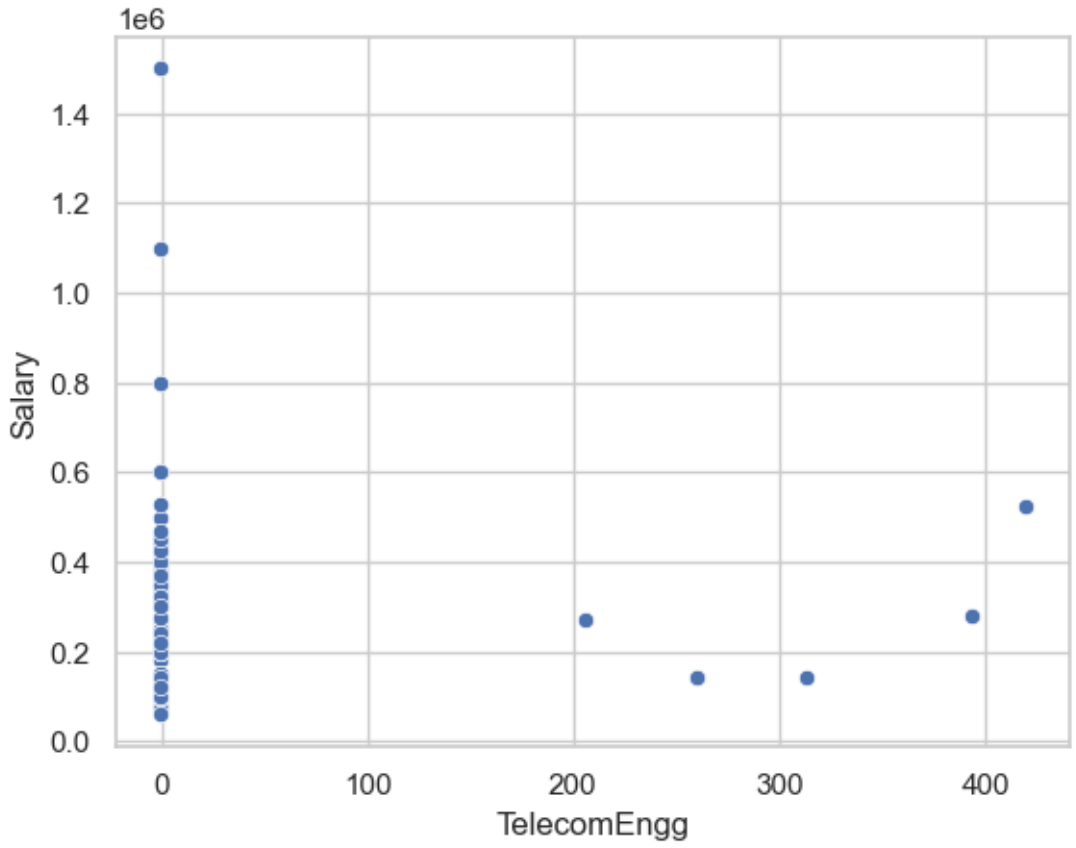
[93]:
```
# scatter plot for Electrical Engineering Vs Salary
sns.scatterplot(x=df_data['ElectricalEngg'].head(100),y=df_data['Salary'].
↪head(100))
```

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



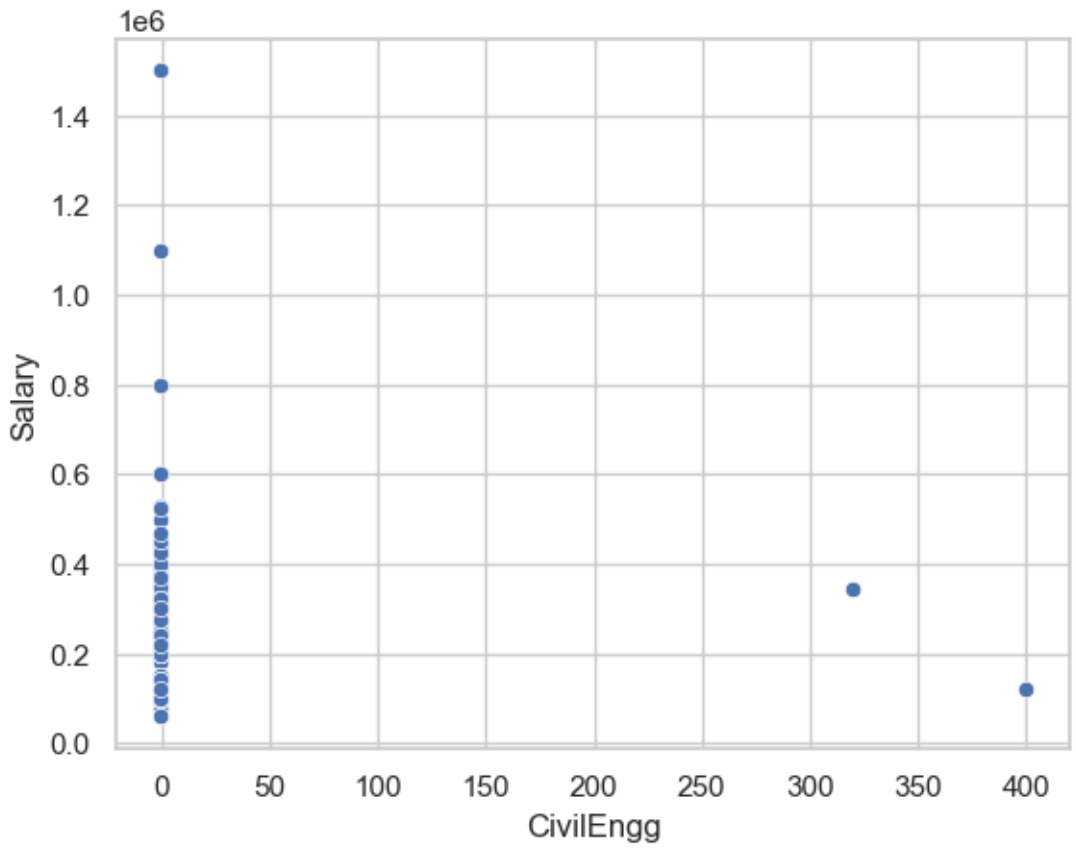
[94]:
```
# scatter plot for Telecom Engineering Vs Salary
sns.scatterplot(x=df_data['TelecomEngg'].head(100),y=df_data['Salary'].
↪head(100))
```

[94]:
```
<Axes: xlabel='TelecomEngg', ylabel='Salary'>
```



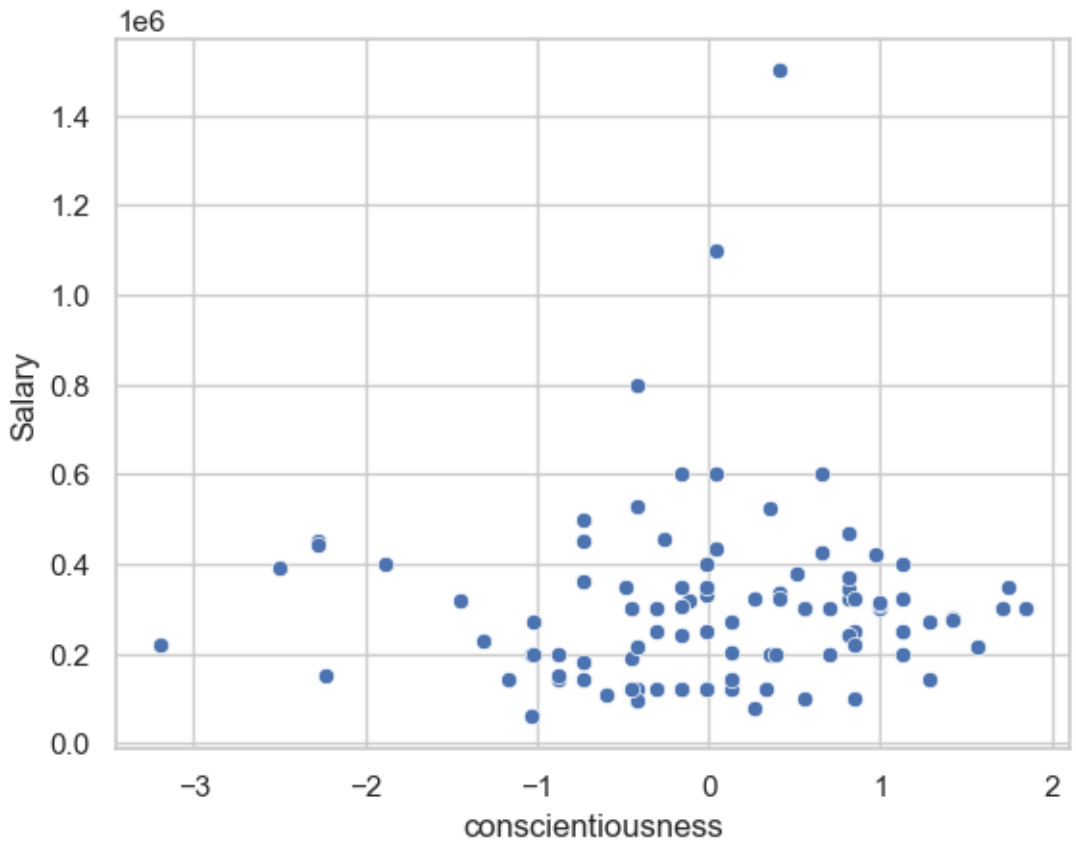
[95]:
```
# scatter plot for Civil Engineering Vs Salary
sns.scatterplot(x=df_data['CivilEngg'].head(100),y=df_data['Salary'].head(100))
```

[95]: <Axes: xlabel='CivilEngg', ylabel='Salary'>



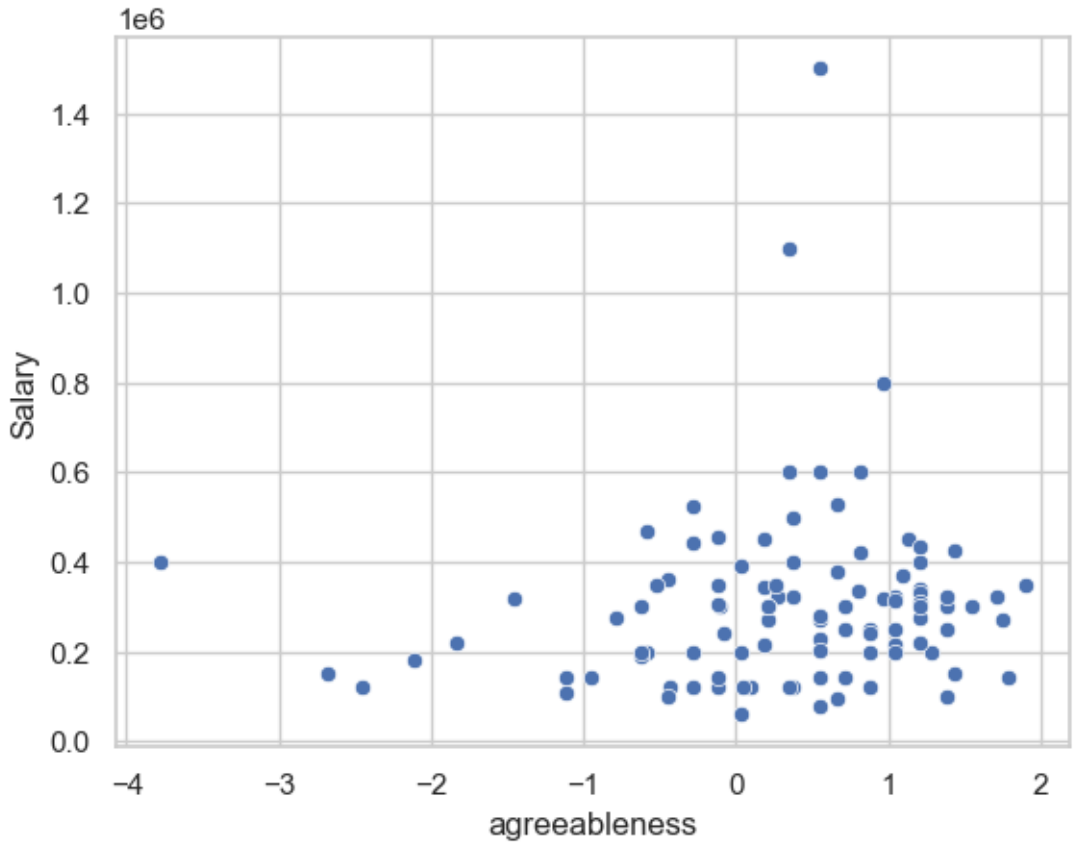
[96]:
```
# scatter plot for conscientiousness Vs Salary
sns.scatterplot(x=df_data['conscientiousness'].head(100),y=df_data['Salary'].
↪head(100))
```

[96]: <Axes: xlabel='conscientiousness', ylabel='Salary'>



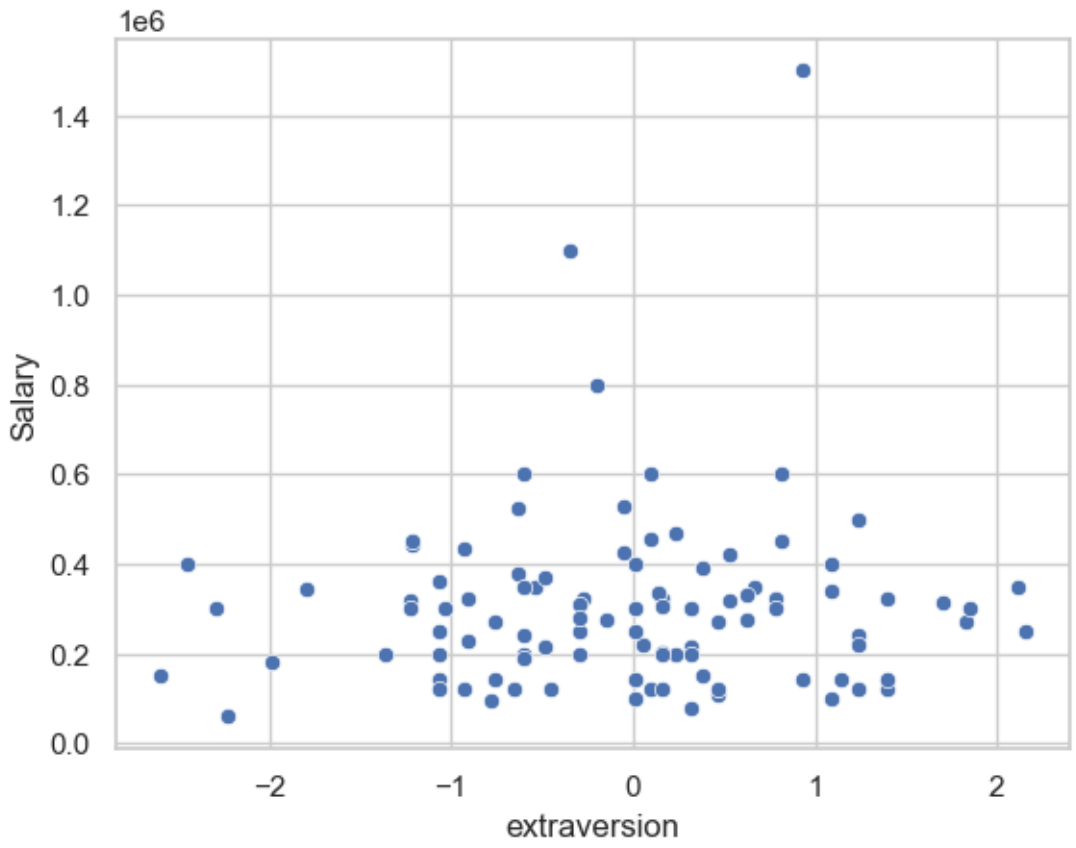
[97]:
```
# scatter plot for agreeableness Vs Salary
sns.scatterplot(x=df_data['agreeableness'].head(100),y=df_data['Salary'].
↪head(100))
```

[97]:
```
<Axes: xlabel='agreeableness', ylabel='Salary'>
```



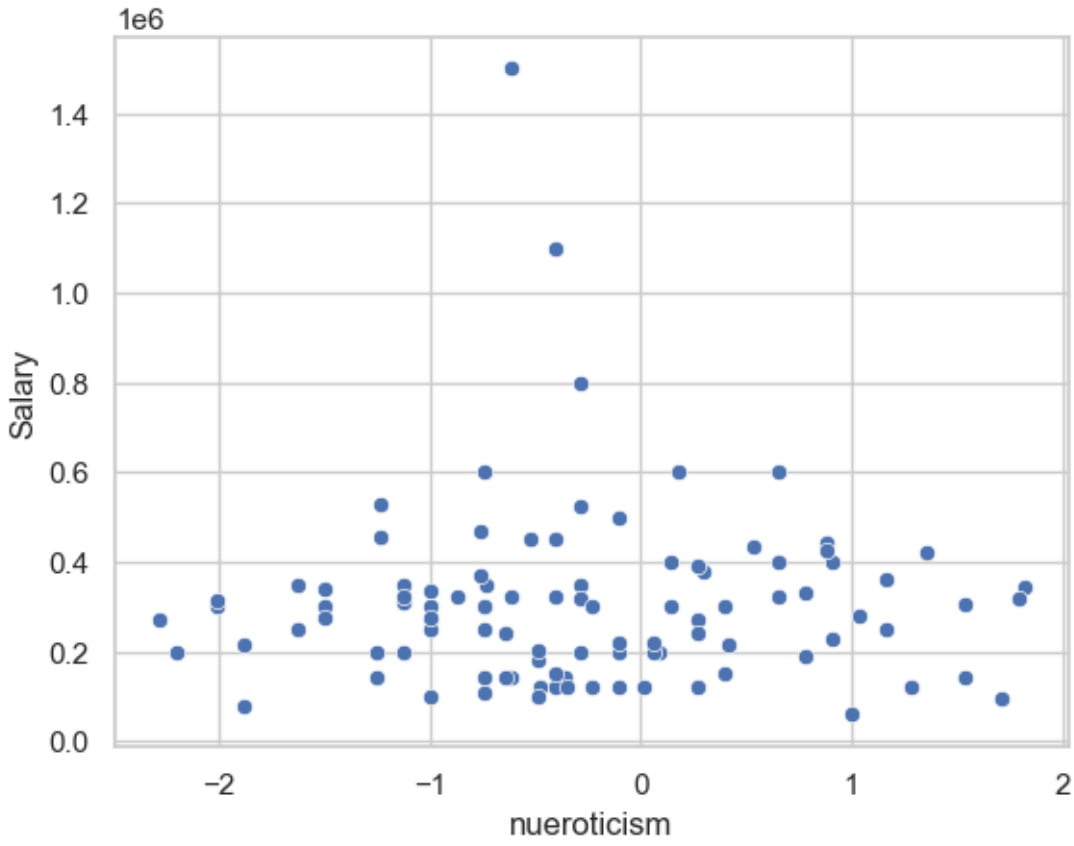
[98]:
```
# scatter plot for extraversion Vs Salary
sns.scatterplot(x=df_data['extraversion'].head(100),y=df_data['Salary'].
↪head(100))
```

[98]: <Axes: xlabel='extraversion', ylabel='Salary'>



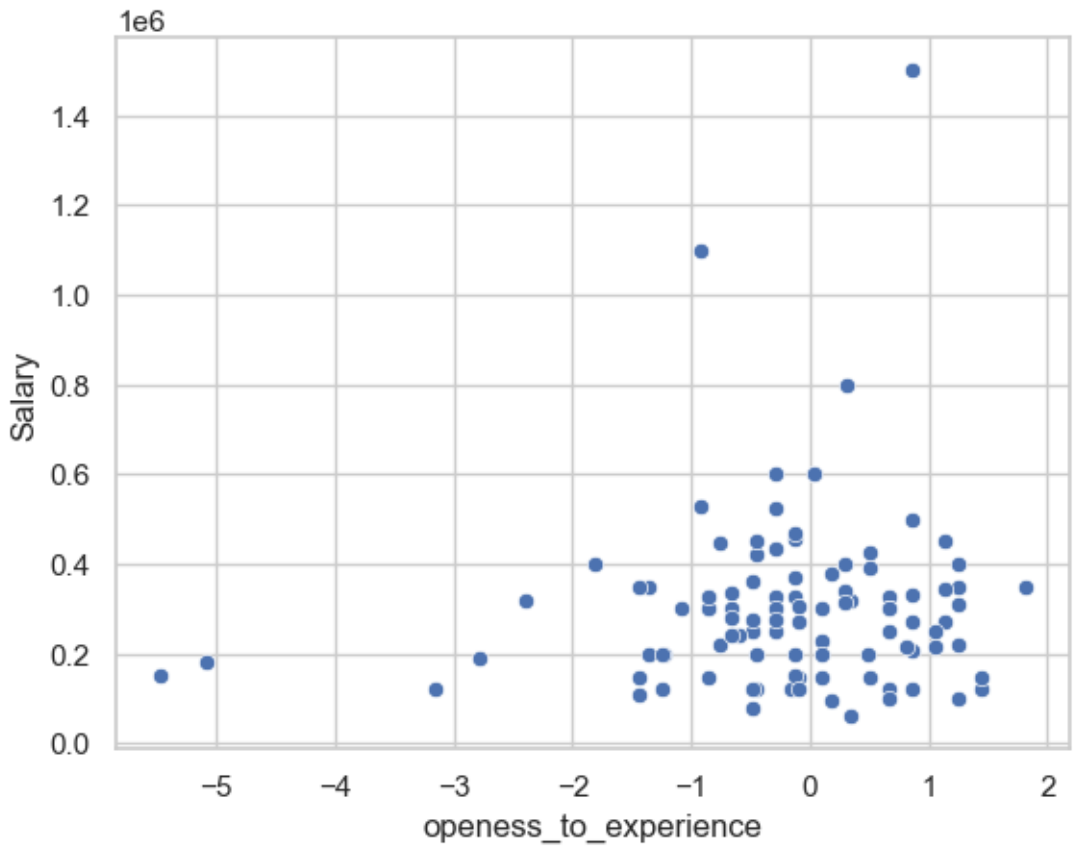
[99]:
```
# scatter plot for nueroticism Vs Salary
sns.scatterplot(x=df_data['nueroticism'].head(100),y=df_data['Salary'].
↪head(100))
```

[99]: <Axes: xlabel='nueroticism', ylabel='Salary'>



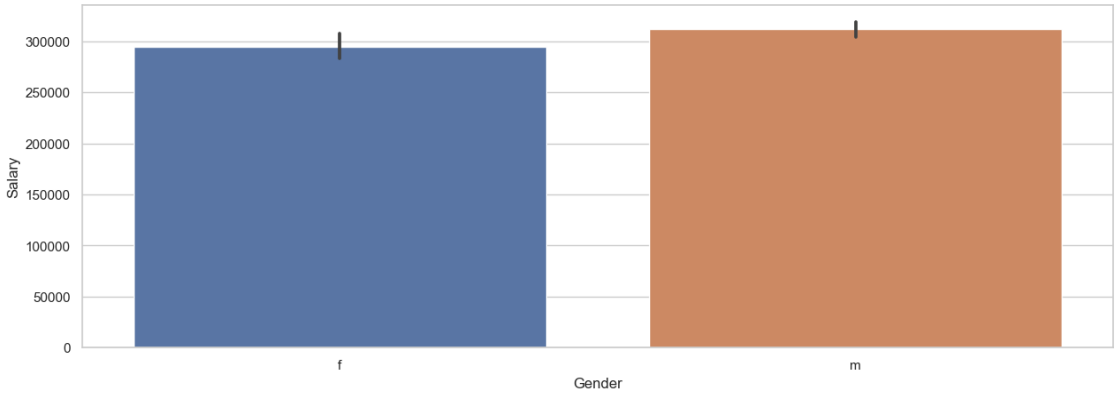
[100]:
```
# scatter plot for openness_to_experience Vs Salqry
sns.scatterplot(x=df_data['openess_to_experience'].
  ↪head(100),y=df_data['Salary'].head(100))
```

[100]: <Axes: xlabel='openess_to_experience', ylabel='Salary'>



[101]:
```
# Bar plot for Gender Vs Salary
import matplotlib.pyplot as plt
plt.figure(figsize=(15, 5))
sns.barplot(x=df_data['Gender'], y=df_data['Salary'])
plt.xticks(rotation='horizontal')
```

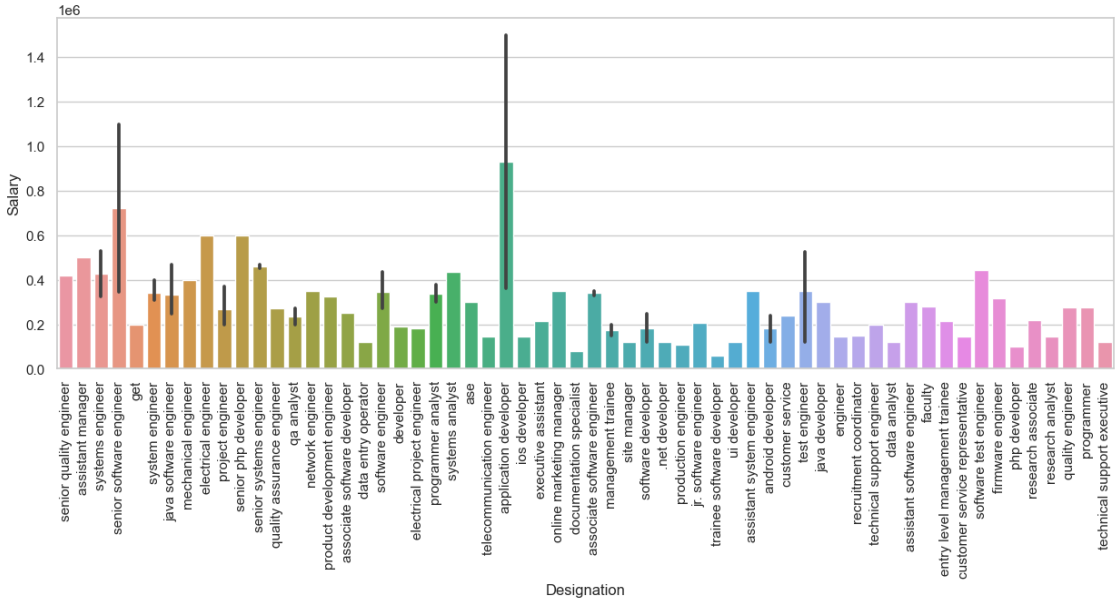
[101]: (array([0, 1]), [Text(0, 0, 'f'), Text(1, 0, 'm')])



[102]:
```
# Bar plot for Designation Vs Salary
import matplotlib.pyplot as plt
plt.figure(figsize=(15, 5))
sns.barplot(x=df_data['Designation'].head(100), y=df_data['Salary'].head(100))
plt.xticks(rotation='vertical')
```

[102]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59]),
 [Text(0, 0, 'senior quality engineer'),
  Text(1, 0, 'assistant manager'),
  Text(2, 0, 'systems engineer'),
  Text(3, 0, 'senior software engineer'),
  Text(4, 0, 'get'),
  Text(5, 0, 'system engineer'),
  Text(6, 0, 'java software engineer'),
  Text(7, 0, 'mechanical engineer'),
  Text(8, 0, 'electrical engineer'),
  Text(9, 0, 'project engineer'),
  Text(10, 0, 'senior php developer'),
  Text(11, 0, 'senior systems engineer'),
  Text(12, 0, 'quality assurance engineer'),
  Text(13, 0, 'qa analyst'),
  Text(14, 0, 'network engineer'),
  Text(15, 0, 'product development engineer'),
  Text(16, 0, 'associate software developer'),
  Text(17, 0, 'data entry operator'),
  Text(18, 0, 'software engineer'),
  Text(19, 0, 'developer'),
  Text(20, 0, 'electrical project engineer'),
  Text(21, 0, 'programmer analyst'),
  Text(22, 0, 'systems analyst'),
  Text(23, 0, 'ase'),
  Text(24, 0, 'telecommunication engineer'),
  Text(25, 0, 'application developer'),
  Text(26, 0, 'ios developer'),
  Text(27, 0, 'executive assistant'),
  Text(28, 0, 'online marketing manager'),
  Text(29, 0, 'documentation specialist'),
  Text(30, 0, 'associate software engineer'),
  Text(31, 0, 'management trainee'),
  Text(32, 0, 'site manager'),
  Text(33, 0, 'software developer'),
  Text(34, 0, '.net developer'),
  Text(35, 0, 'production engineer'),
```

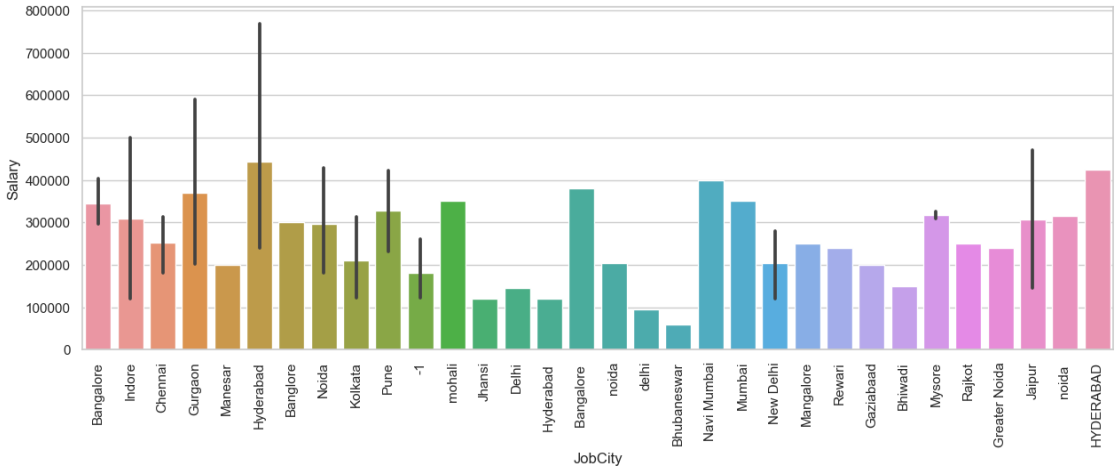
```
Text(36, 0, 'jr. software engineer'),
Text(37, 0, 'trainee software developer'),
Text(38, 0, 'ui developer'),
Text(39, 0, 'assistant system engineer'),
Text(40, 0, 'android developer'),
Text(41, 0, 'customer service'),
Text(42, 0, 'test engineer'),
Text(43, 0, 'java developer'),
Text(44, 0, 'engineer'),
Text(45, 0, 'recruitment coordinator'),
Text(46, 0, 'technical support engineer'),
Text(47, 0, 'data analyst'),
Text(48, 0, 'assistant software engineer'),
Text(49, 0, 'faculty'),
Text(50, 0, 'entry level management trainee'),
Text(51, 0, 'customer service representative'),
Text(52, 0, 'software test engineer'),
Text(53, 0, 'firmware engineer'),
Text(54, 0, 'php developer'),
Text(55, 0, 'research associate'),
Text(56, 0, 'research analyst'),
Text(57, 0, 'quality engineer'),
Text(58, 0, 'programmer'),
Text(59, 0, 'technical support executive')])
```



[103]:
```
# Bar plot for JobCity Vs Salary
import matplotlib.pyplot as plt
```

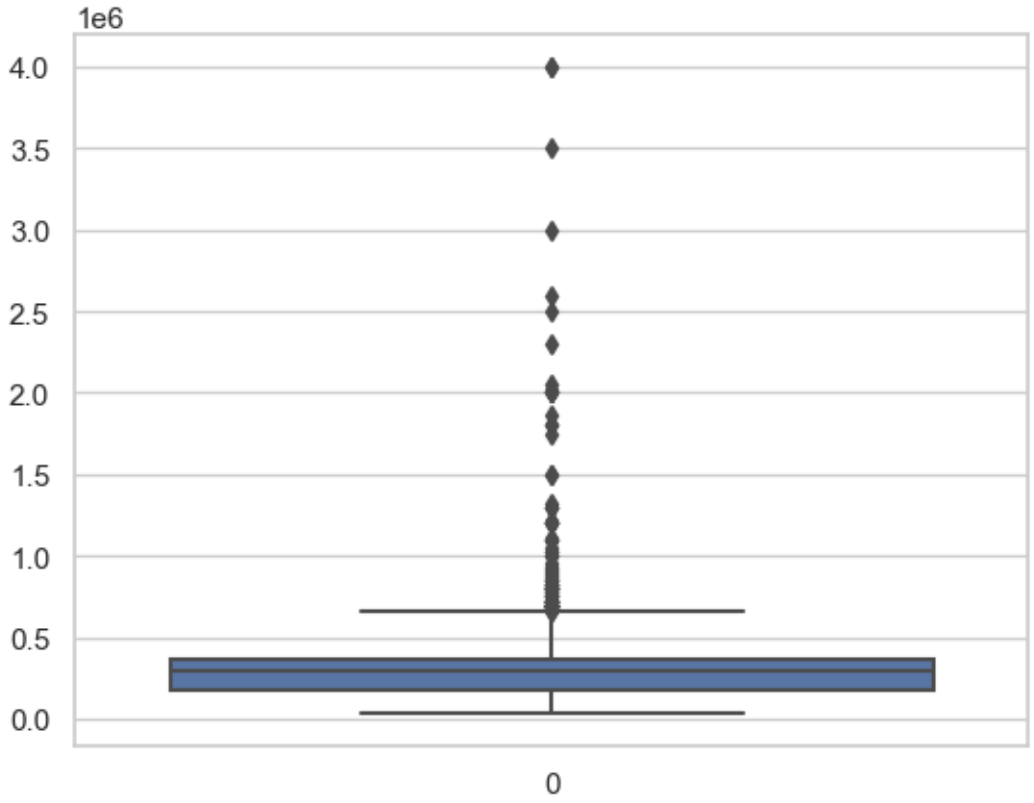
```
plt.figure(figsize=(15, 5))
sns.barplot(x=df_data['JobCity'].head(100), y=df_data['Salary'].head(100))
plt.xticks(rotation='vertical')
```

[103]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]),
 [Text(0, 0, 'Bangalore'),
  Text(1, 0, 'Indore'),
  Text(2, 0, 'Chennai'),
  Text(3, 0, 'Gurgaon'),
  Text(4, 0, 'Manesar'),
  Text(5, 0, 'Hyderabad'),
  Text(6, 0, 'Banglore'),
  Text(7, 0, 'Noida'),
  Text(8, 0, 'Kolkata'),
  Text(9, 0, 'Pune'),
  Text(10, 0, '-1'),
  Text(11, 0, 'mohali'),
  Text(12, 0, 'Jhansi'),
  Text(13, 0, 'Delhi'),
  Text(14, 0, 'Hyderabad '),
  Text(15, 0, 'Bangalore '),
  Text(16, 0, 'noida'),
  Text(17, 0, 'delhi'),
  Text(18, 0, 'Bhubaneswar'),
  Text(19, 0, 'Navi Mumbai'),
  Text(20, 0, 'Mumbai'),
  Text(21, 0, 'New Delhi'),
  Text(22, 0, 'Mangalore'),
  Text(23, 0, 'Rewari'),
  Text(24, 0, 'Gaziabaad'),
  Text(25, 0, 'Bhiwadi'),
  Text(26, 0, 'Mysore'),
  Text(27, 0, 'Rajkot'),
  Text(28, 0, 'Greater Noida'),
  Text(29, 0, 'Jaipur'),
  Text(30, 0, 'noida '),
  Text(31, 0, 'HYDERABAD')])
```

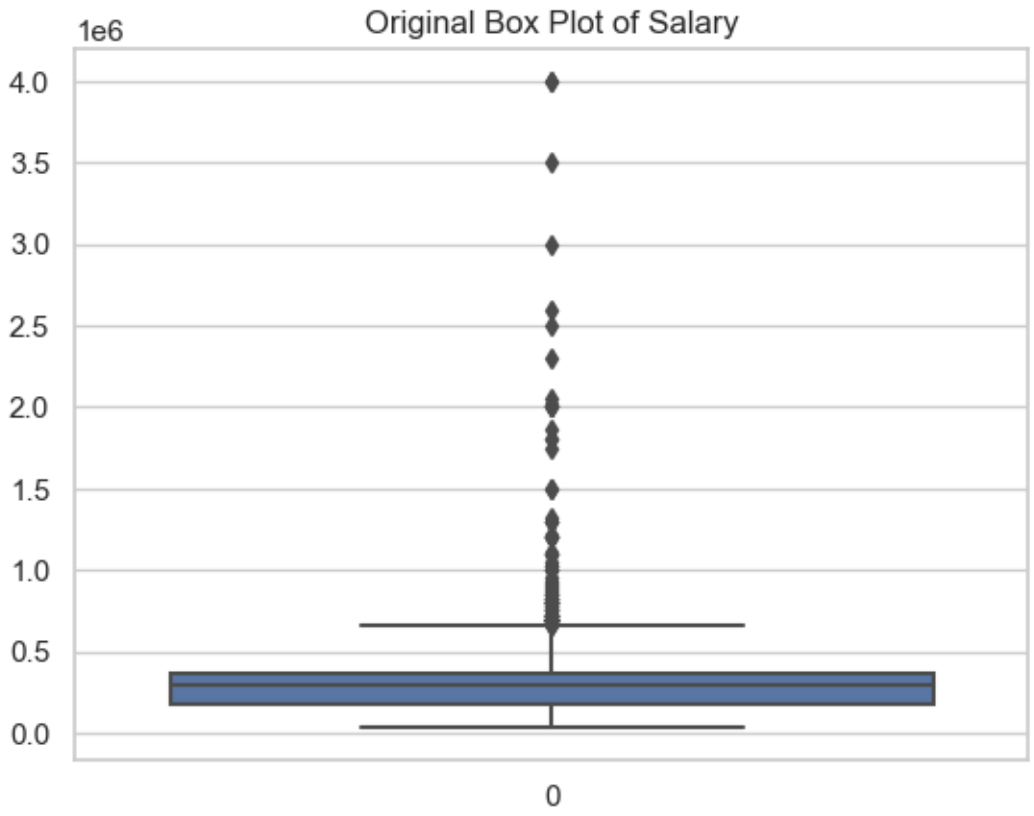


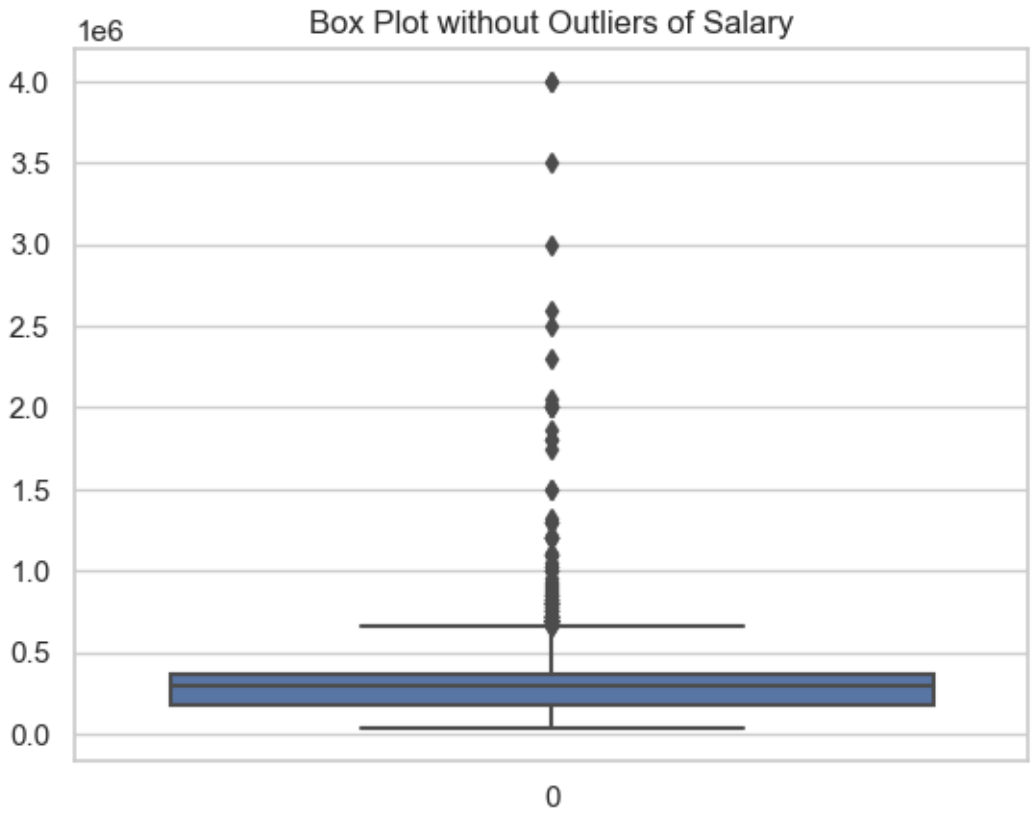
[104]:
```
# Box plot for Salary
import seaborn as sns
sns.boxplot(df_data['Salary'])
```

[104]: <Axes: >



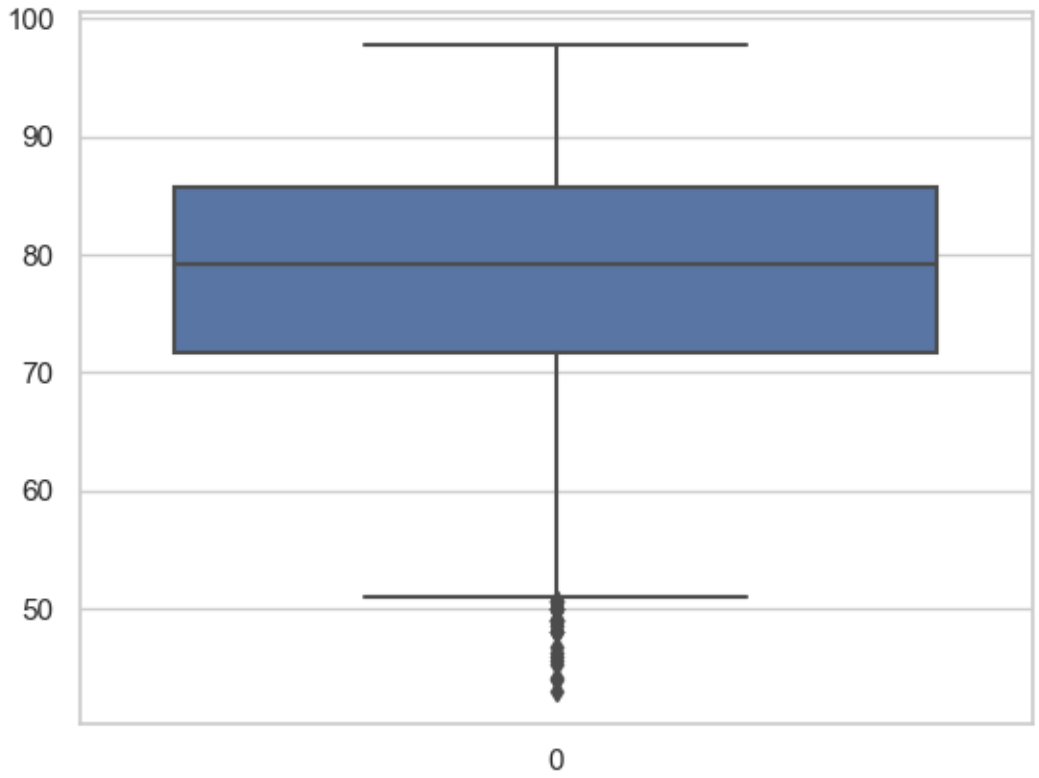
[105]:
```
import seaborn as sns
import matplotlib.pyplot as plt
column='Salary'
threshold = 0.6
sns.boxplot(df_data['Salary'])
plt.title(f'Original Box Plot of {column}')
plt.show()
removed_outliers = df_data[df_data['Salary'] >= threshold]
sns.boxplot(removed_outliers['Salary'])
plt.title(f'Box Plot without Outliers of {column}')
plt.show()
```



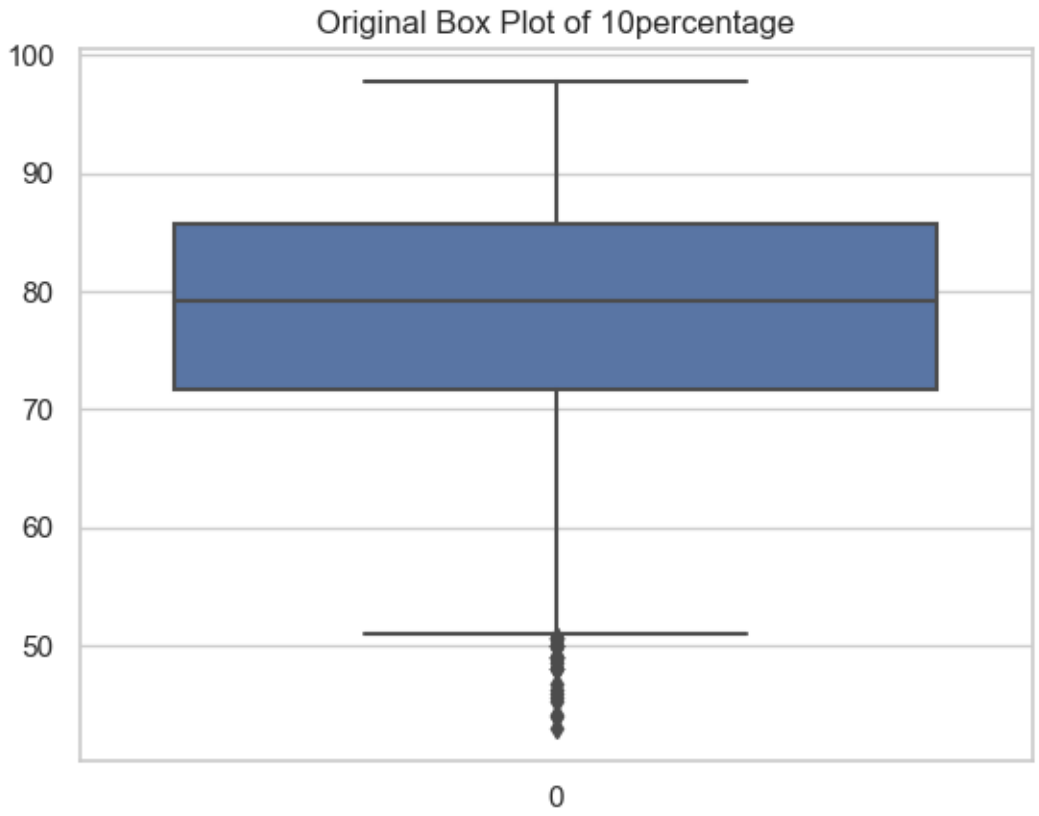


[106]:
```
# Box Plot for 10percentage
import seaborn as sns
sns.boxplot(df_data['10percentage'])
```

[106]: <Axes: >

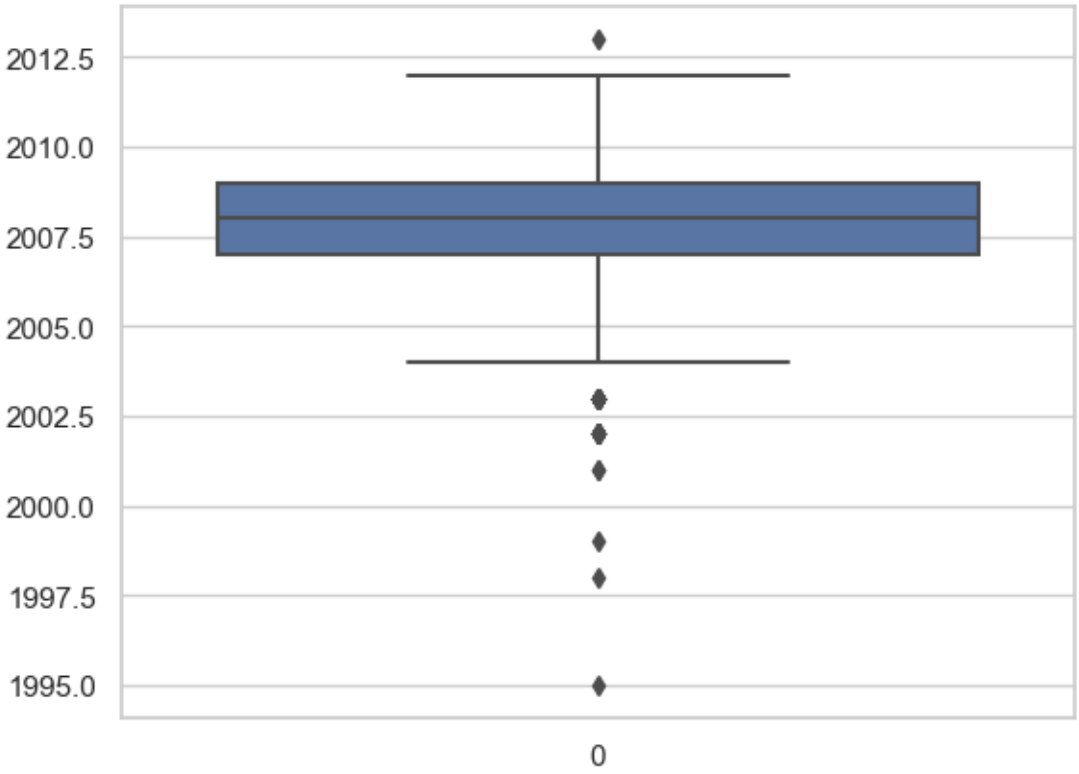


[107]:
```
import seaborn as sns
import matplotlib.pyplot as plt
column='10percentage'
threshold = 55
sns.boxplot(df_data['10percentage'])
plt.title(f'Original Box Plot of {column}')
plt.show()
```



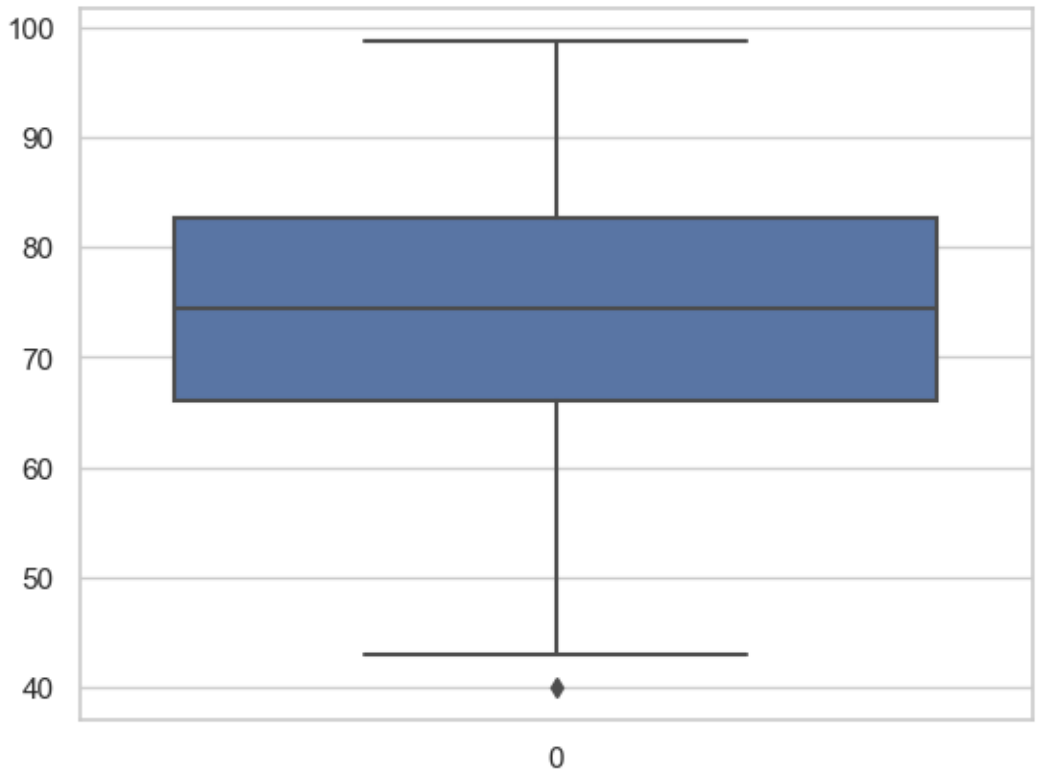
[108]:
```
# Box plot for 12graduation
import seaborn as sns
sns.boxplot(df_data['12graduation'])
```

[108]: <Axes: >



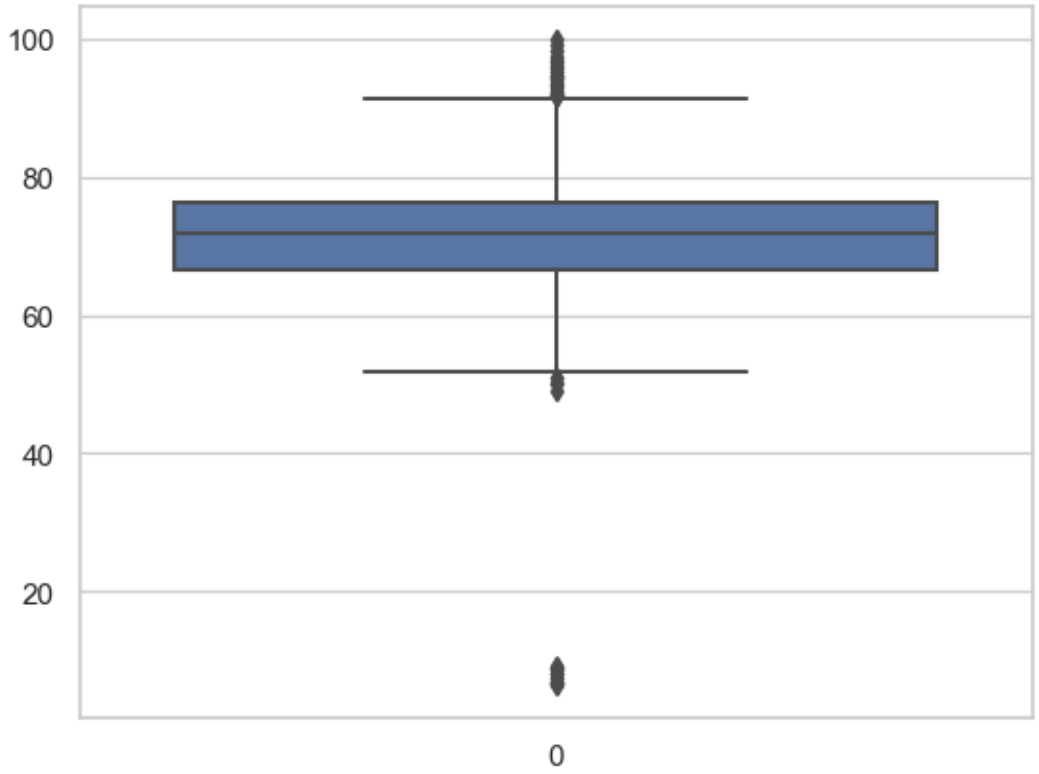
[109]:
```
#Box plot for 12percentage
import seaborn as sns
sns.boxplot(df_data['12percentage'])
```

[109]: <Axes: >



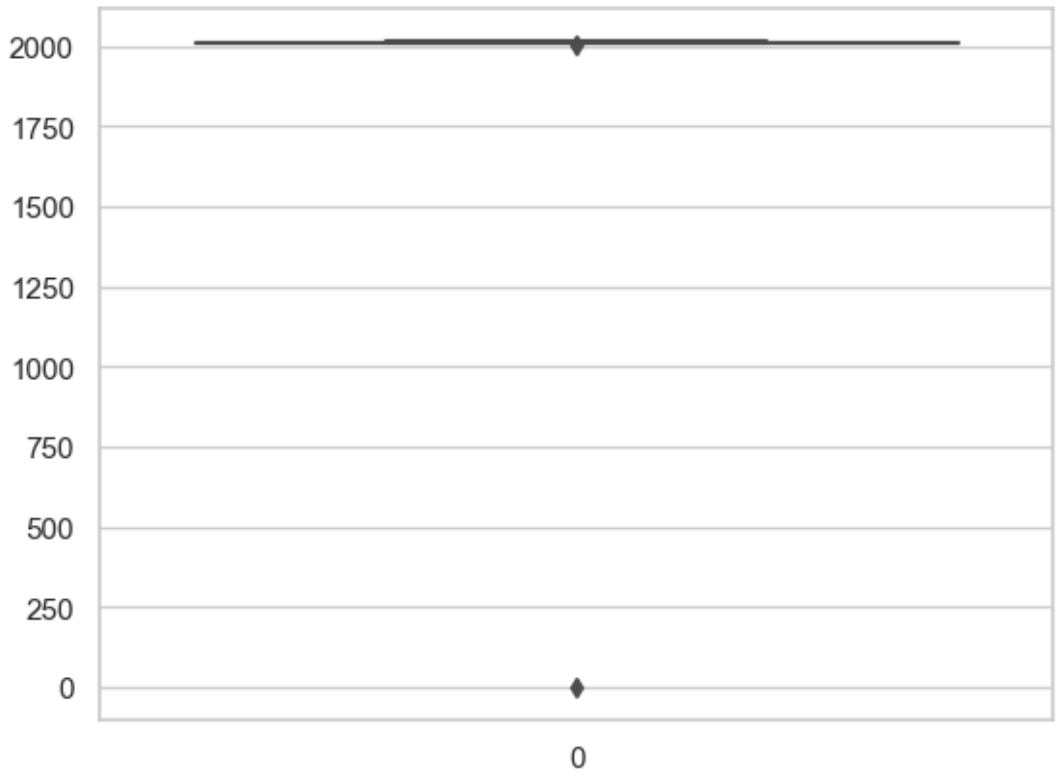
[110]:
```
# Box plot for collegeGPA
import seaborn as sns
sns.boxplot(df_data['collegeGPA'])
```

[110]: <Axes: >



[111]:
```
#Box plot for Graduation Year
import seaborn as sns
sns.boxplot(df_data['GraduationYear'])
```

[111]: <Axes: >



[112]:
```
# Compute Quartile1,Quartile3 and IQR for Salary
Quartile1 = np.percentile(df_data['Salary'], 25, method='midpoint')
Quartile3 = np.percentile(df_data['Salary'], 75, method='midpoint')
IQR = Quartile3 - Quartile1
print(IQR)
```

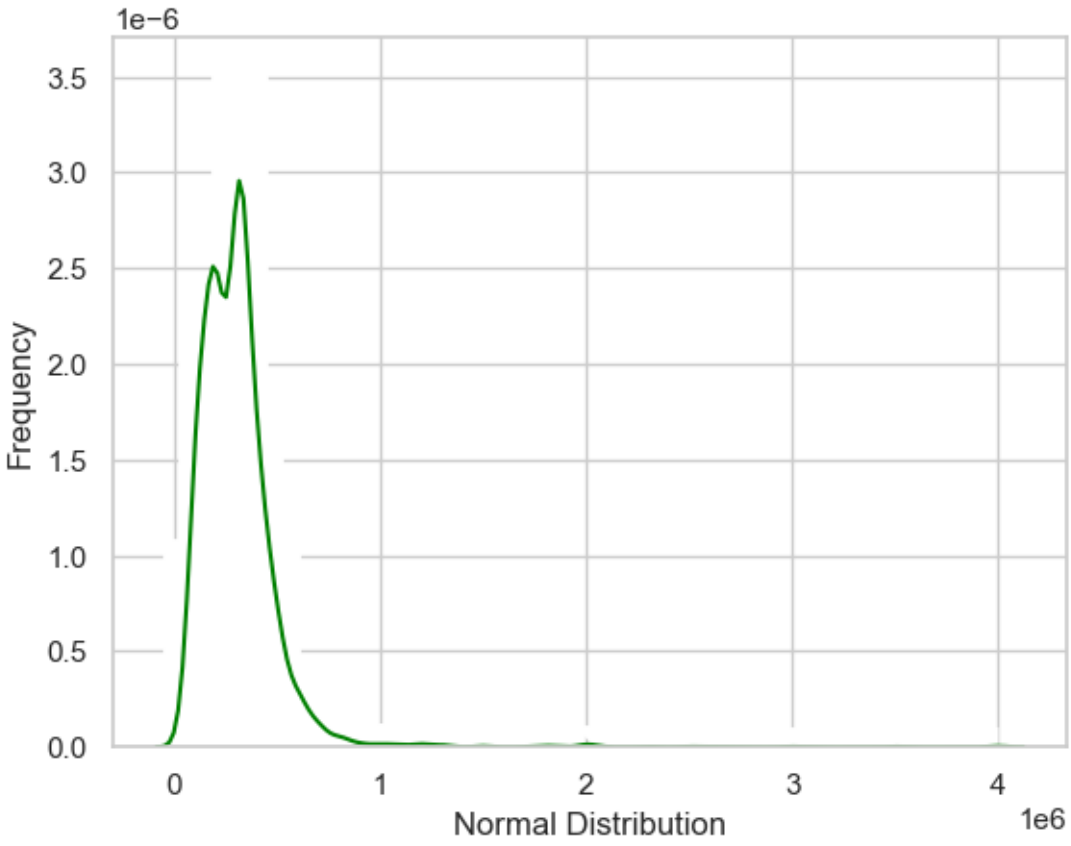
```
190000.0
```

[113]:
```
# Compute Upper Quartile and Lower Quartile
upper_quartile = Quartile3+1.5*IQR
upper_array_salary = np.array(df_data['Salary'] >= upper_quartile)
print("Upper Bound:", upper_quartile)
print(upper_array_salary.sum())
lower_quartile = Quartile1-1.5*IQR
lower_array_salary = np.array(df_data['Salary'] <= lower_quartile)
print("Lower Bound:", lower_quartile)
print(lower_array_salary.sum())
```

```
Upper Bound: 655000.0
112
Lower Bound: -105000.0
0
```

[114]:
```
# Normal Distribution for Salary
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution = sns.distplot(df_data['Salary'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution.set(xlabel='Normal Distribution', ylabel='Frequency')

plt.show()
```

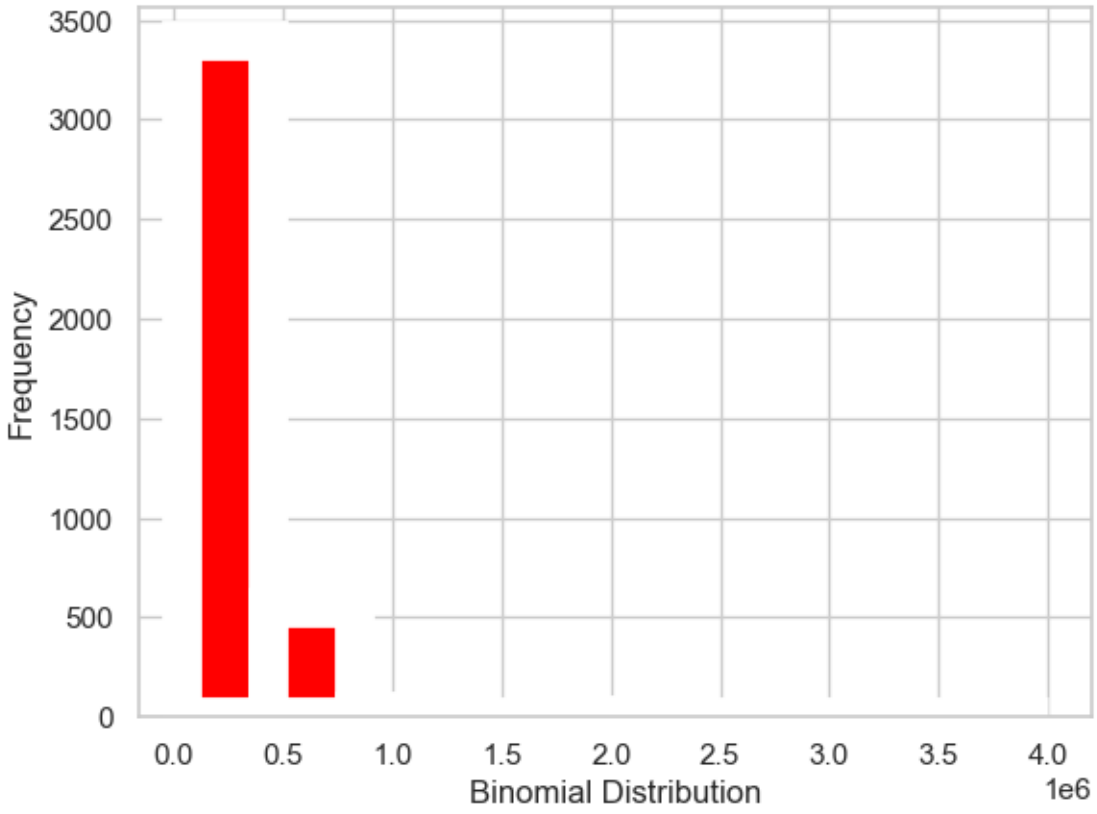
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\869582968.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution = sns.distplot(df_data['Salary'],
```



[115]:
```
#Binomial Distribution for Salary
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_salary = sns.distplot(df_data['Salary'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_salary.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3126793656.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

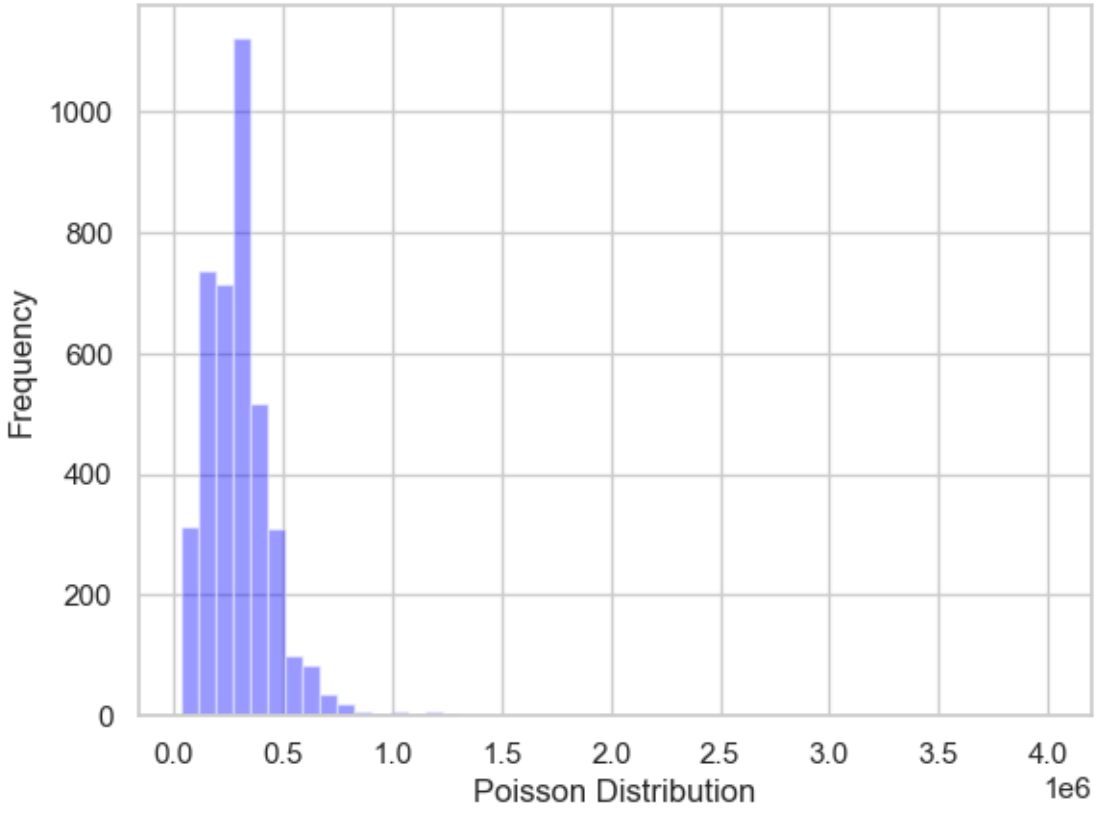
```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_salary = sns.distplot(df_data['Salary'],
```



[116]:
```
# Poisson's distribution for Salary
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_salary = sns.distplot(df_data['Salary'],
                  kde=False,
                  color='blue')
poisson_distribution_salary.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\4057766026.py:7: UserWarning:
```

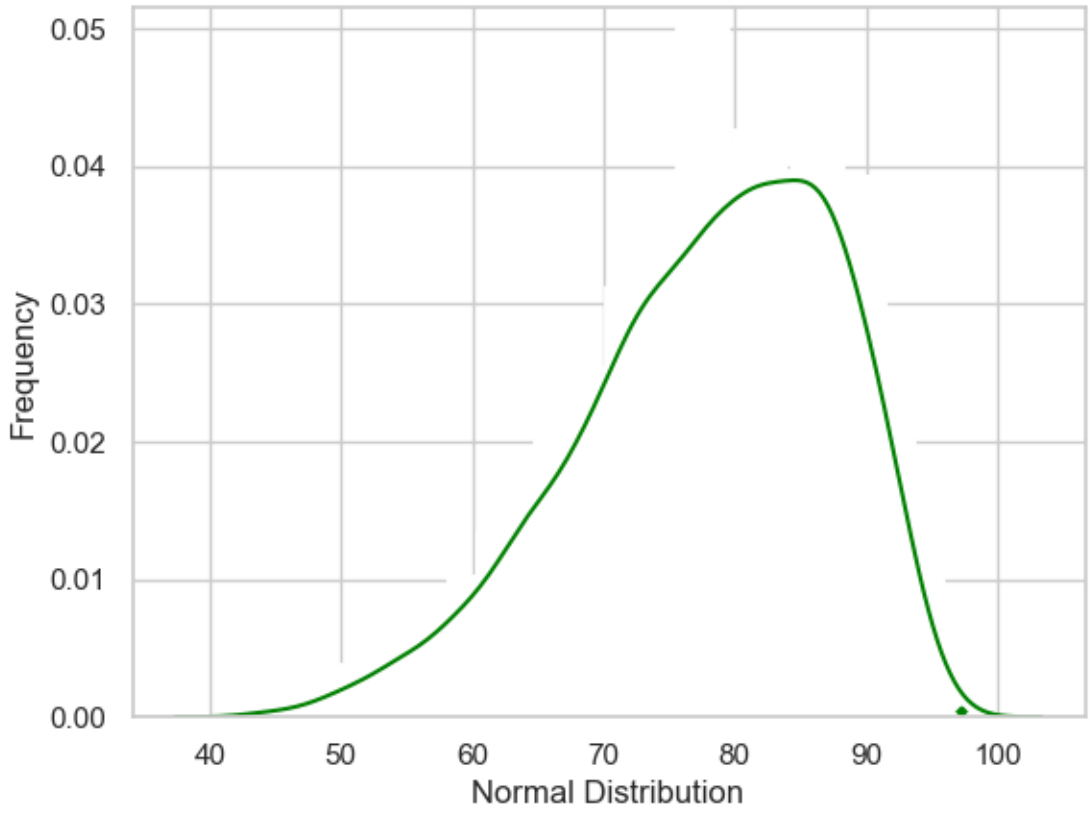
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_salary = sns.distplot(df_data['Salary'],
```



[117]:
```
# Normal Distribution for 10 percentage
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution = sns.distplot(df_data['10percentage'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution.set(xlabel='Normal Distribution', ylabel='Frequency')
```

```
plt.show()
```

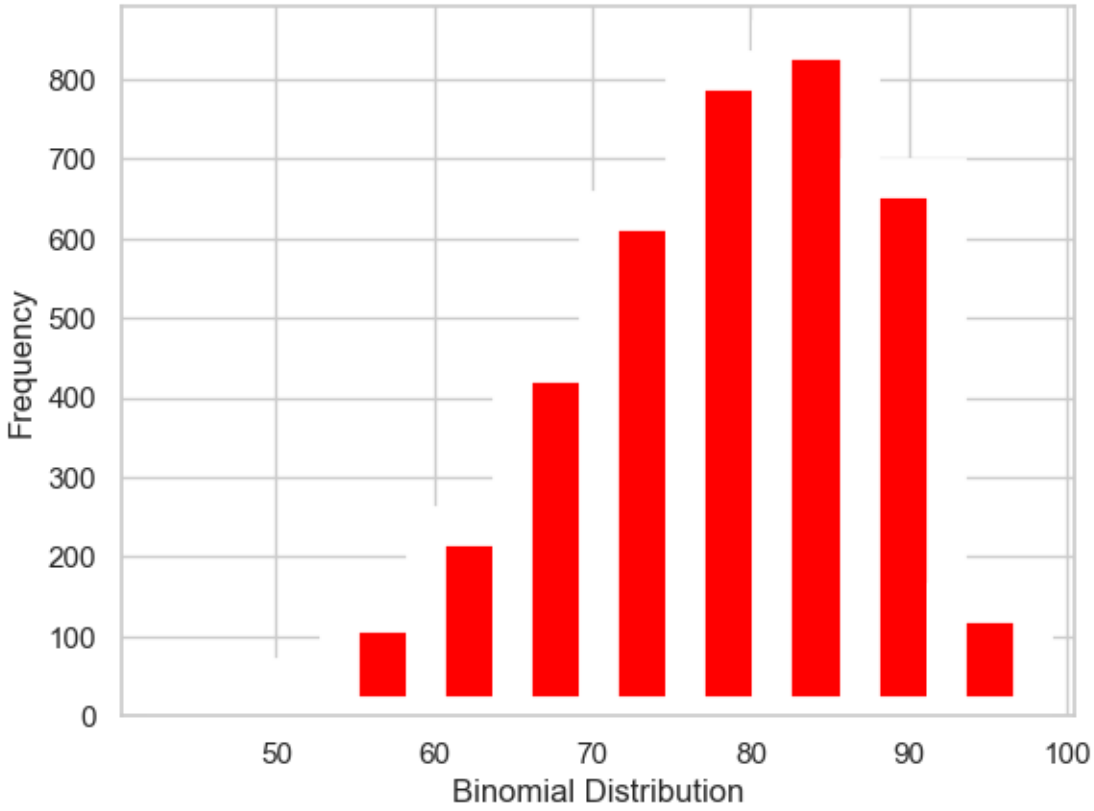
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1916135650.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution = sns.distplot(df_data['10percentage'],
```



[118]:
```
# Binomial Distribution for 10 percentage
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_10percentage = sns.distplot(df_data['10percentage'],
                  bins=10,
```

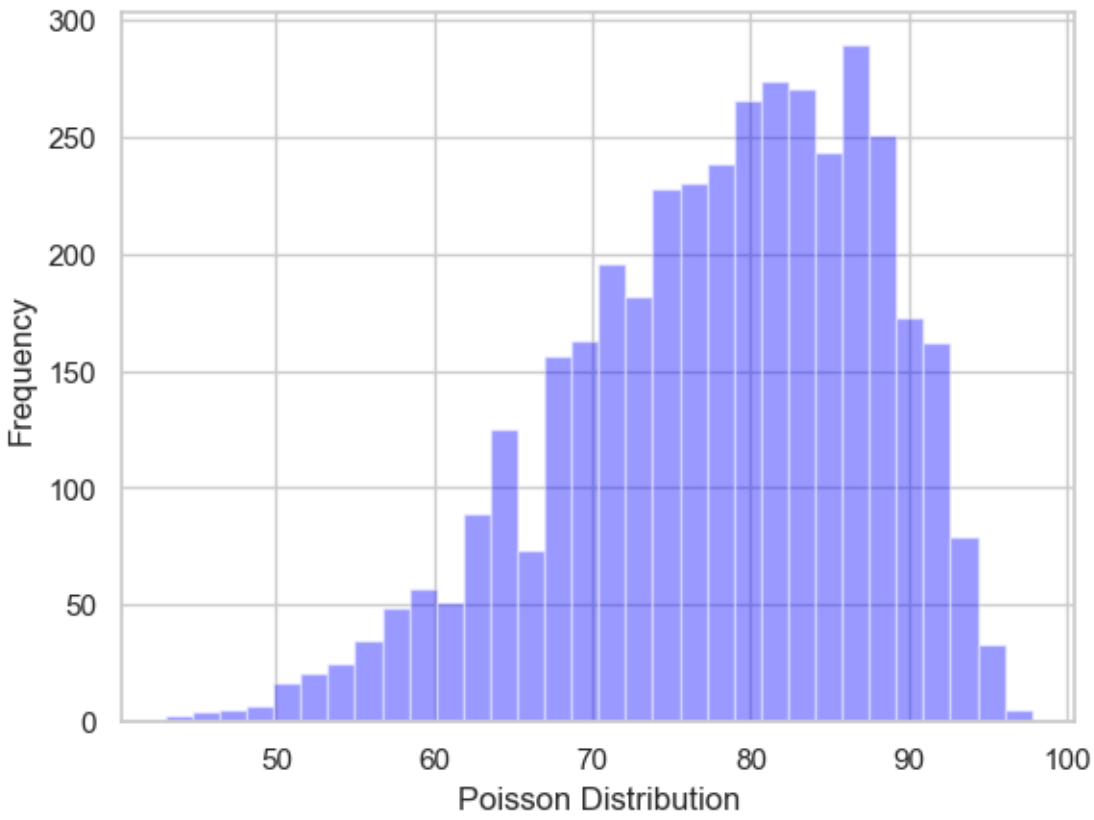
```
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_10percentage.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1499863000.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_10percentage = sns.distplot(df_data['10percentage'],
```



[119]:
```
# Poisson's Distribution for 10 percentage
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_10percentage = sns.distplot(df_data['10percentage'],
                  kde=False,
                  color='blue')
poisson_distribution_10percentage.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

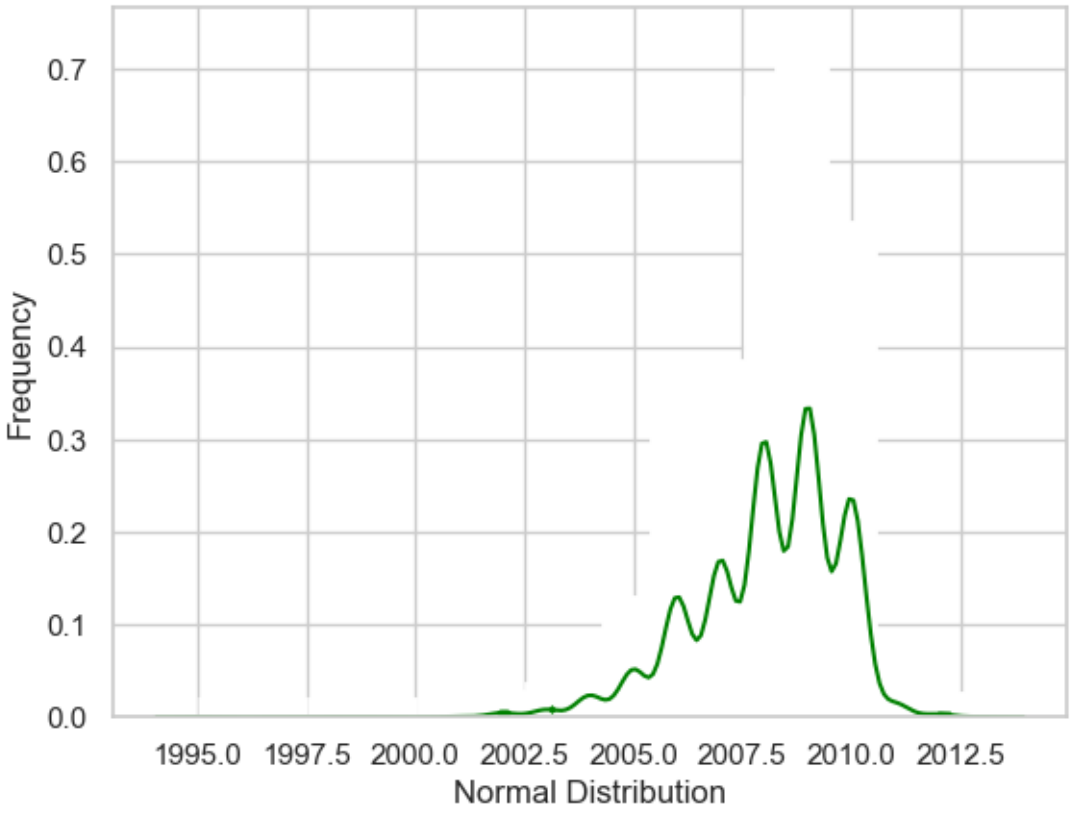
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1151869874.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_10percentage = sns.distplot(df_data['10percentage'],
```



[120]:
```
# Normal Distribution for 12 graduation
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_12graduation= sns.distplot(df_data['12graduation'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_12graduation.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2686432086.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

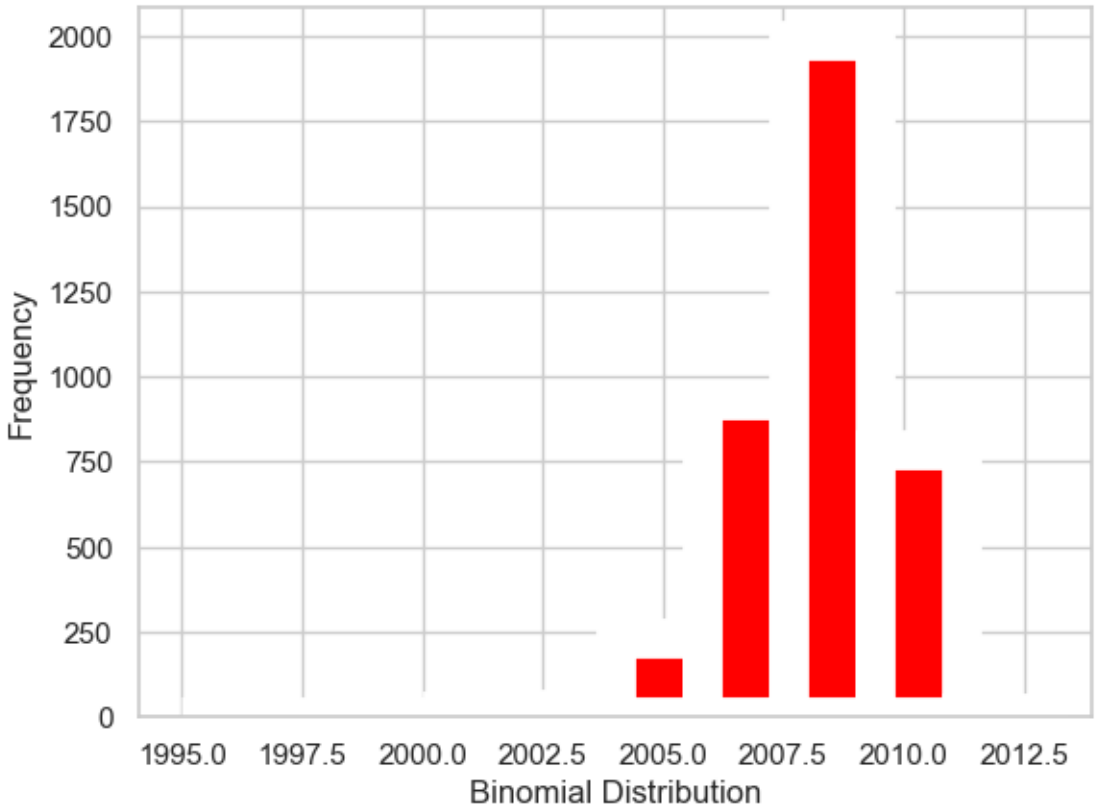
```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_12graduation= sns.distplot(df_data['12graduation'],
```



[121]:
```
# Binomial Distribution for 12 graduation
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_12graduation = sns.distplot(df_data['12graduation'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_12graduation.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3068263270.py:5: UserWarning:
```

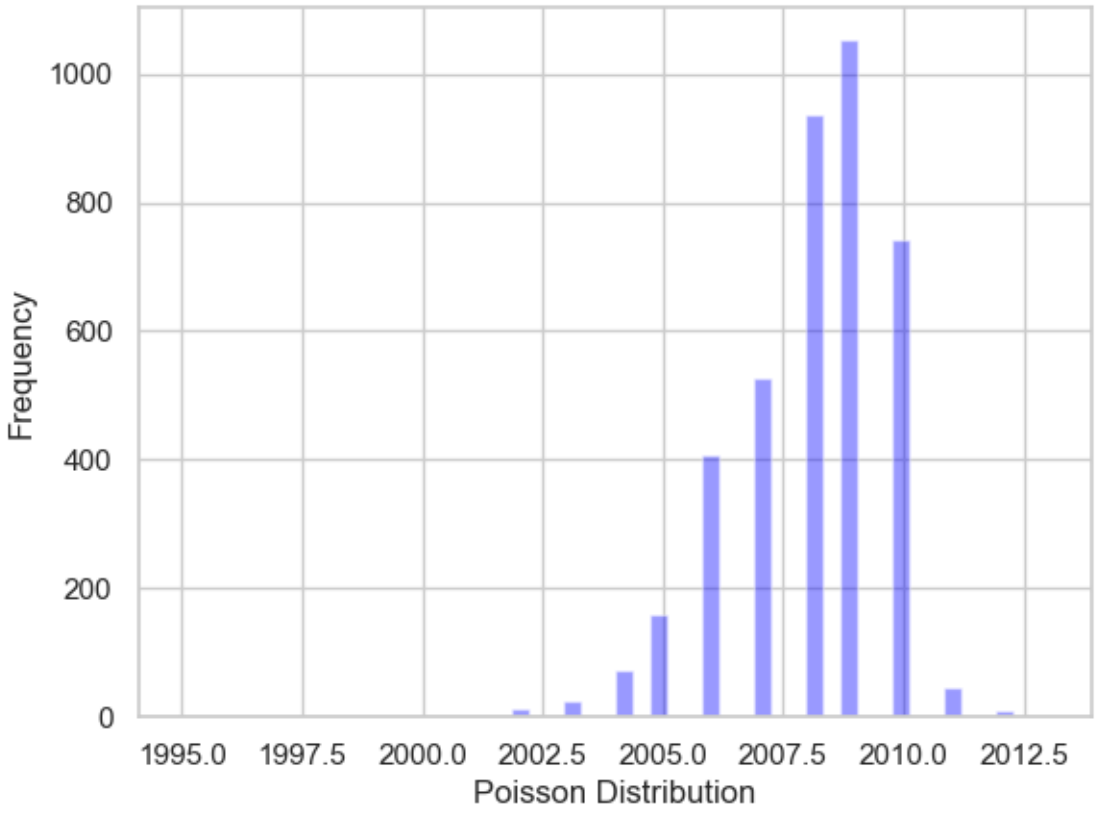
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_12graduation = sns.distplot(df_data['12graduation'],
```



[122]:
```
# Poissons Distribution for 12 Graduation
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_12graduation = sns.distplot(df_data['12graduation'],
                 kde=False,
                 color='blue')
poisson_distribution_12graduation.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')
```

```
plt.show()
```

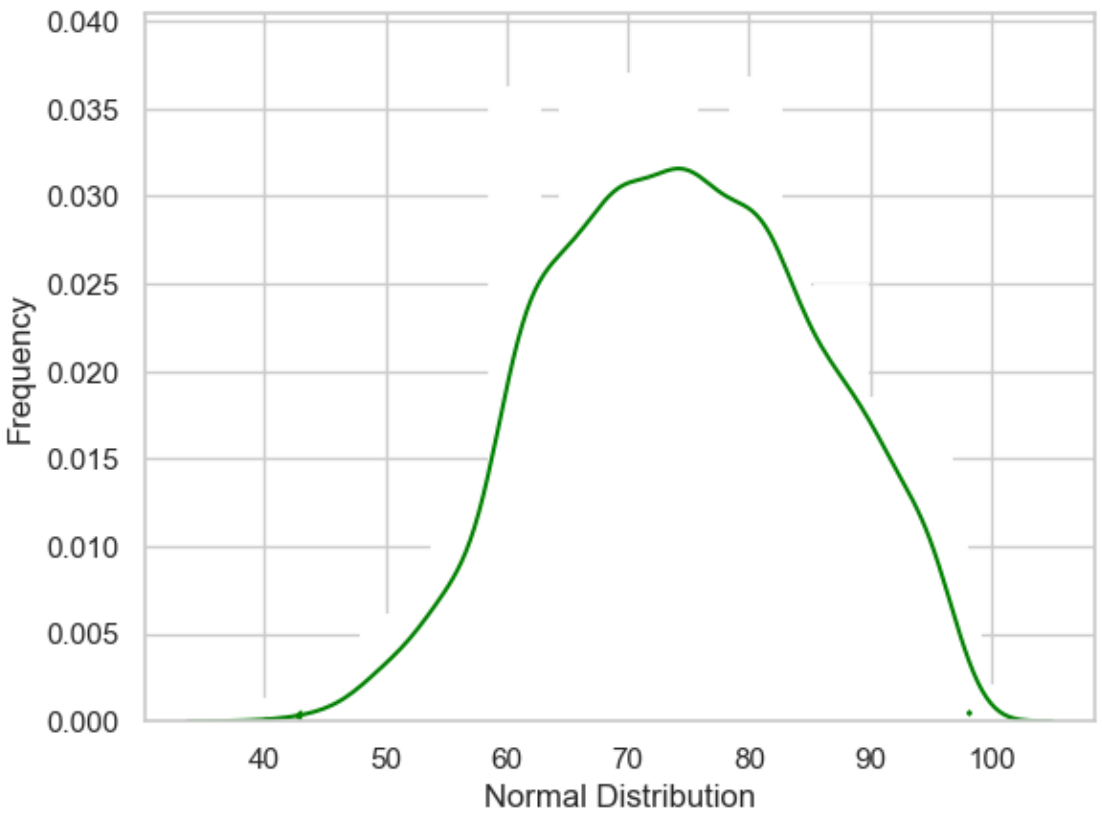
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2755439095.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_12graduation = sns.distplot(df_data['12graduation'],
```



[123]:
```
# Normal Distribution for 12 percentage

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_12percentage= sns.distplot(df_data['12percentage'],
```

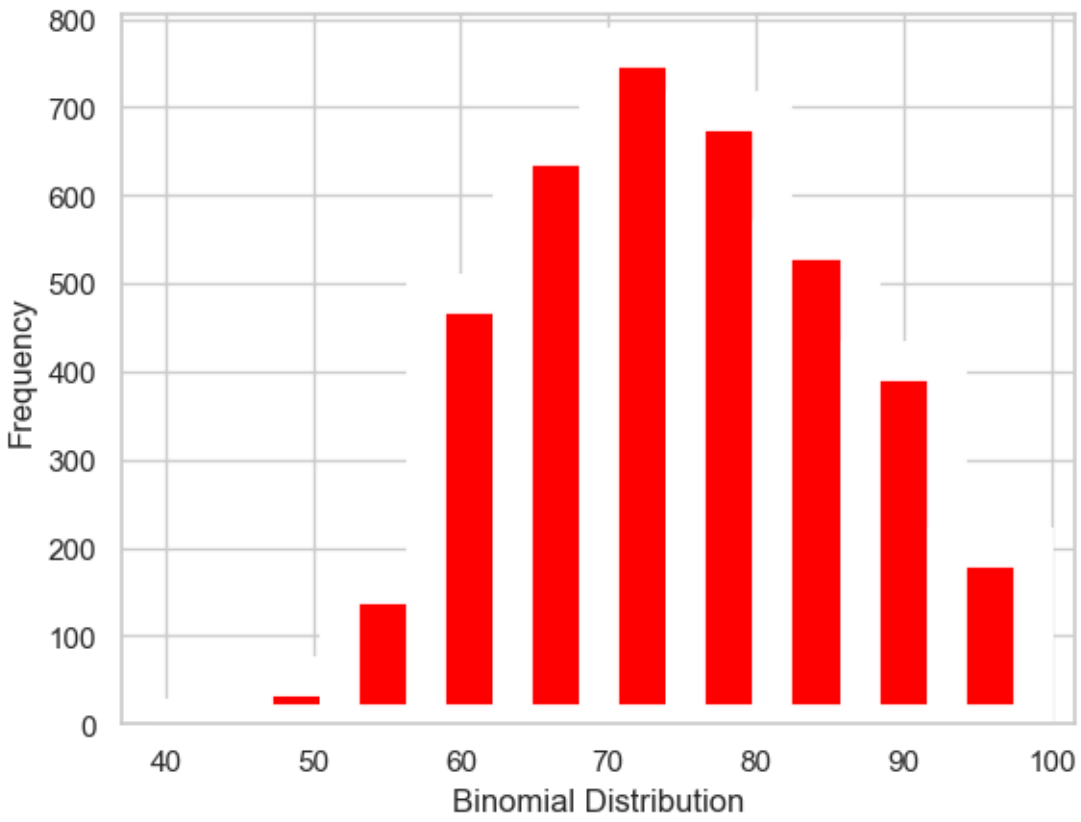
```
                    bins=50,
                    kde=True,
                    color='green',
                    hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_12percentage.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\693724528.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_12percentage= sns.distplot(df_data['12percentage'],
```



[124]:
```
# Binomial Distribution for 12 percentage
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_12percentage = sns.distplot(df_data['12percentage'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_12percentage.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\4200962130.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_12percentage = sns.distplot(df_data['12percentage'],
```

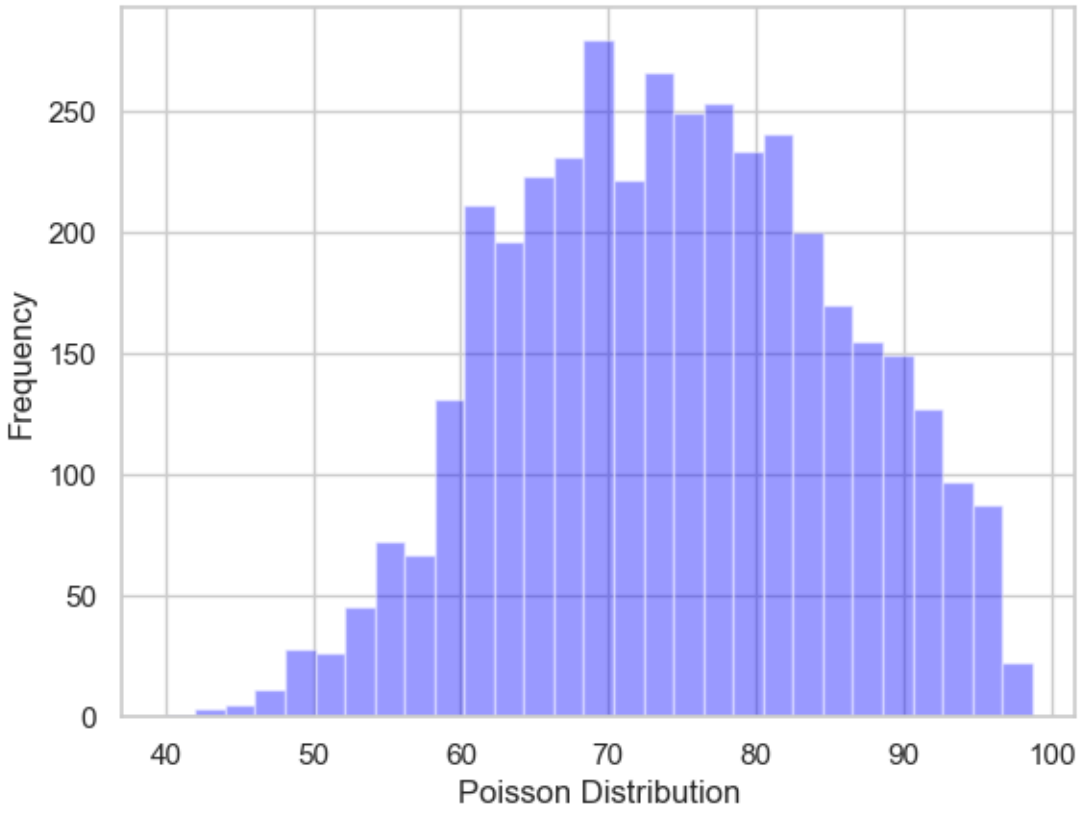


[125]:
```
# Poisson Distribution for 12 percentage
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_12percentage = sns.distplot(df_data['12percentage'],
                  kde=False,
                  color='blue')
poisson_distribution_12percentage.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1893170946.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
```

https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

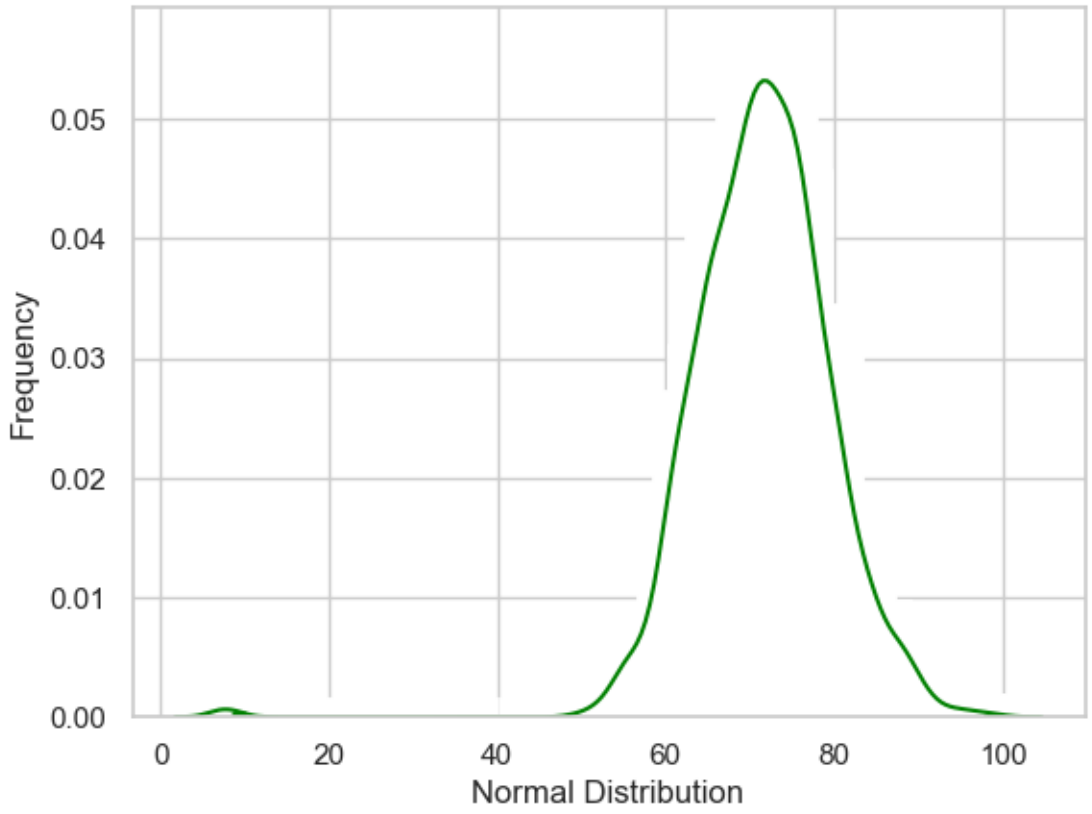
```
poisson_distribution_12percentage = sns.distplot(df_data['12percentage'],
```



[126]:
```
# Normal Distribution for college GPA
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_collegeGPA= sns.distplot(df_data['collegeGPA'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_collegeGPA.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1758807521.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

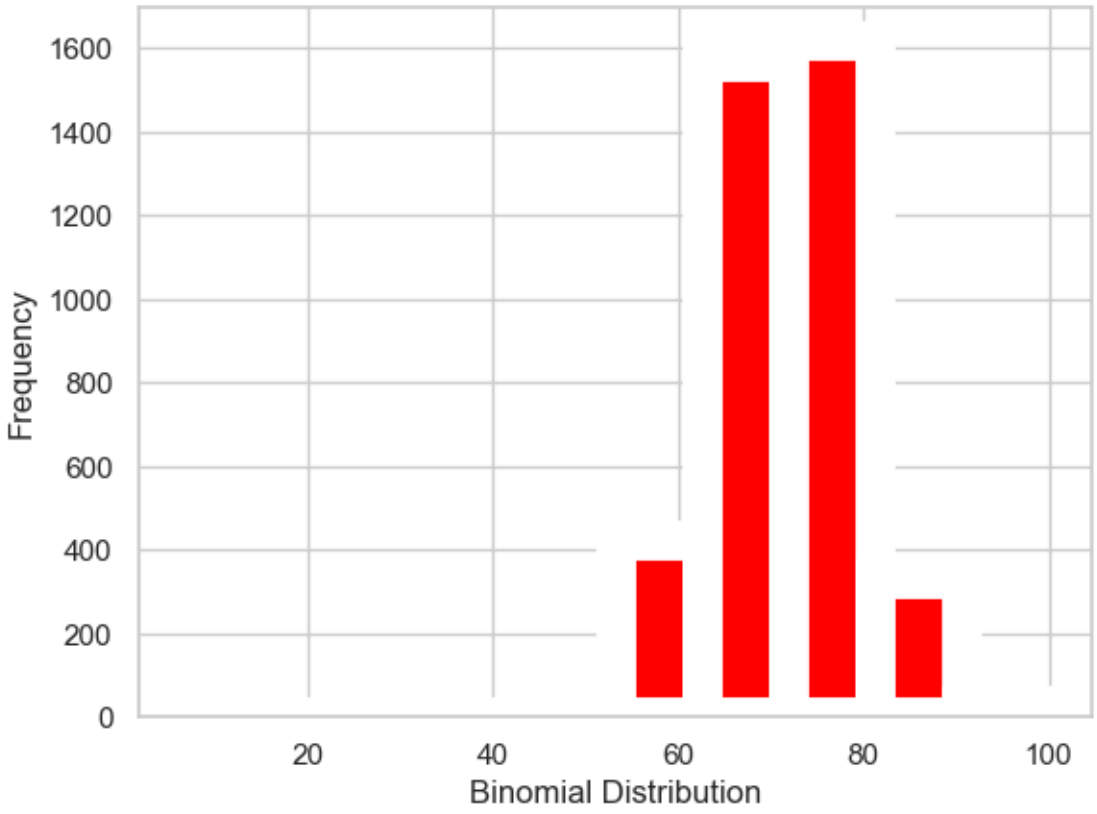
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_collegeGPA= sns.distplot(df_data['collegeGPA'],
```



[127]:
```
# Binomial Distribution for college GPA
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_collegeGPA = sns.distplot(df_data['collegeGPA'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_collegeGPA.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')
```

```
plt.show()
```

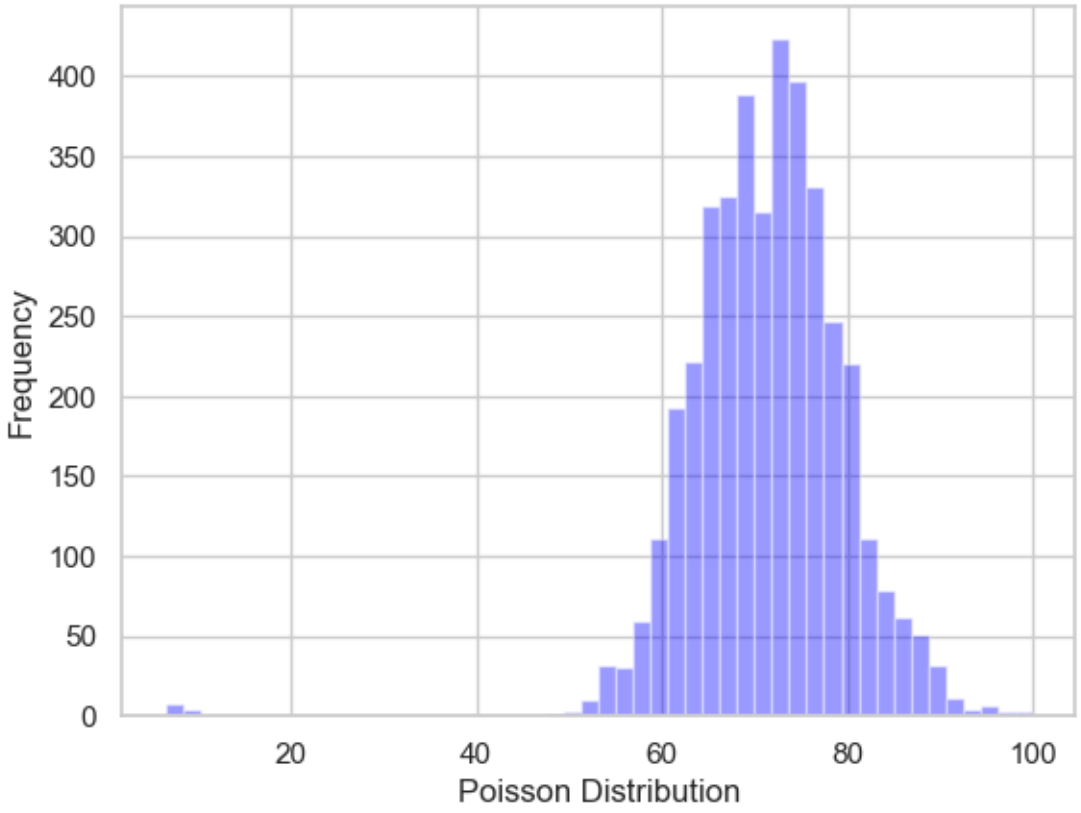
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2572854532.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_collegeGPA = sns.distplot(df_data['collegeGPA'],
```



[128]:
```
# Poisson Distribution for college GPA
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_collegeGPA = sns.distplot(df_data['collegeGPA'],
```

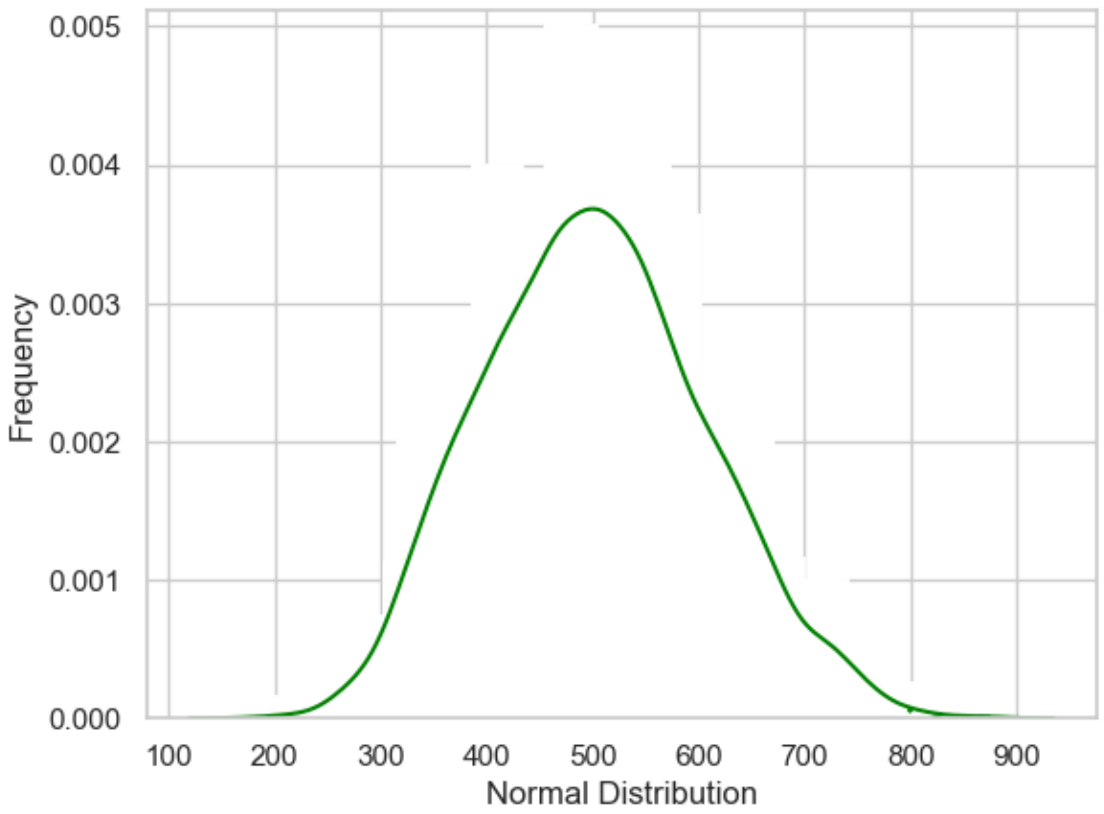
```
                  kde=False,
                  color='blue')
poisson_distribution_collegeGPA.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2479398859.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_collegeGPA = sns.distplot(df_data['collegeGPA'],
```



[129]:
```
# Normal Distribution for Englisg
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_English= sns.distplot(df_data['English'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_English.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

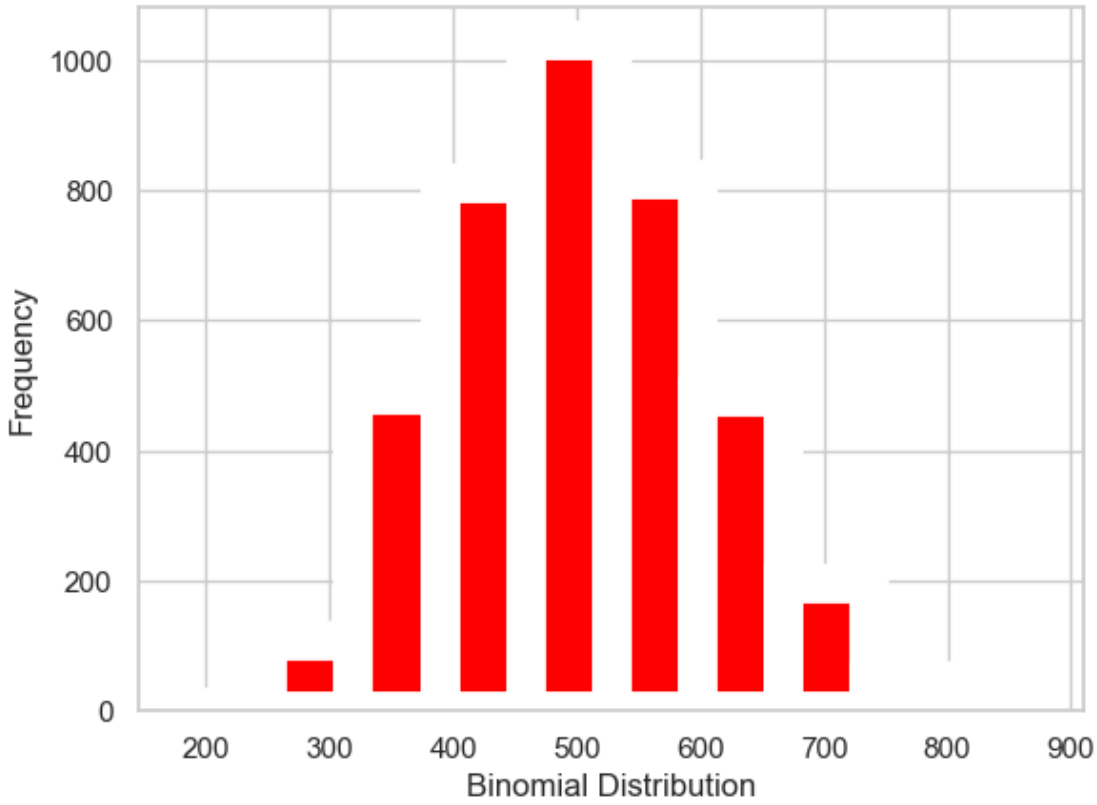
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\994274159.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_English= sns.distplot(df_data['English'],
```



[130]:
```
# Binomial Distribution for English
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_English = sns.distplot(df_data['English'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_English.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1211841180.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

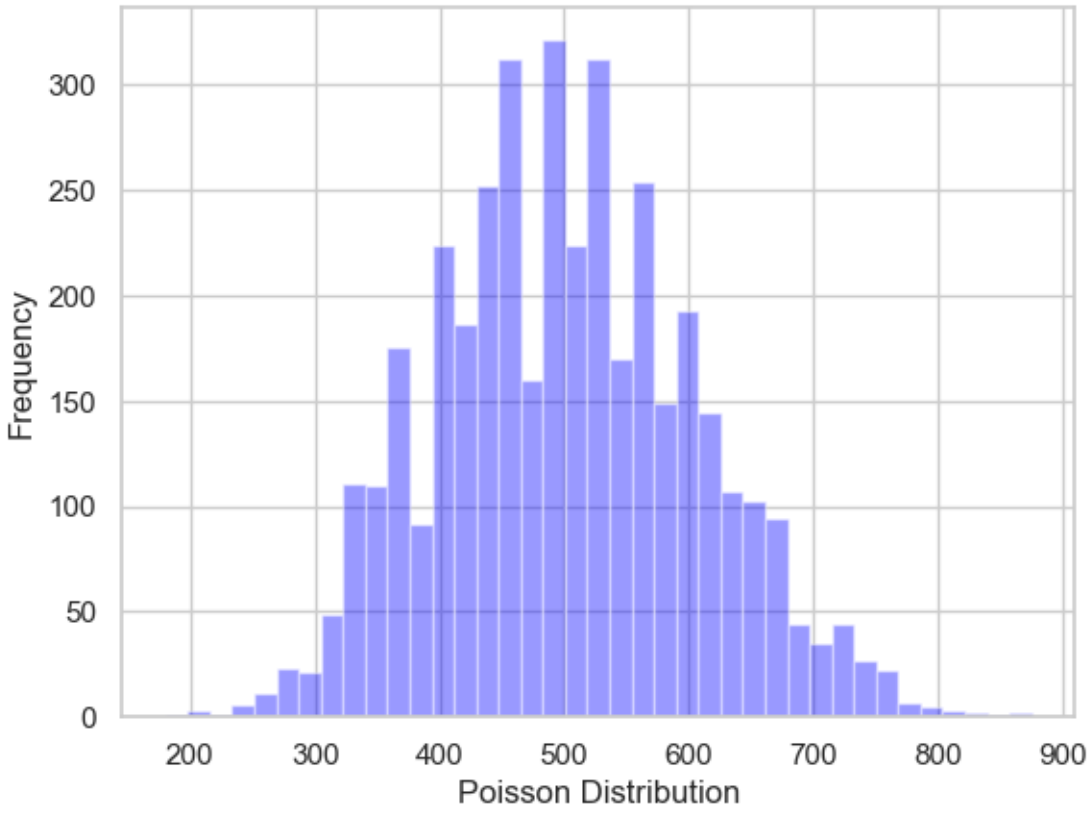
```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_English = sns.distplot(df_data['English'],
```



[131]:
```
# Poisson Distribution for English
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_English = sns.distplot(df_data['English'],
                  kde=False,
                  color='blue')
poisson_distribution_English.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\641263967.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

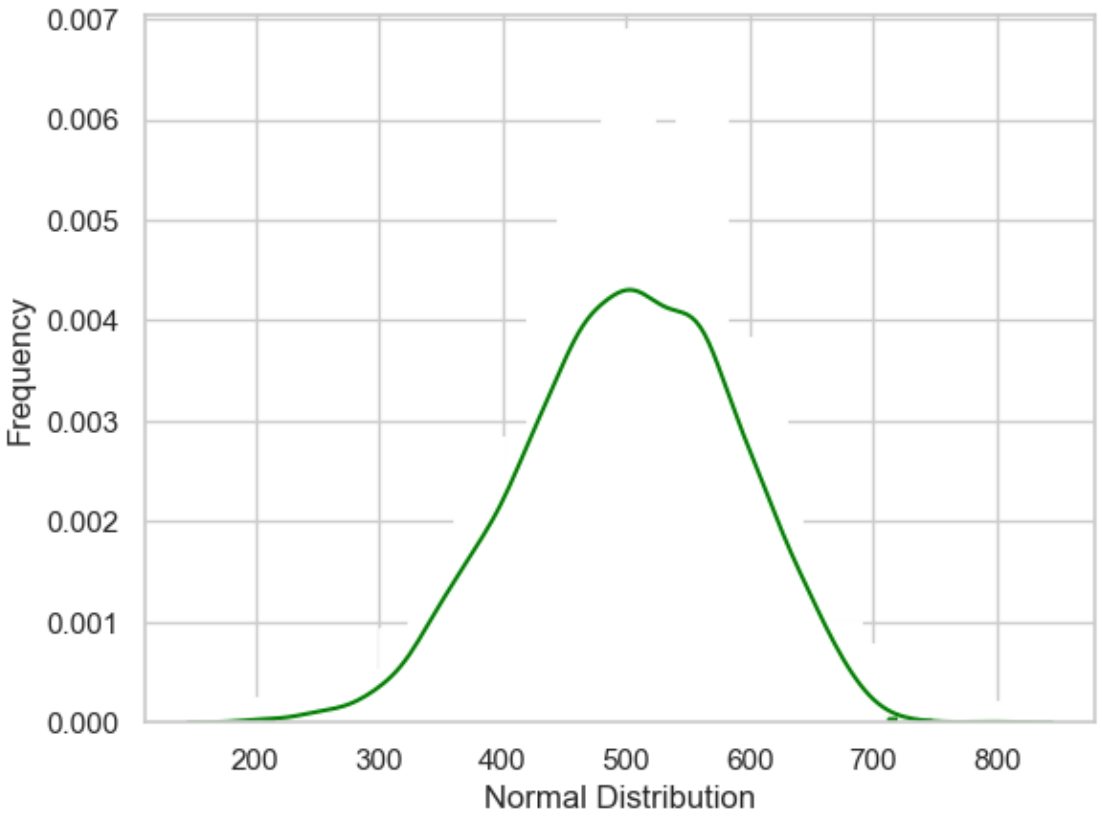
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_English = sns.distplot(df_data['English'],
```



[132]:
```
# Normal Distribution for Logical
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_Logical= sns.distplot(df_data['Logical'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_Logical.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
plt.show()
```

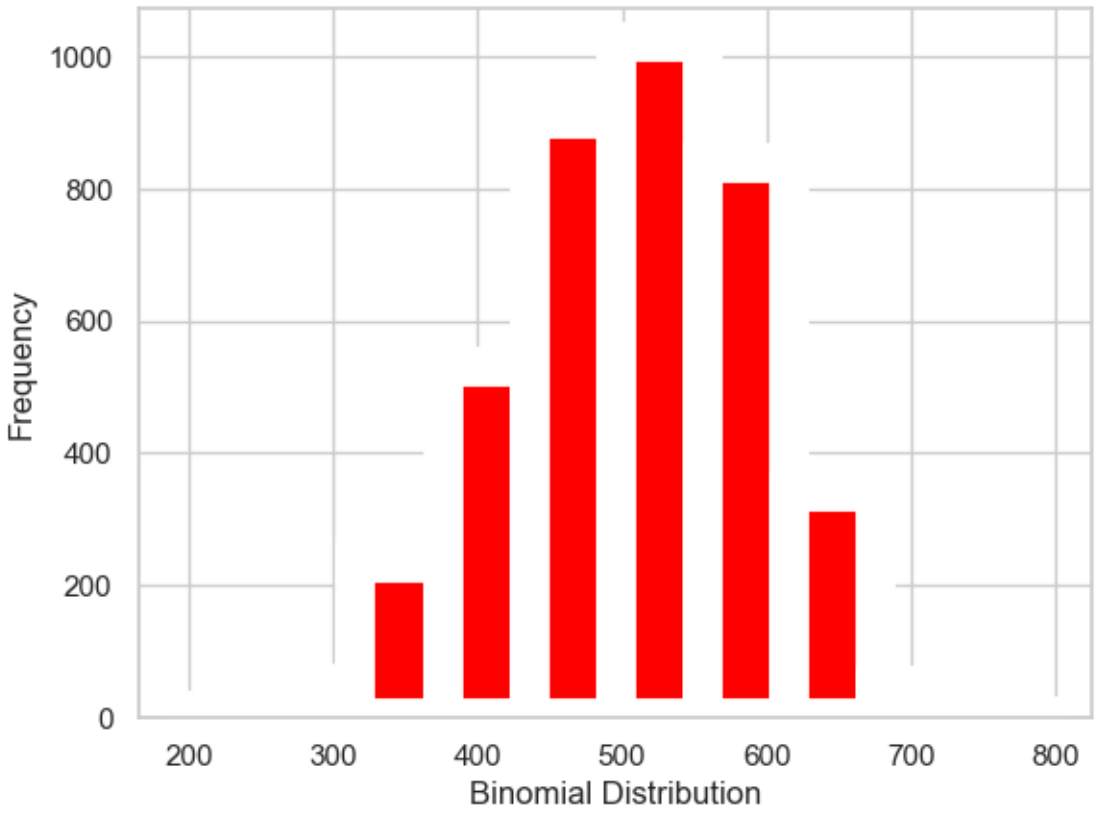
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\4290294683.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_Logical= sns.distplot(df_data['Logical'],
```



[133]:
```
# Binomial Distribution for Logical
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_Logical = sns.distplot(df_data['Logical'],
                  bins=10,
                  kde=False,
```

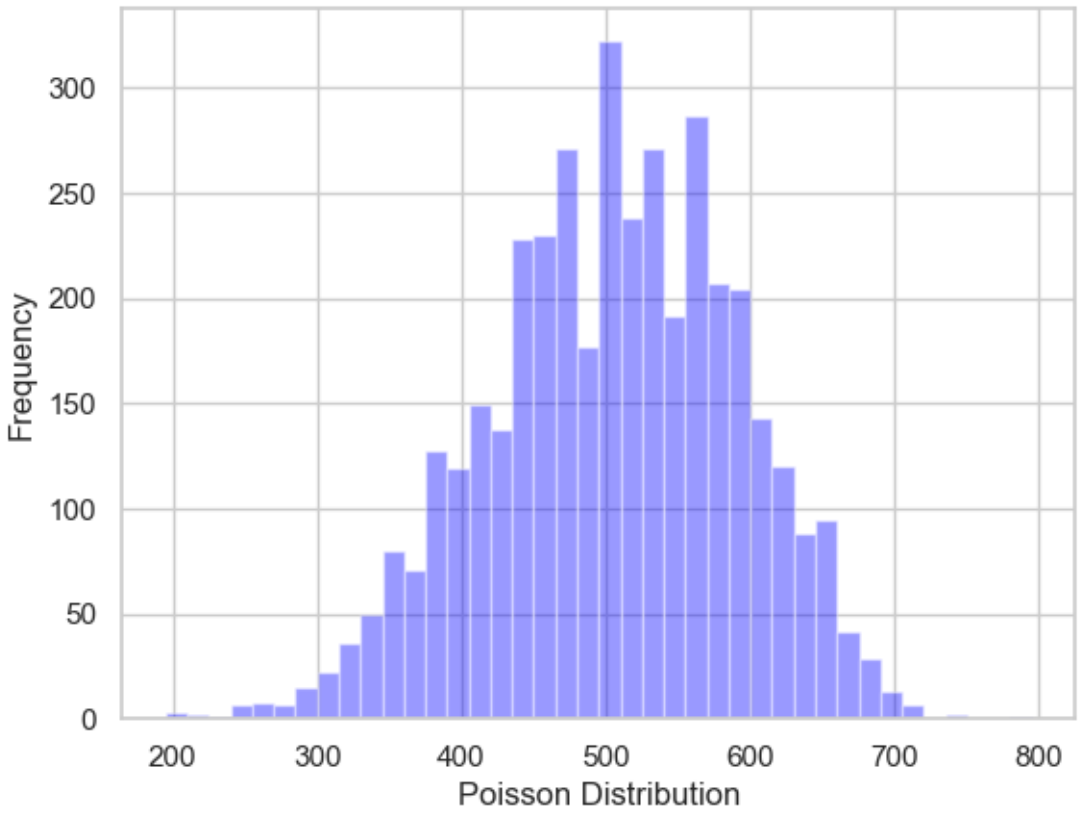
```
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_Logical.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3385089301.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_Logical = sns.distplot(df_data['Logical'],
```



[134]:
```
# Poisson Distribution for Logical

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_Logical = sns.distplot(df_data['Logical'],
                  kde=False,
                  color='blue')
poisson_distribution_Logical.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

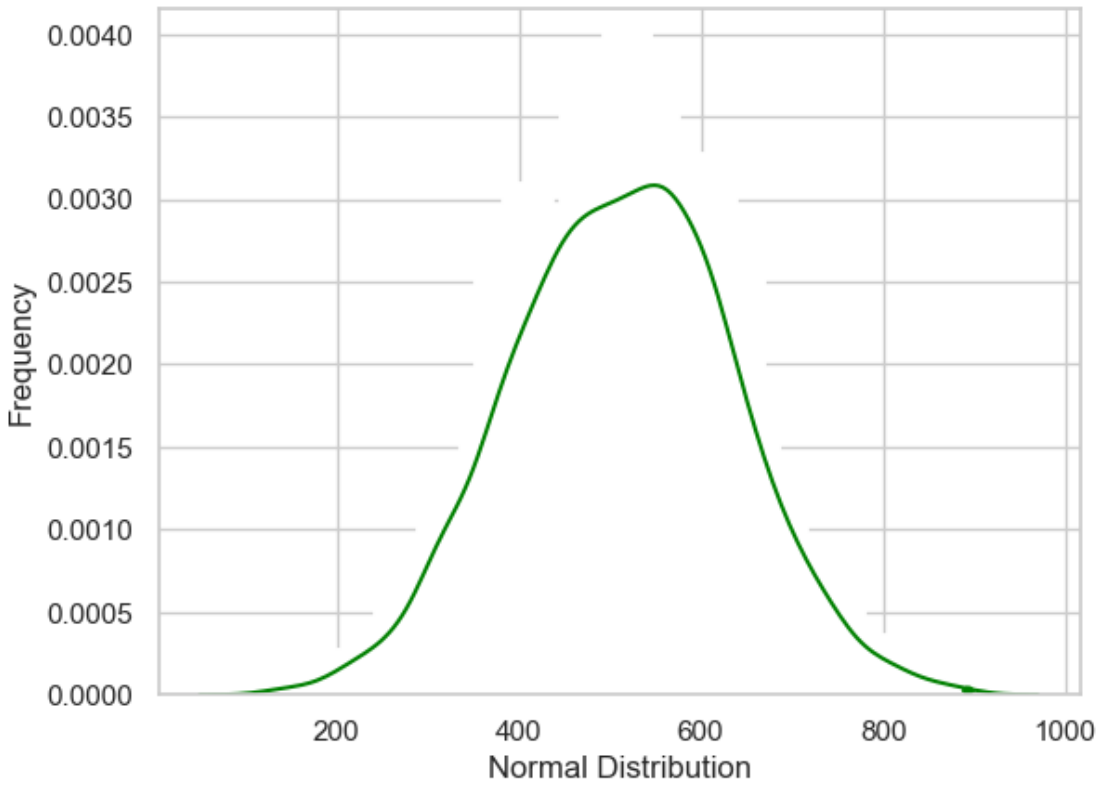
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2443655025.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_Logical = sns.distplot(df_data['Logical'],
```



[135]:
```
# Normal Distribution for Quant

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_Quant= sns.distplot(df_data['Quant'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_Quant.set(xlabel='Normal Distribution', ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\339486116.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

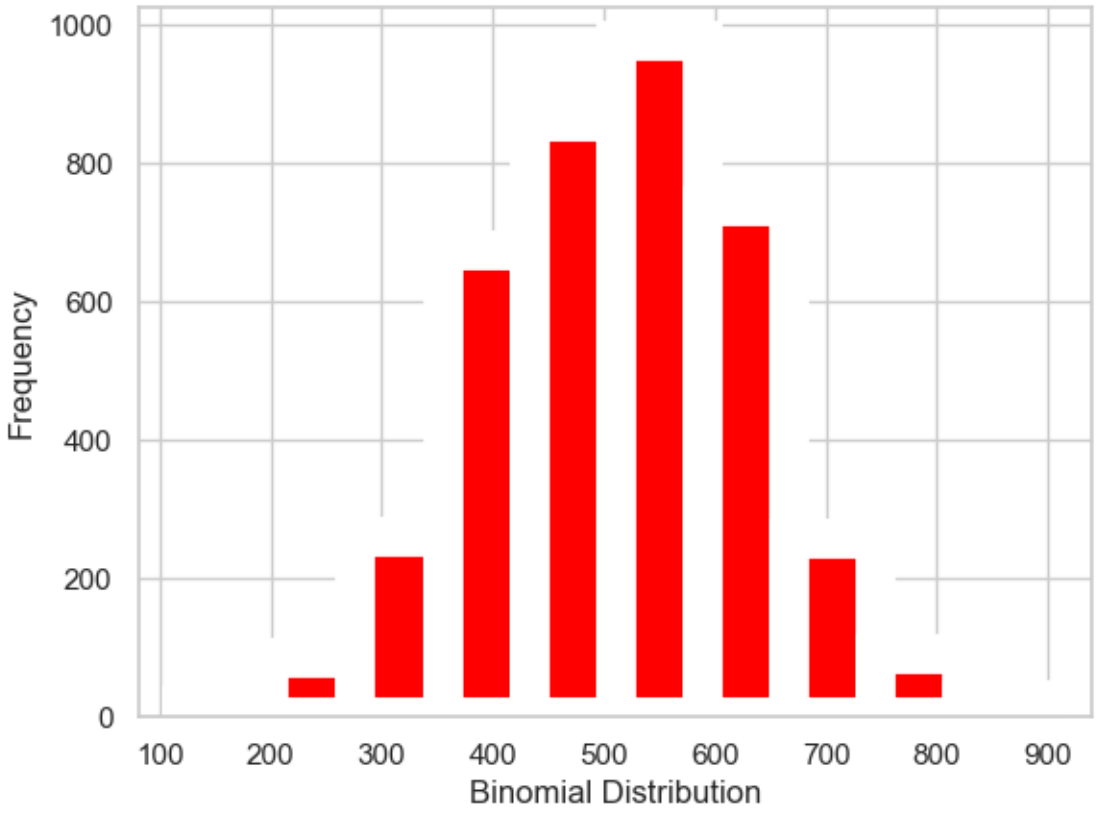
```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_Quant= sns.distplot(df_data['Quant'],
```



[136]:
```
# Binomial Distribution for Quant
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_Quant = sns.distplot(df_data['Quant'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_Quant.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\368089653.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

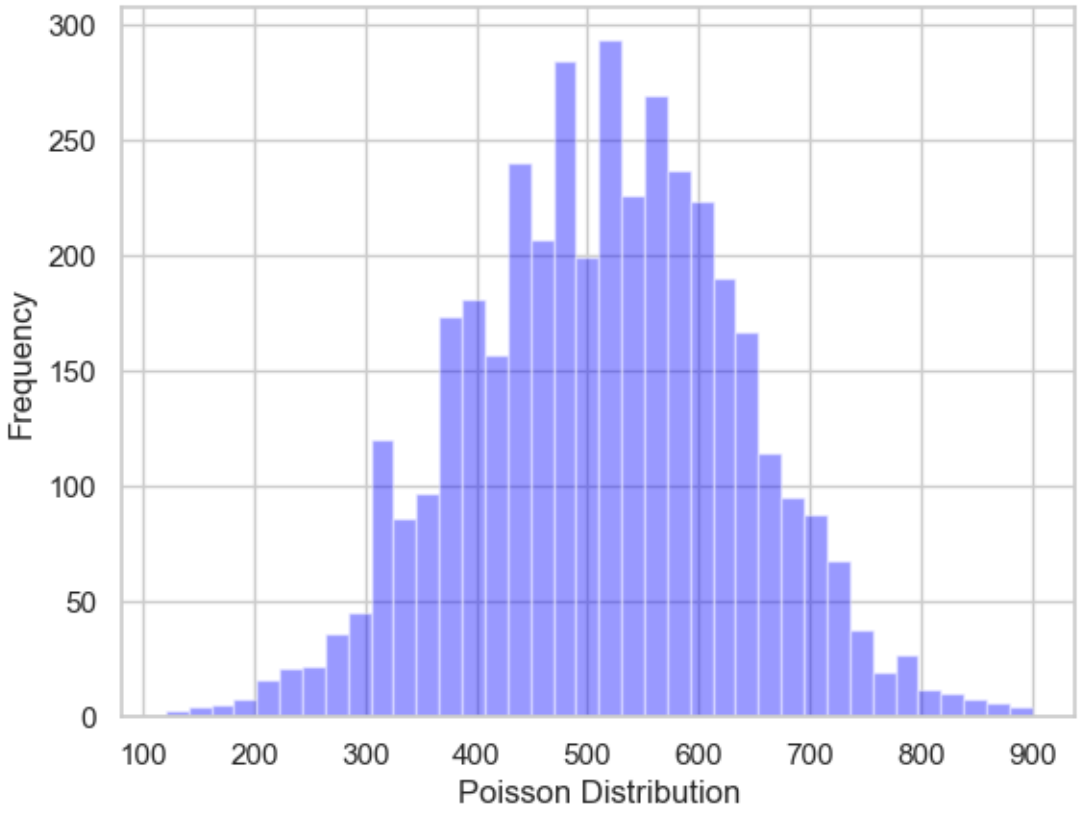
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_Quant = sns.distplot(df_data['Quant'],
```



[137]:
```
# Poisson Distribution for Quant

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_Quant = sns.distplot(df_data['Quant'],
                  kde=False,
                  color='blue')
poisson_distribution_Quant.set(xlabel='Poisson Distribution',␣
  ↪ylabel='Frequency')
```

```
plt.show()
```

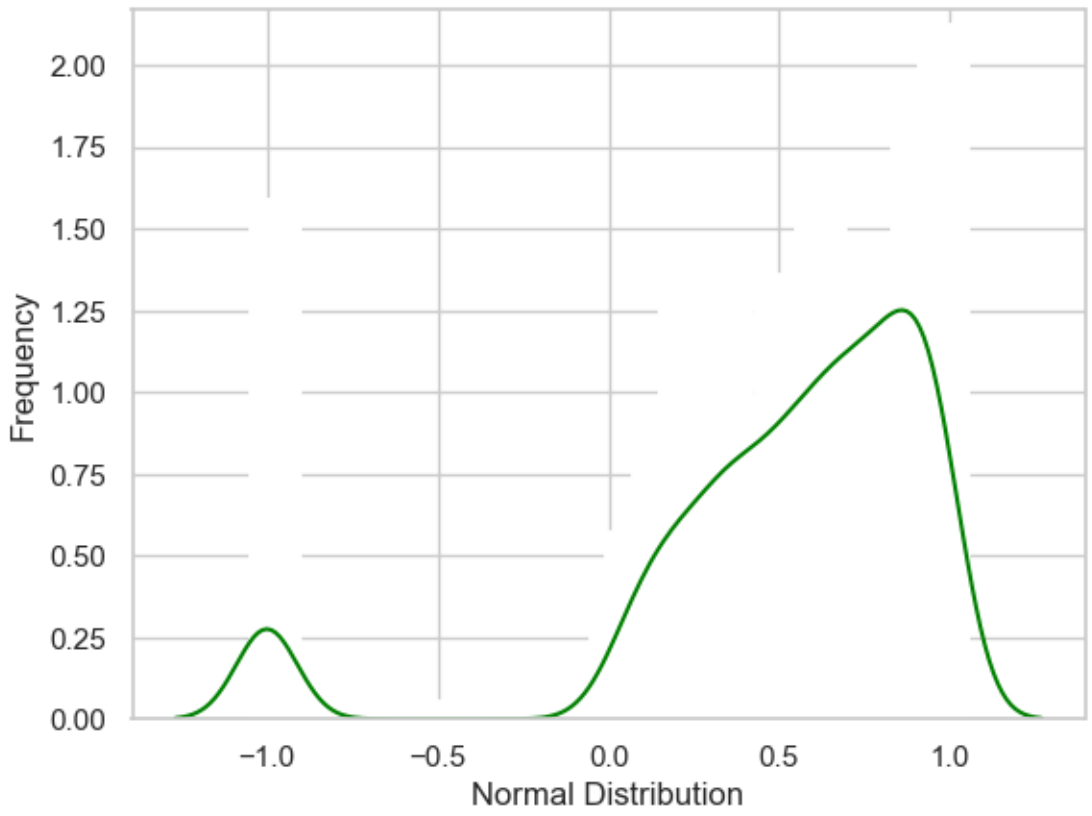
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3558832202.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_Quant = sns.distplot(df_data['Quant'],
```



[138]:
```
# Normal Distribution for Domain

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_Domain= sns.distplot(df_data['Domain'],
                 bins=50,
```

```
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_Domain.set(xlabel='Normal Distribution', ylabel='Frequency')

plt.show()
```

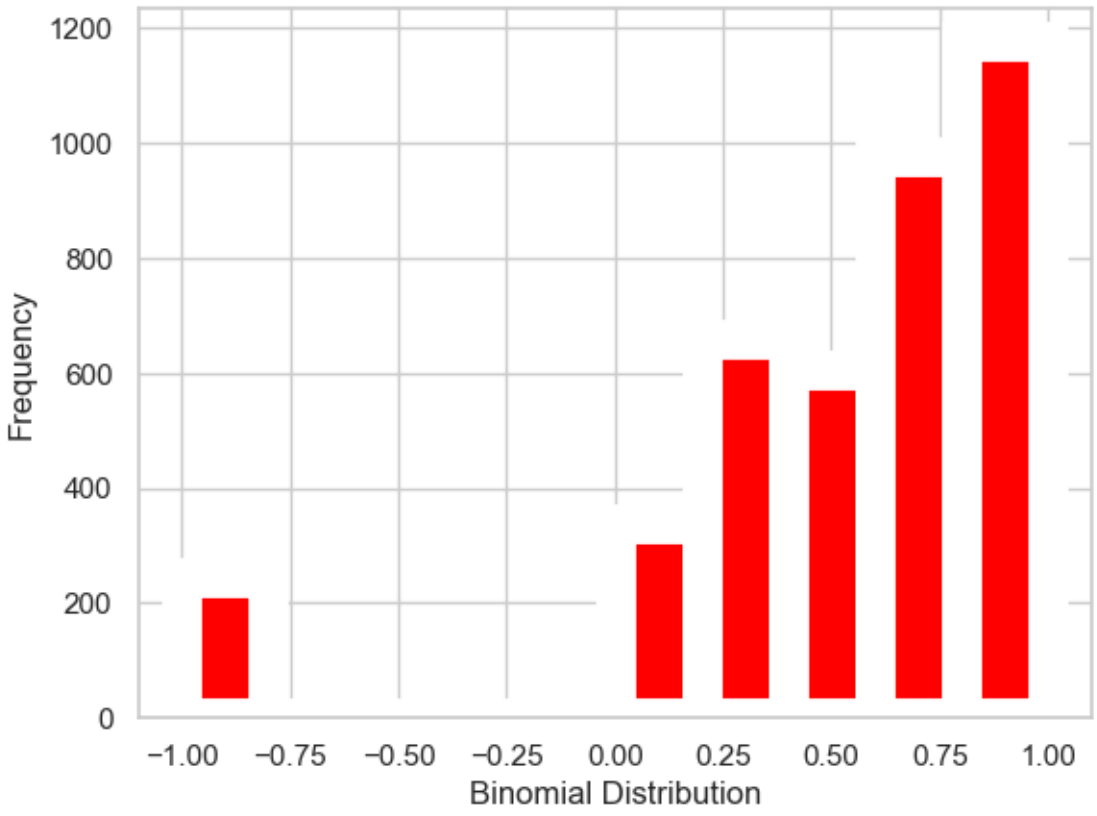
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3500999162.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_Domain= sns.distplot(df_data['Domain'],
```



[139]:
```
# Binomial Distribution for Domain
import seaborn as sns
```

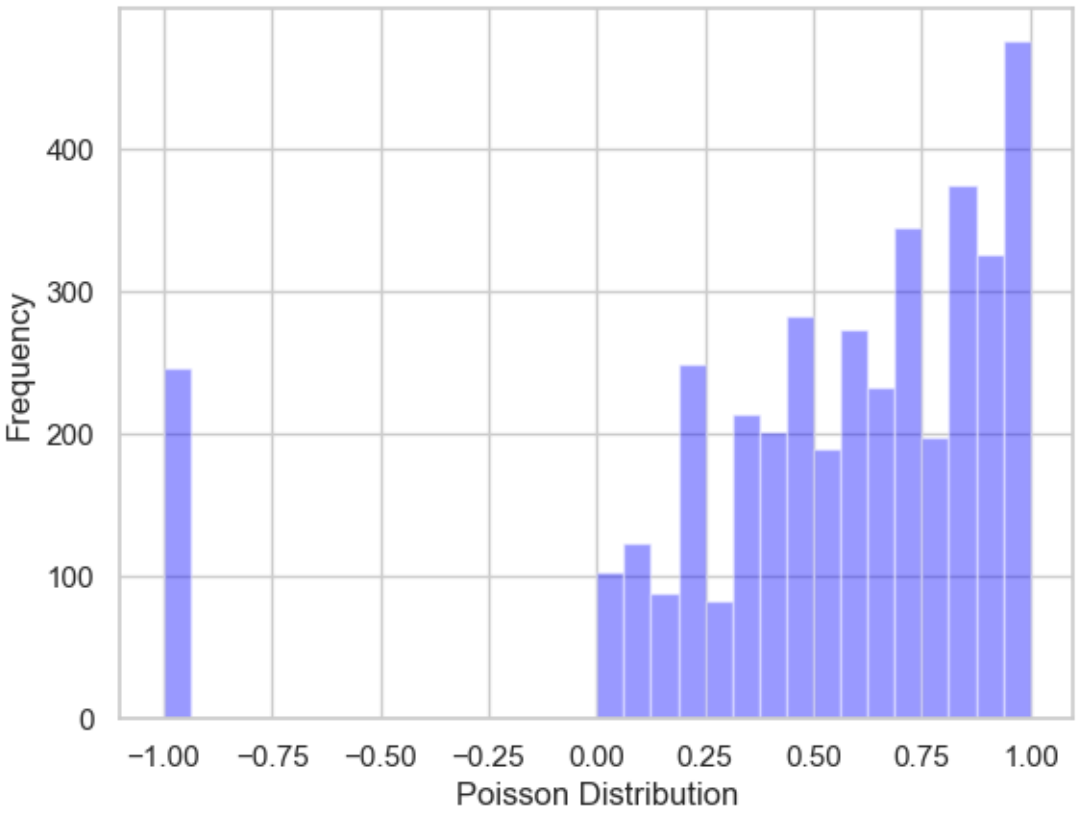
```
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_Domain = sns.distplot(df_data['Domain'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_Domain.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2590080586.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_Domain = sns.distplot(df_data['Domain'],
```



[140]:
```
# Poisson Distribution for Domain

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_Domain = sns.distplot(df_data['Domain'],
                  kde=False,
                  color='blue')
poisson_distribution_Domain.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

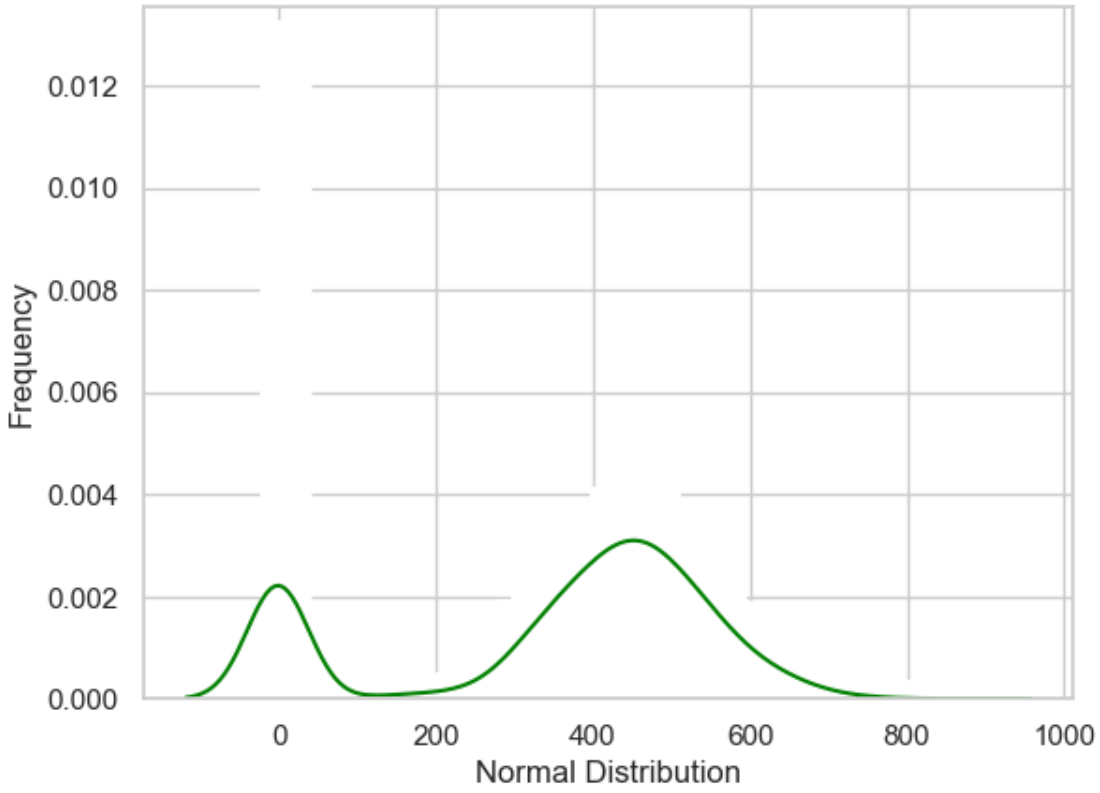
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2994031512.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_Domain = sns.distplot(df_data['Domain'],
```



[141]:
```
# Normal Distribution for Computer Programming

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_ComputerProgramming= sns.
 ↪distplot(df_data['ComputerProgramming'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_ComputerProgramming.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3002384001.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

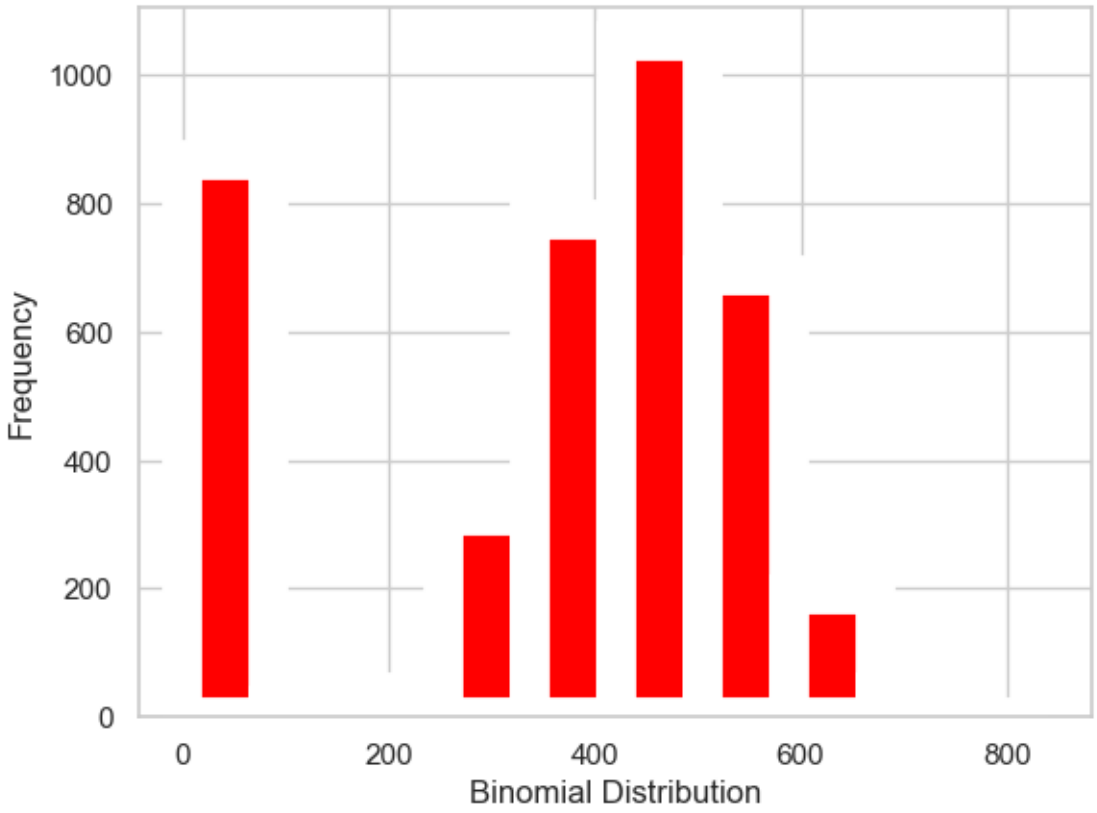
```
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_ComputerProgramming=
sns.distplot(df_data['ComputerProgramming'],
```



[142]:
```
# Binomial Distribution for Computer Programming

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_ComputerProgramming = sns.
 ↪distplot(df_data['ComputerProgramming'],
                 bins=10,
                 kde=False,
                 color='red',
                 hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_ComputerProgramming.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')
```

```
plt.show()
```

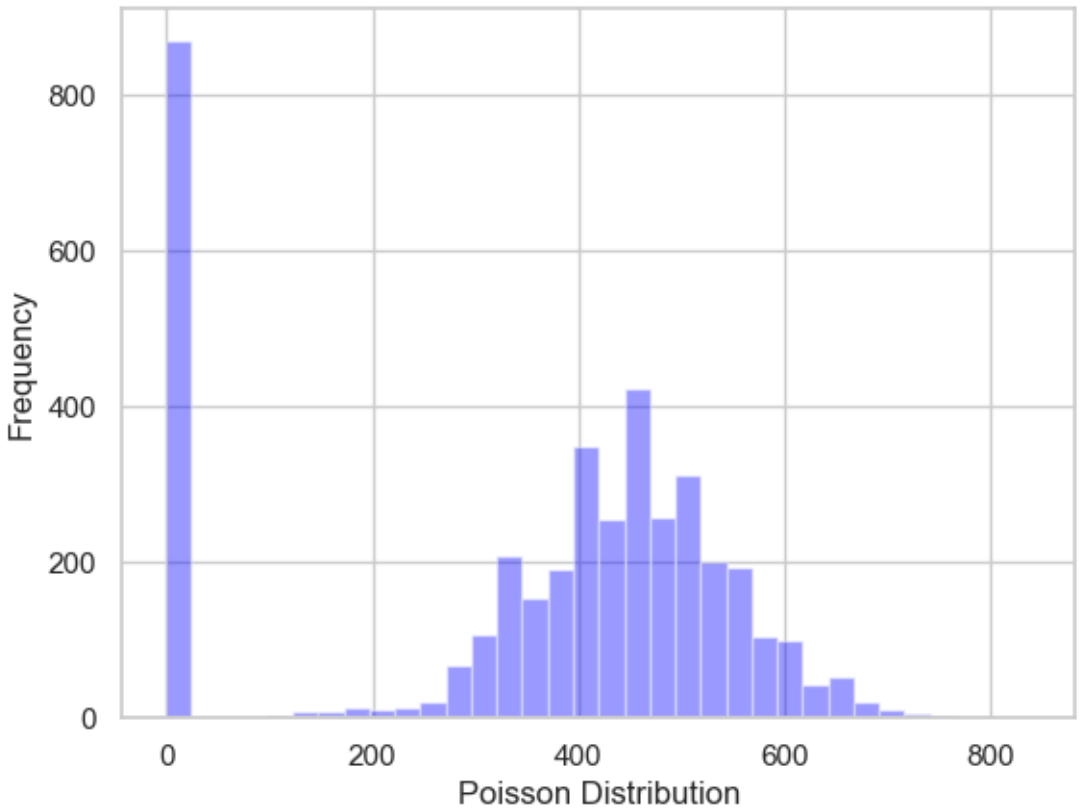
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\876706802.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_ComputerProgramming =
sns.distplot(df_data['ComputerProgramming'],
```



[143]:
```
# Poisson Distribution for Computer Programming

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

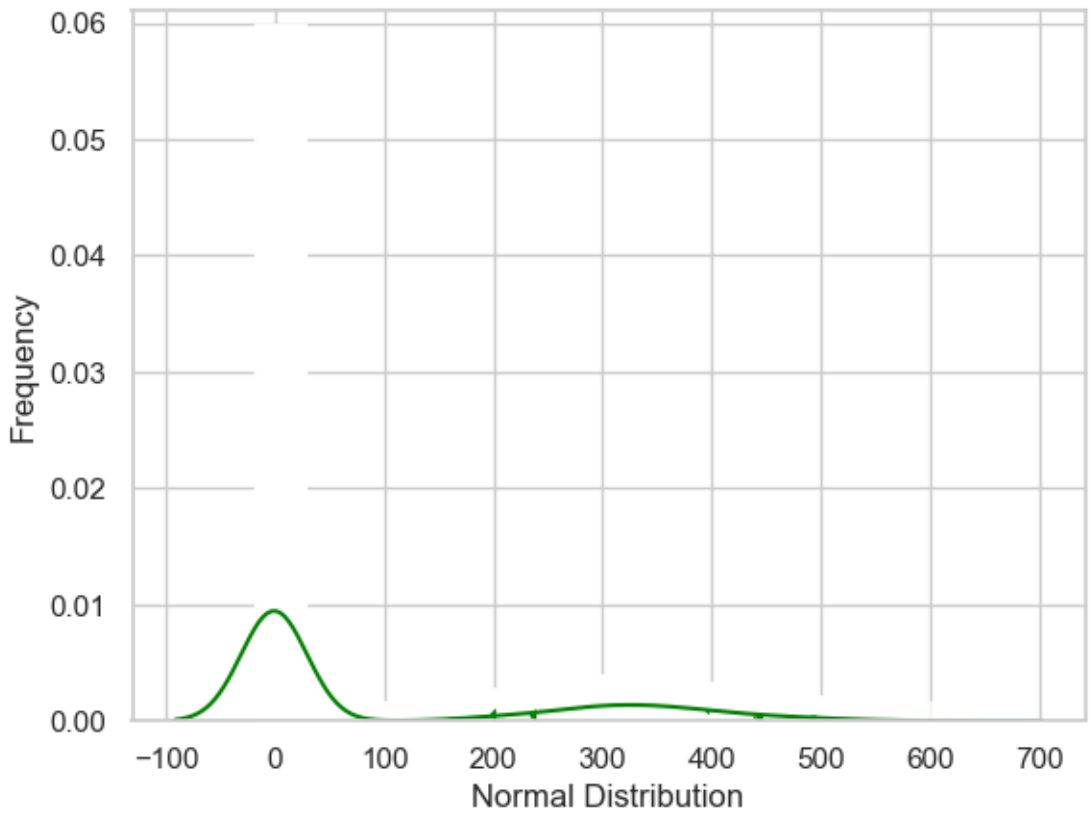
```
poisson_distribution_ComputerProgramming = sns.
 ↪distplot(df_data['ComputerProgramming'],
                  kde=False,
                  color='blue')
poisson_distribution_ComputerProgramming.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1550334184.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_ComputerProgramming =
sns.distplot(df_data['ComputerProgramming'],
```



[144]:
```
# Normal Distribution for Electronics and Semiconductors
import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_ElectronicsAndSemicon= sns.
 ↪distplot(df_data['ElectronicsAndSemicon'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_ElectronicsAndSemicon.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

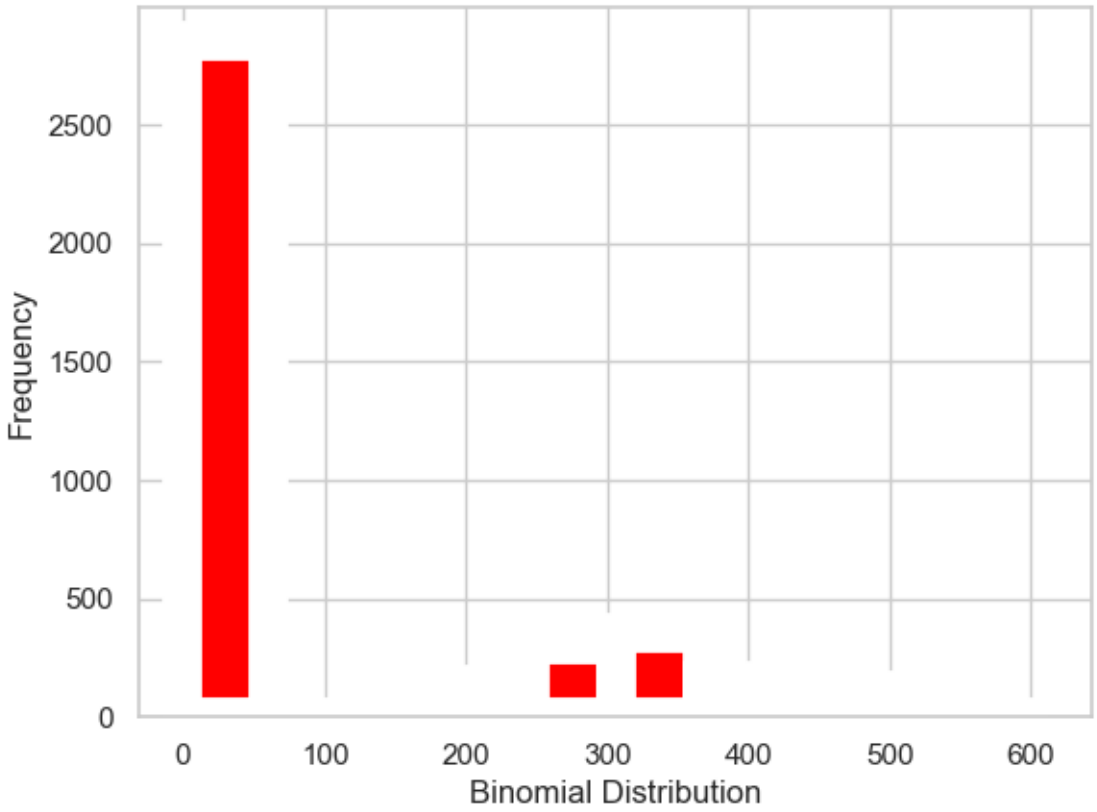
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1976769817.py:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_ElectronicsAndSemicon=
sns.distplot(df_data['ElectronicsAndSemicon'],
```



[145]:
```
# Binomial Distribution for Electronics and Semiconductors

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_ElectronicsAndSemicon = sns.
 ↪distplot(df_data['ElectronicsAndSemicon'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_ElectronicsAndSemicon.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\140925955.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

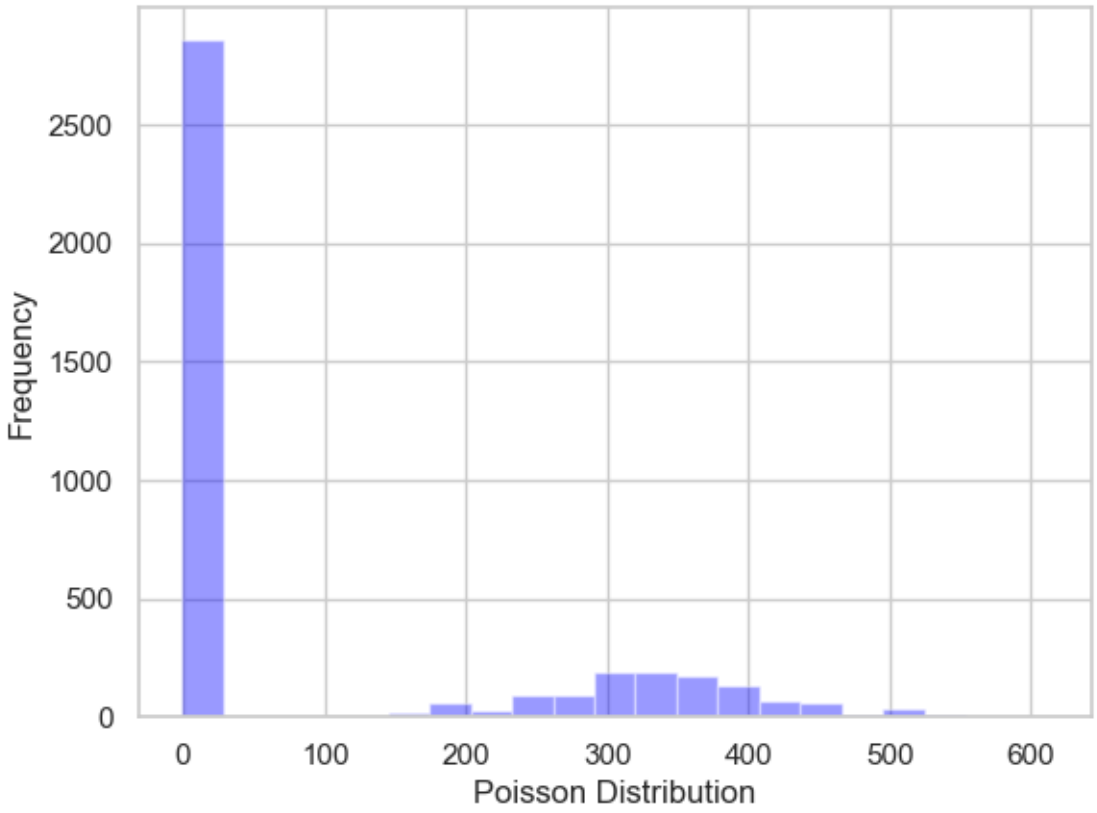
```
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_ElectronicsAndSemicon =
sns.distplot(df_data['ElectronicsAndSemicon'],
```



[146]:
```
# Poisson Distribution for Electronics and Semiconductors

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_ElectronicsAndSemicon = sns.
  ↪distplot(df_data['ElectronicsAndSemicon'],
                  kde=False,
                  color='blue')
poisson_distribution_ElectronicsAndSemicon.set(xlabel='Poisson Distribution',␣
  ↪ylabel='Frequency')
```

```
plt.show()
```

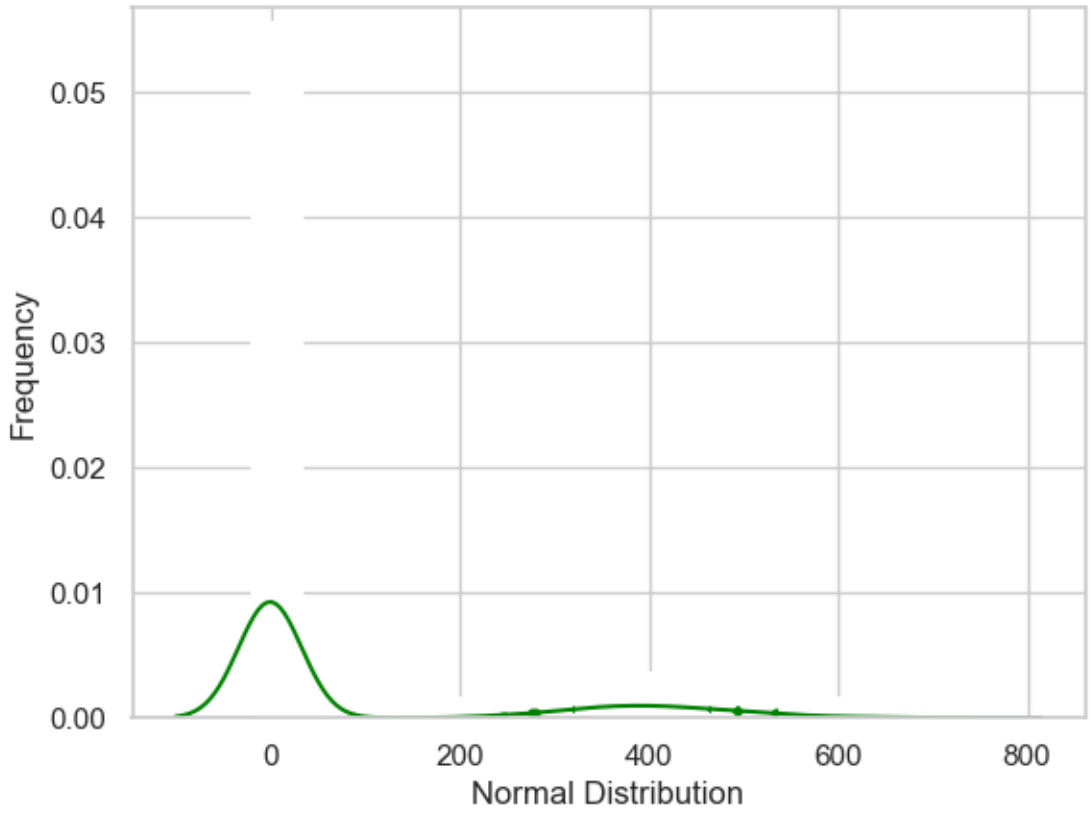
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3408202504.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_ElectronicsAndSemicon =
sns.distplot(df_data['ElectronicsAndSemicon'],
```



[147]:
```
# Normal Distribution for Computer Science

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_ComputerScience= sns.distplot(df_data['ComputerScience'],
```

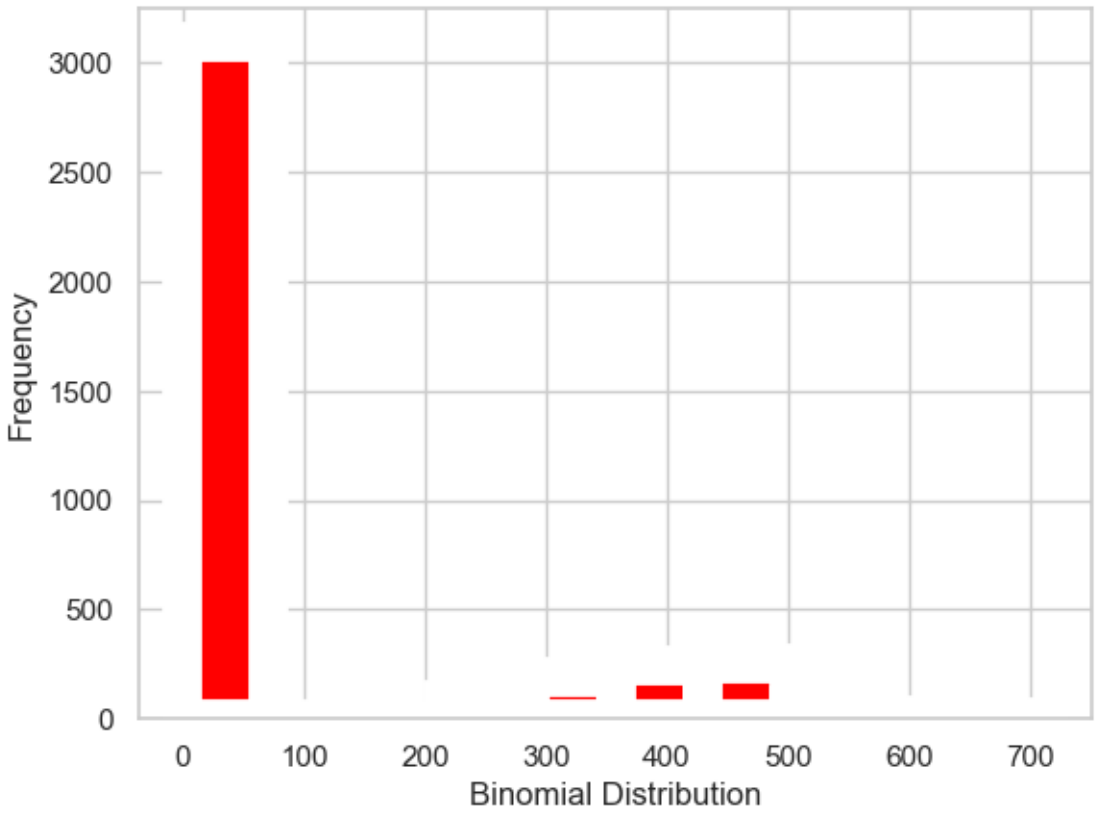
```
                   bins=50,
                   kde=True,
                   color='green',
                   hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_ComputerScience.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1185852212.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_ComputerScience= sns.distplot(df_data['ComputerScience'],
```



[148]:
```
# Binomial Distribution for Computer Science

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_ComputerScience = sns.distplot(df_data['ComputerScience'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_ComputerScience.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

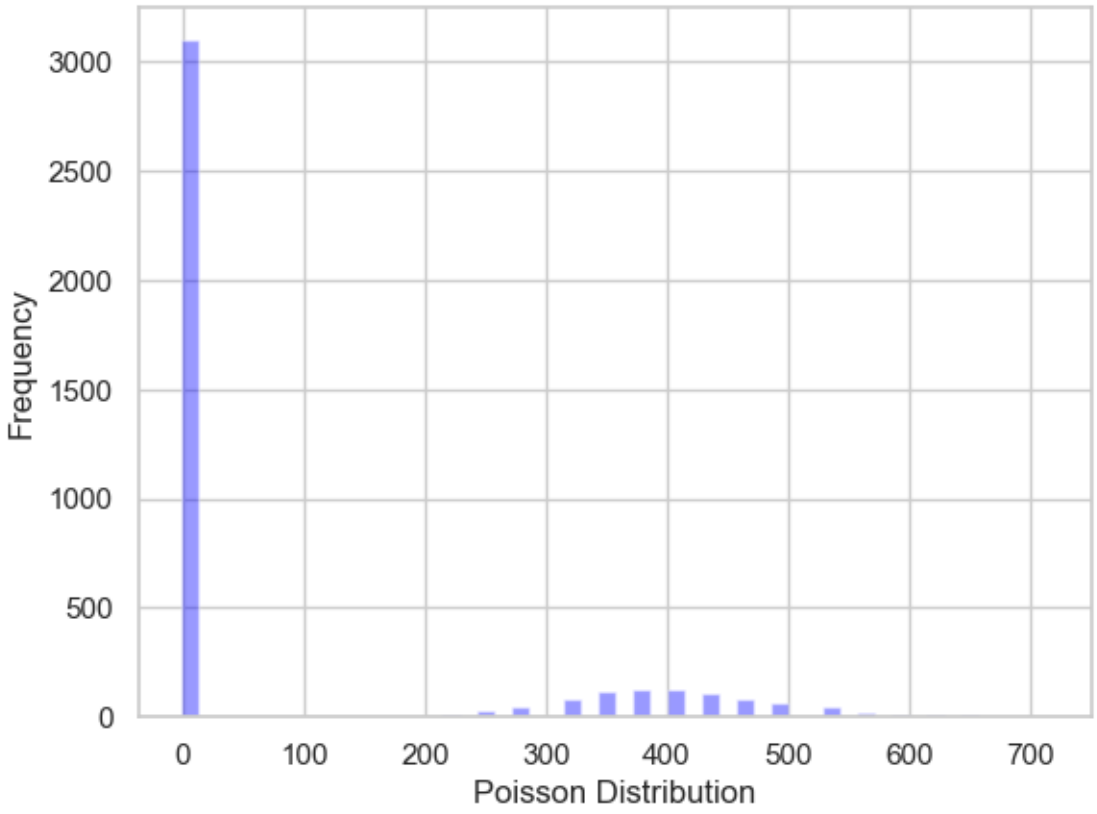
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1154003975.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_ComputerScience =
sns.distplot(df_data['ComputerScience'],
```



[149]:
```
# Poisson Distribution for Computer Science

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_ComputerScience = sns.distplot(df_data['ComputerScience'],
                  kde=False,
                  color='blue')
poisson_distribution_ComputerScience.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

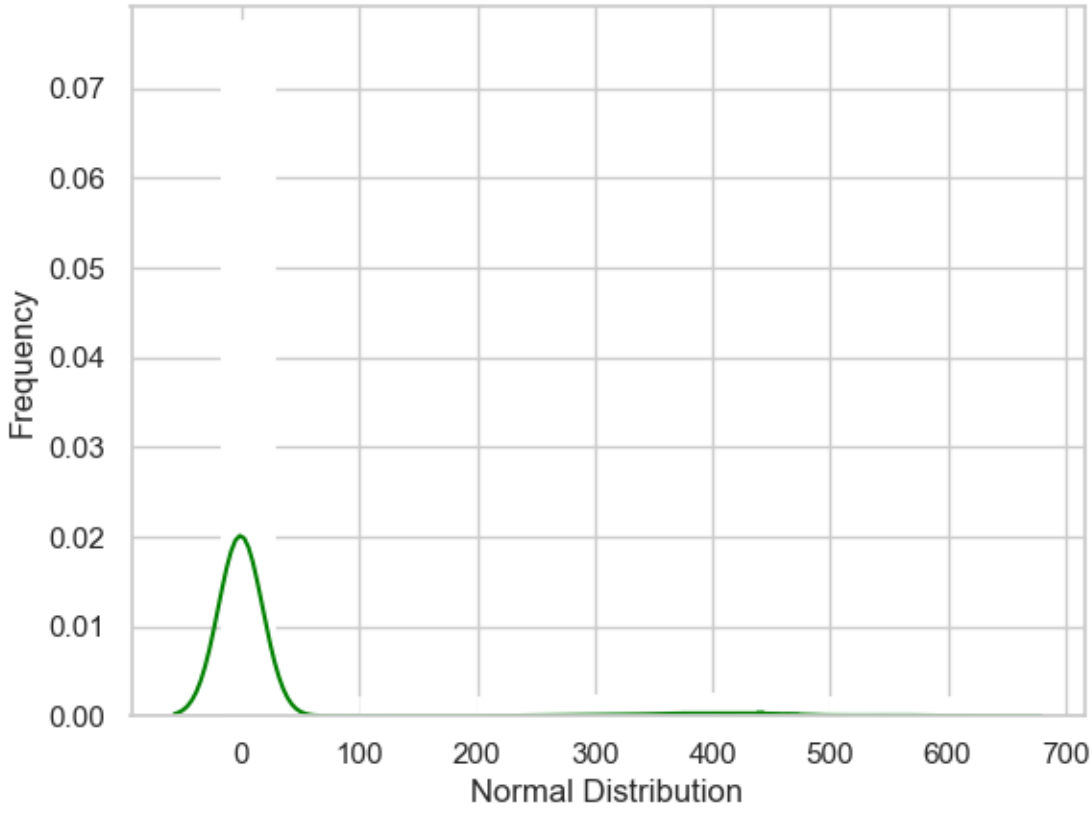
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3096386443.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_ComputerScience =
sns.distplot(df_data['ComputerScience'],
```



[150]:
```
# Normal Distribution for Mechanical Engineering

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_MechanicalEngg= sns.distplot(df_data['MechanicalEngg'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_MechanicalEngg.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

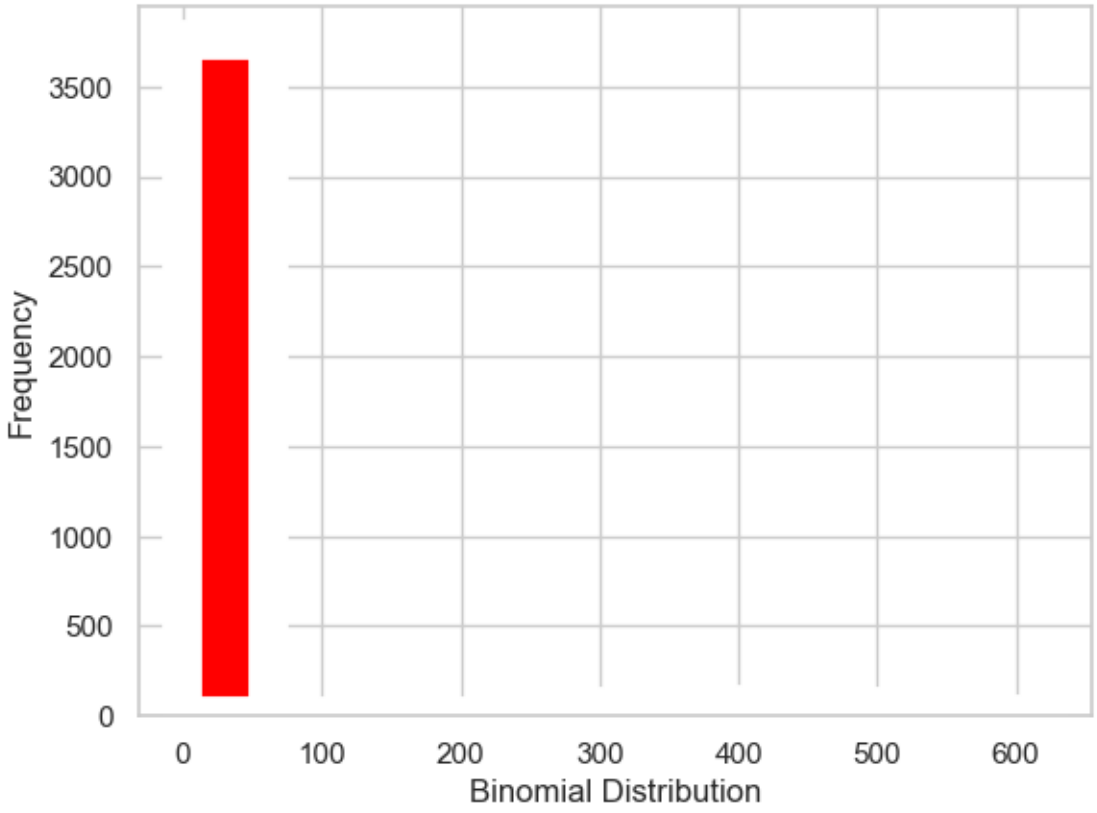
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2522257161.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_MechanicalEngg= sns.distplot(df_data['MechanicalEngg'],
```



[151]:
```
# Binomial Distribution for Mechanical Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_MechanicalEngg = sns.distplot(df_data['MechanicalEngg'],
                  bins=10,
                  kde=False,
                  color='red',
```

```
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_MechanicalEngg.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

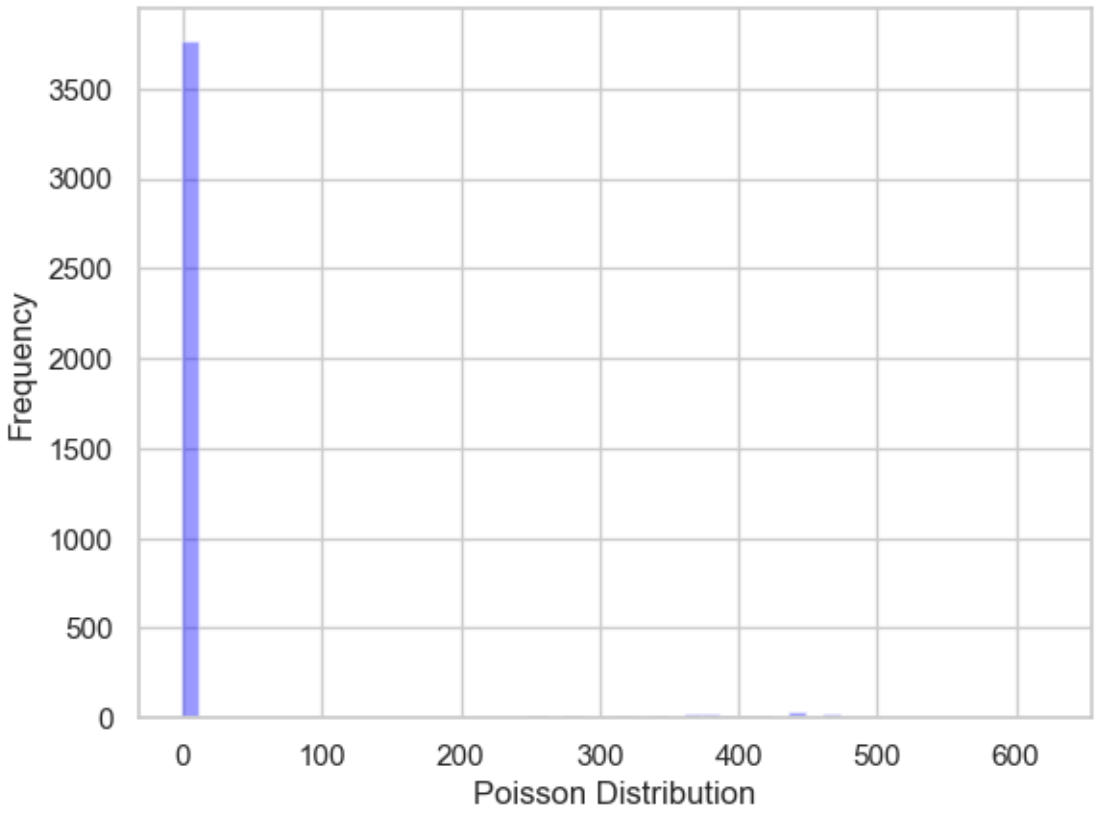
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1733551930.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_MechanicalEngg = sns.distplot(df_data['MechanicalEngg'],
```



[152]:
```
# Poisson Distribution for Mechanical Engineering

import seaborn as sns
```

```
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_MechanicalEngg = sns.distplot(df_data['MechanicalEngg'],
                  kde=False,
                  color='blue')
poisson_distribution_MechanicalEngg.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

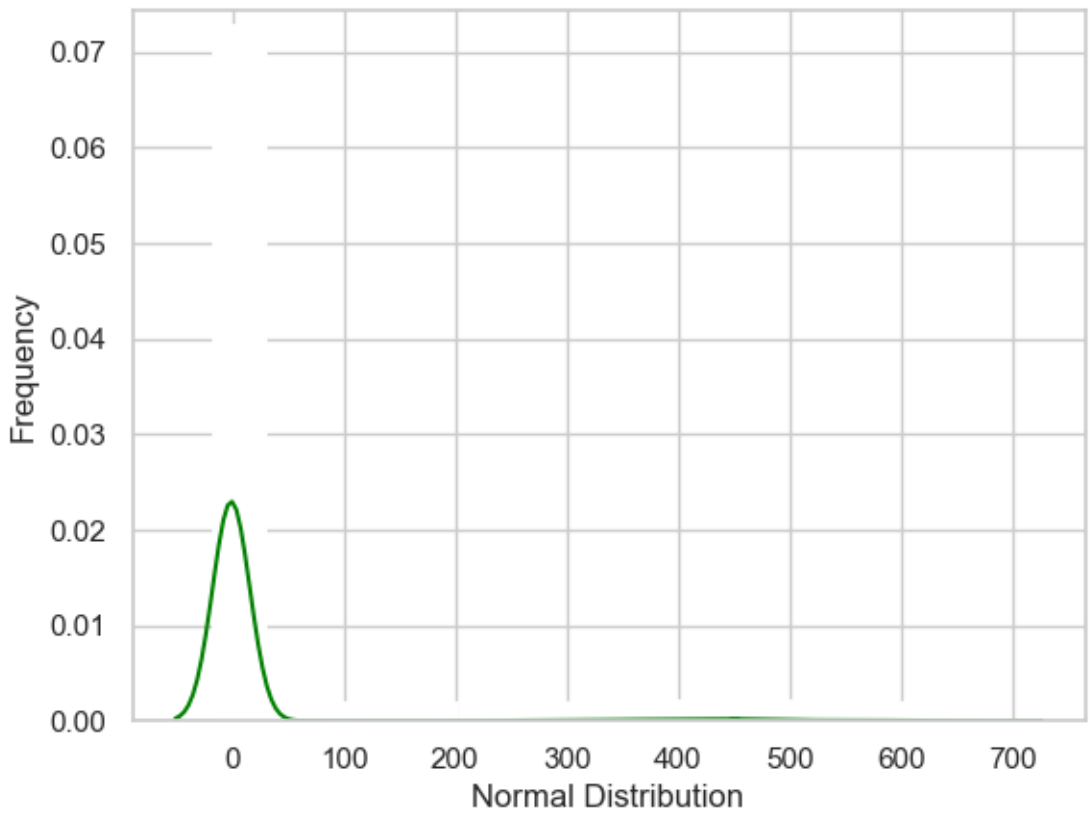
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2218289821.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_MechanicalEngg = sns.distplot(df_data['MechanicalEngg'],
```



[153]:
```
# Normal Distribution for Electrical Engineering

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_ElectricalEngg= sns.distplot(df_data['ElectricalEngg'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_ElectricalEngg.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1884899169.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_ElectricalEngg= sns.distplot(df_data['ElectricalEngg'],
```

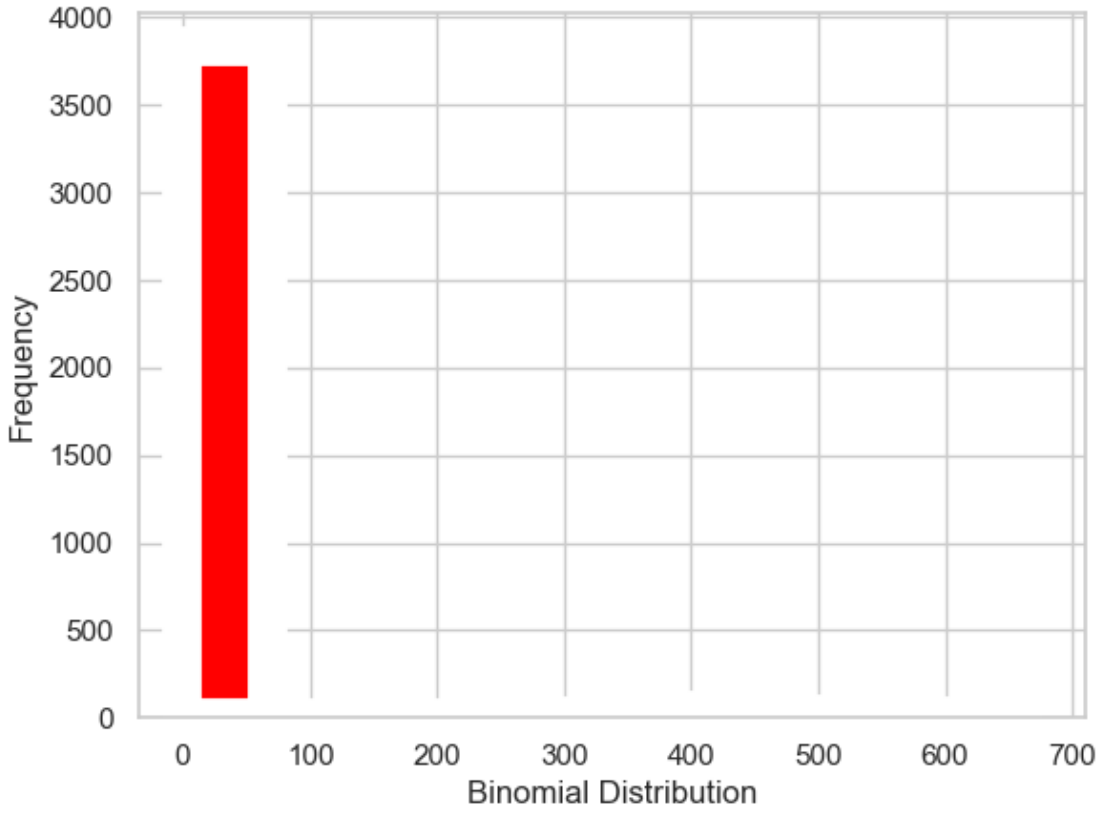
[153]:
```
[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]
```



[154]:
```
# Binomial Distribution for Electrical Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_ElectricalEngg = sns.distplot(df_data['ElectricalEngg'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_ElectricalEngg.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\4188465938.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

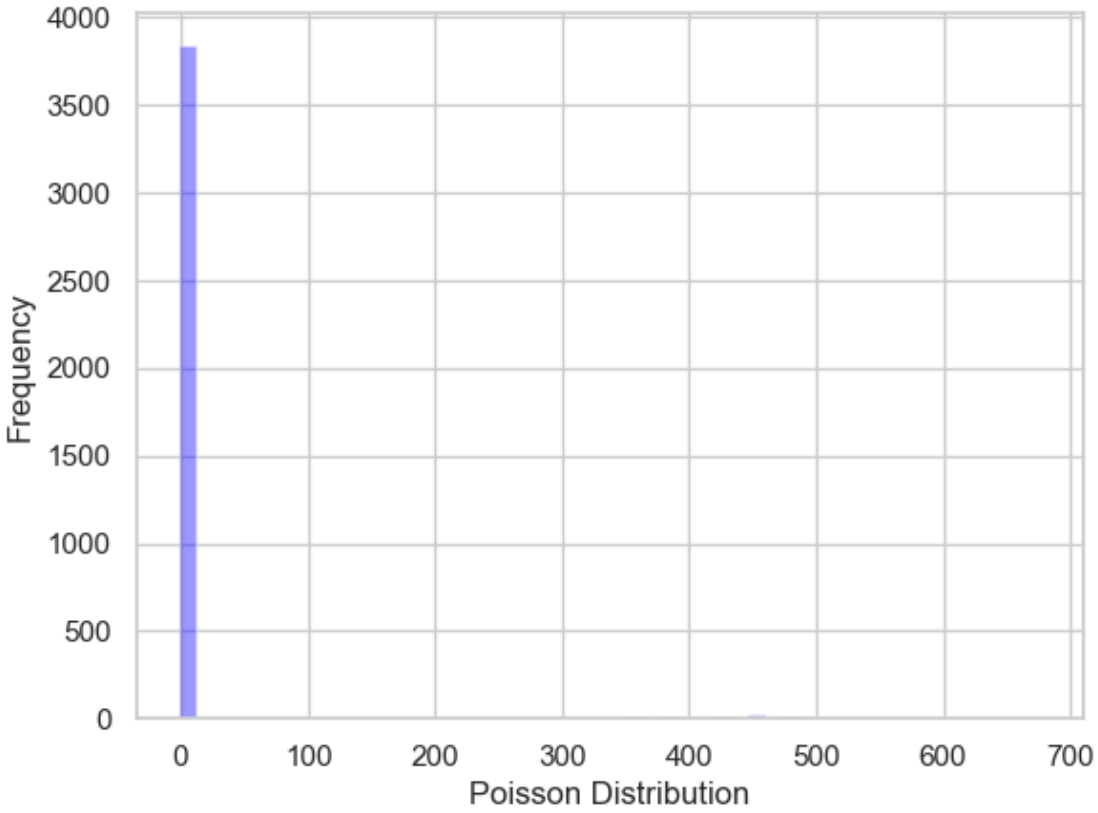
```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_ElectricalEngg = sns.distplot(df_data['ElectricalEngg'],
```



[155]:
```
# Poisson Distribution for Electrical Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_ElectricalEngg = sns.distplot(df_data['ElectricalEngg'],
                  kde=False,
                  color='blue')
poisson_distribution_ElectricalEngg.set(xlabel='Poisson Distribution',␣
  ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\340749053.py:7: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_ElectricalEngg = sns.distplot(df_data['ElectricalEngg'],
```

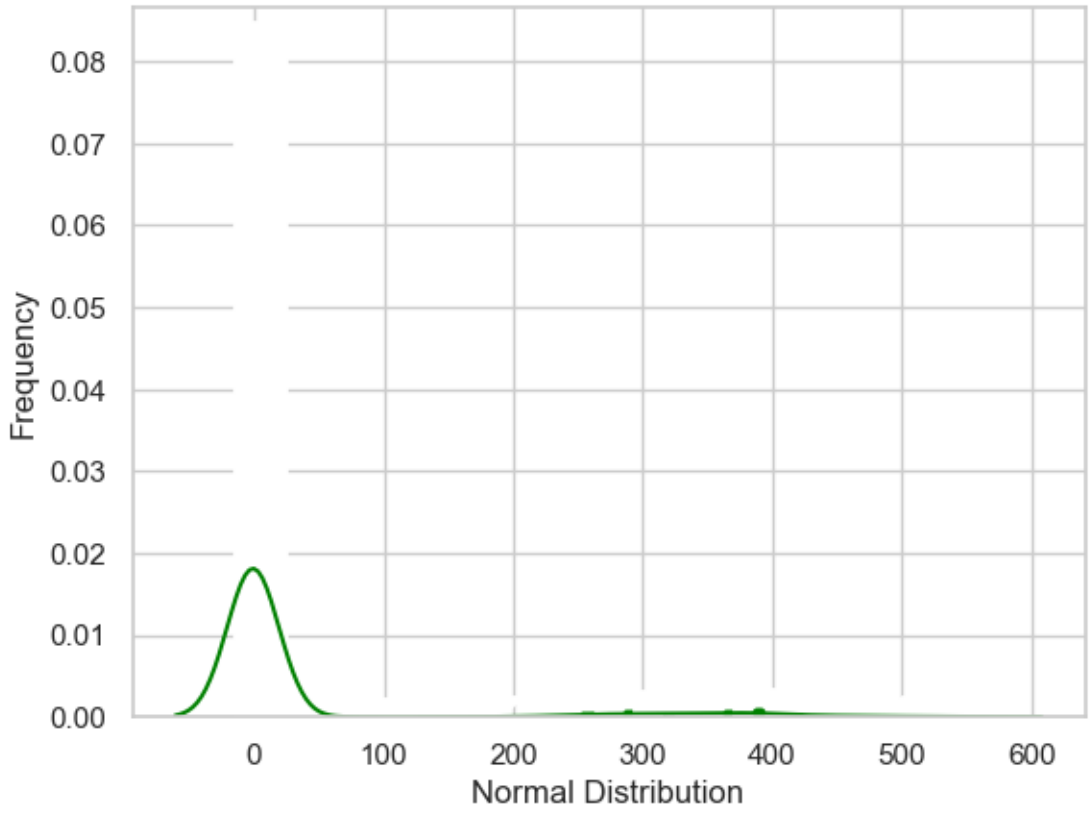


[156]:
```
# Normal Distribution for Telecom Engineering

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_TelecomEngg= sns.distplot(df_data['TelecomEngg'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
```

```
normal_distribution_TelecomEngg.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2560243128.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_TelecomEngg= sns.distplot(df_data['TelecomEngg'],
```

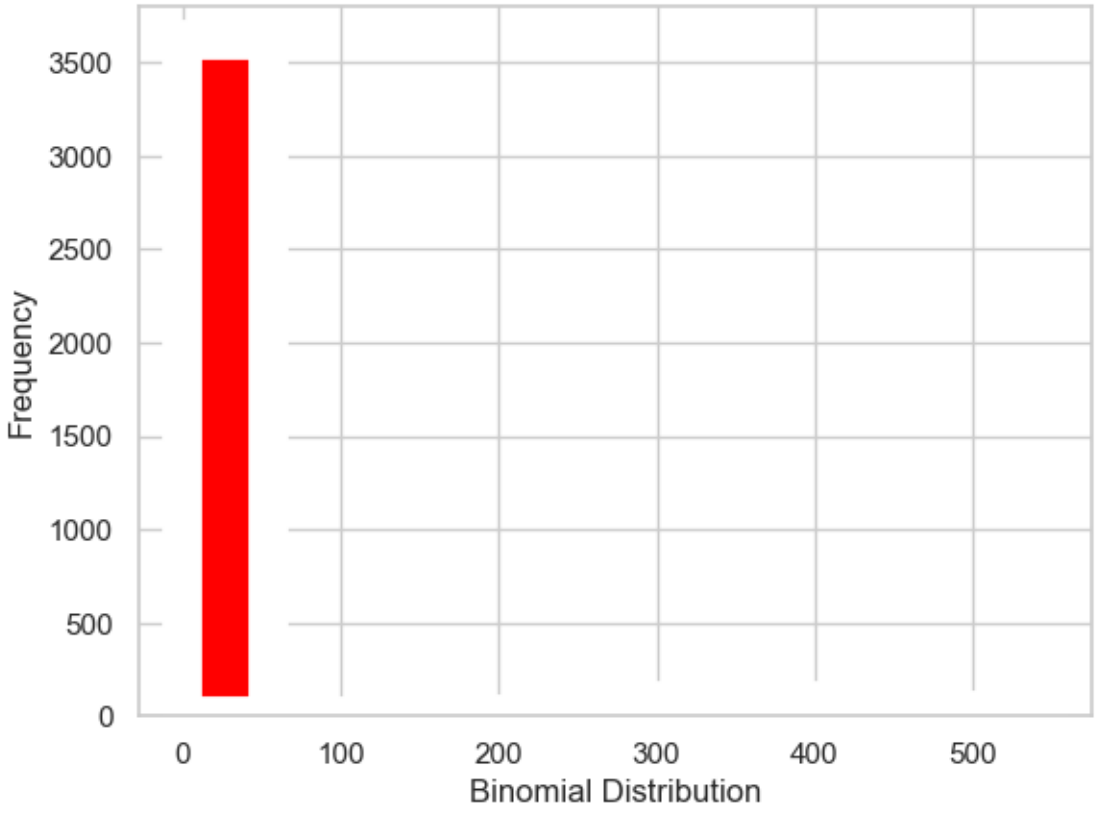
[156]:
```
[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]
```



[157]:
```
# Binomial Distribution for Telecom Engineering

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

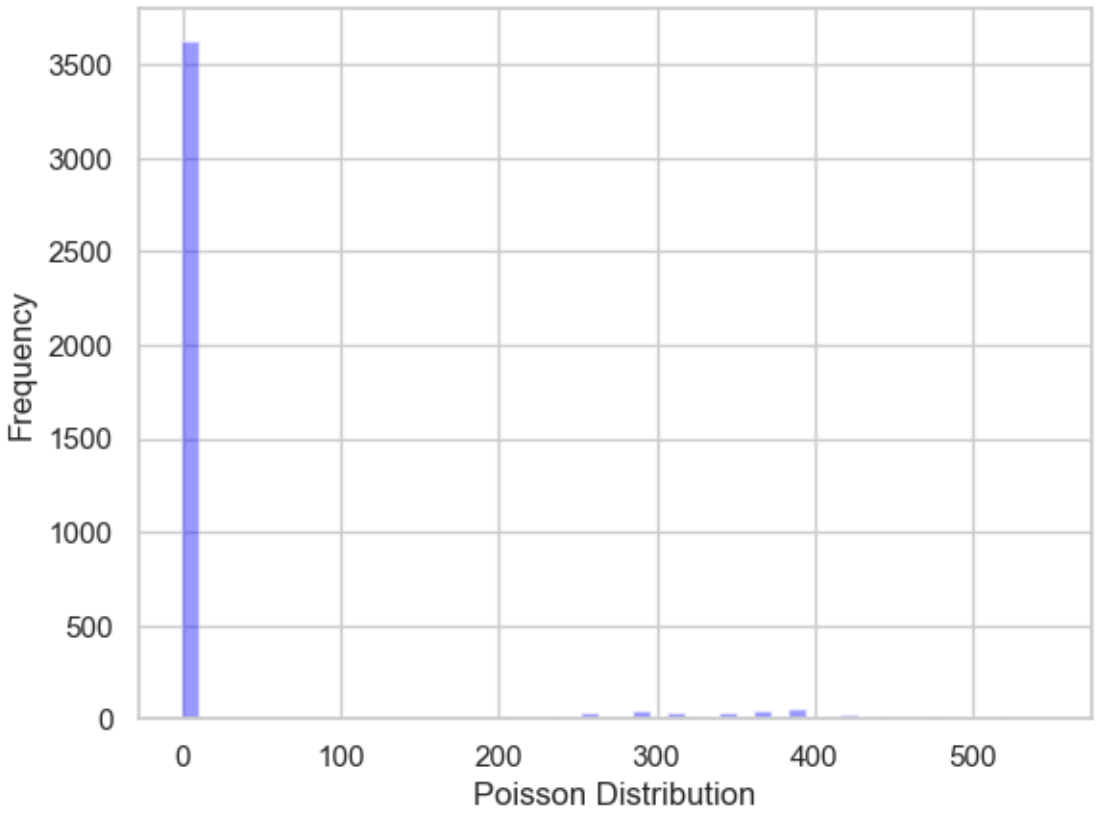
```
import numpy as np
binomial_distribution_TelecomEngg = sns.distplot(df_data['TelecomEngg'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_TelecomEngg.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\867146595.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_TelecomEngg = sns.distplot(df_data['TelecomEngg'],
```



[158]:
```
# Poisson Distribution for Telecom Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_TelecomEngg = sns.distplot(df_data['TelecomEngg'],
                  kde=False,
                  color='blue')
poisson_distribution_TelecomEngg.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2673912038.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_TelecomEngg = sns.distplot(df_data['TelecomEngg'],
```



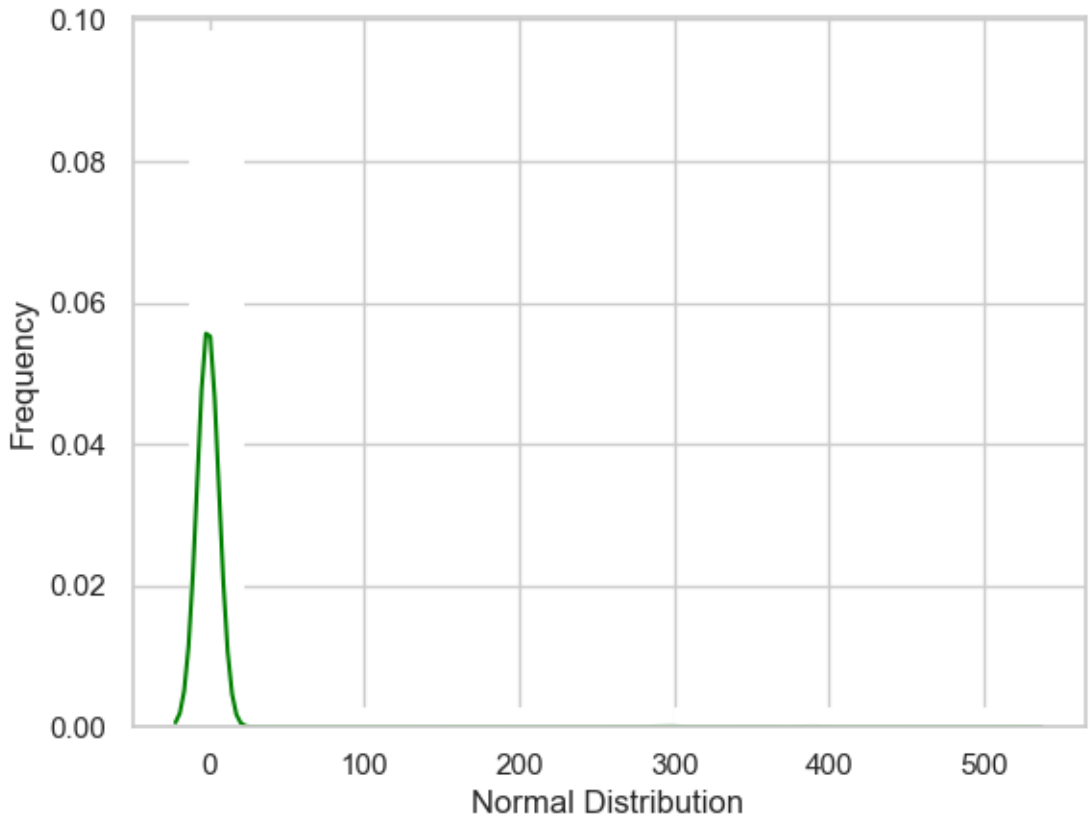
[159]:
```
# Normal Distribution for Civil Engineering

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_CivilEngg= sns.distplot(df_data['CivilEngg'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_CivilEngg.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2752929885.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
```

https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

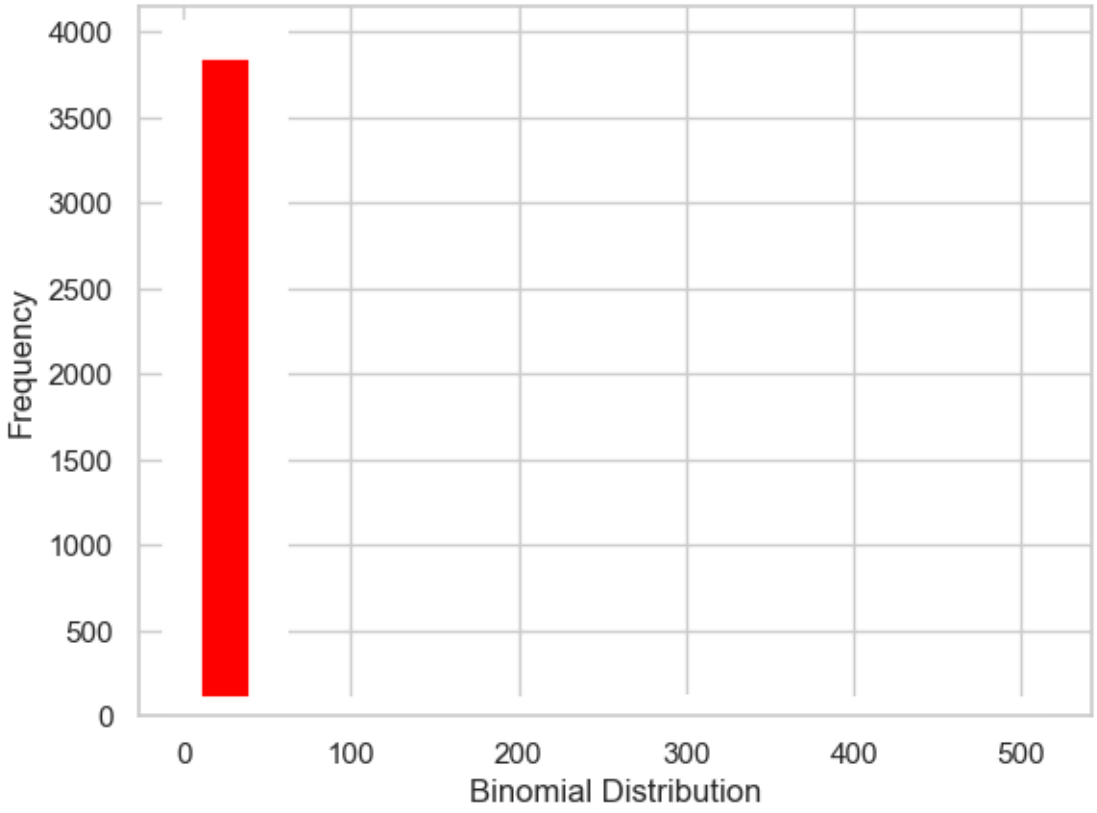
```
normal_distribution_CivilEngg= sns.distplot(df_data['CivilEngg'],
```

[159]:
```
[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]
```



[160]:
```
# Binomial Distribution for Civil Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_CivilEngg = sns.distplot(df_data['CivilEngg'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_CivilEngg.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

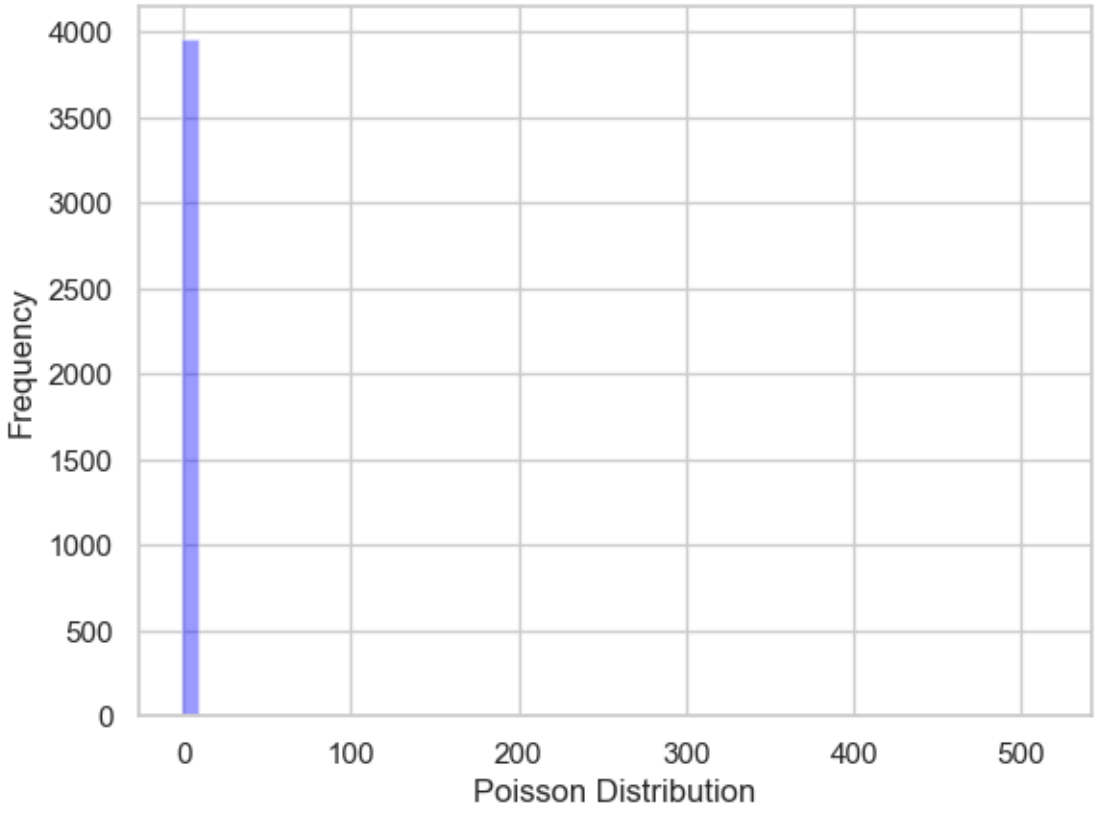
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3197750468.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_CivilEngg = sns.distplot(df_data['CivilEngg'],
```



[161]:
```
# Poisson Distribution for Civil Engineering

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_CivilEngg = sns.distplot(df_data['CivilEngg'],
                  kde=False,
```

```
                  color='blue')
poisson_distribution_CivilEngg.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3761595198.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_CivilEngg = sns.distplot(df_data['CivilEngg'],
```

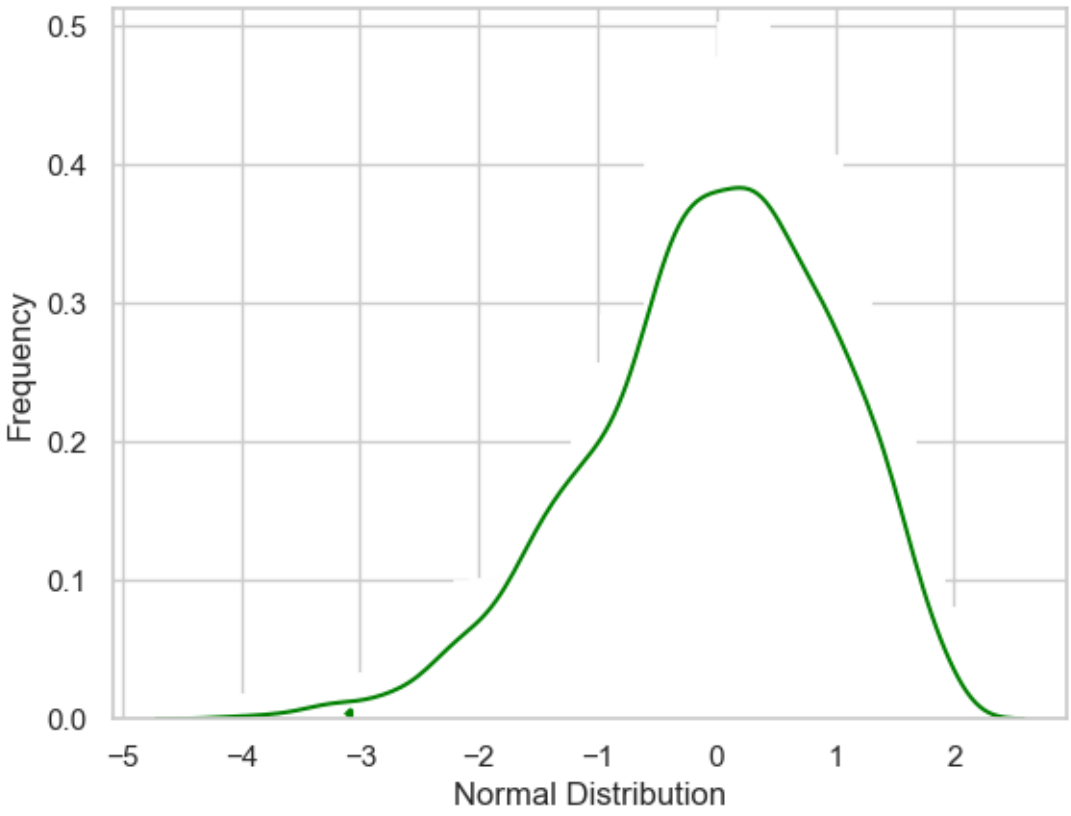


[162]:
```
# Normal Distribution for conscientiousness

import scipy.stats as stats
```

```
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_conscientiousness= sns.
 ↪distplot(df_data['conscientiousness'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_conscientiousness.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1795343295.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_conscientiousness=
sns.distplot(df_data['conscientiousness'],
```

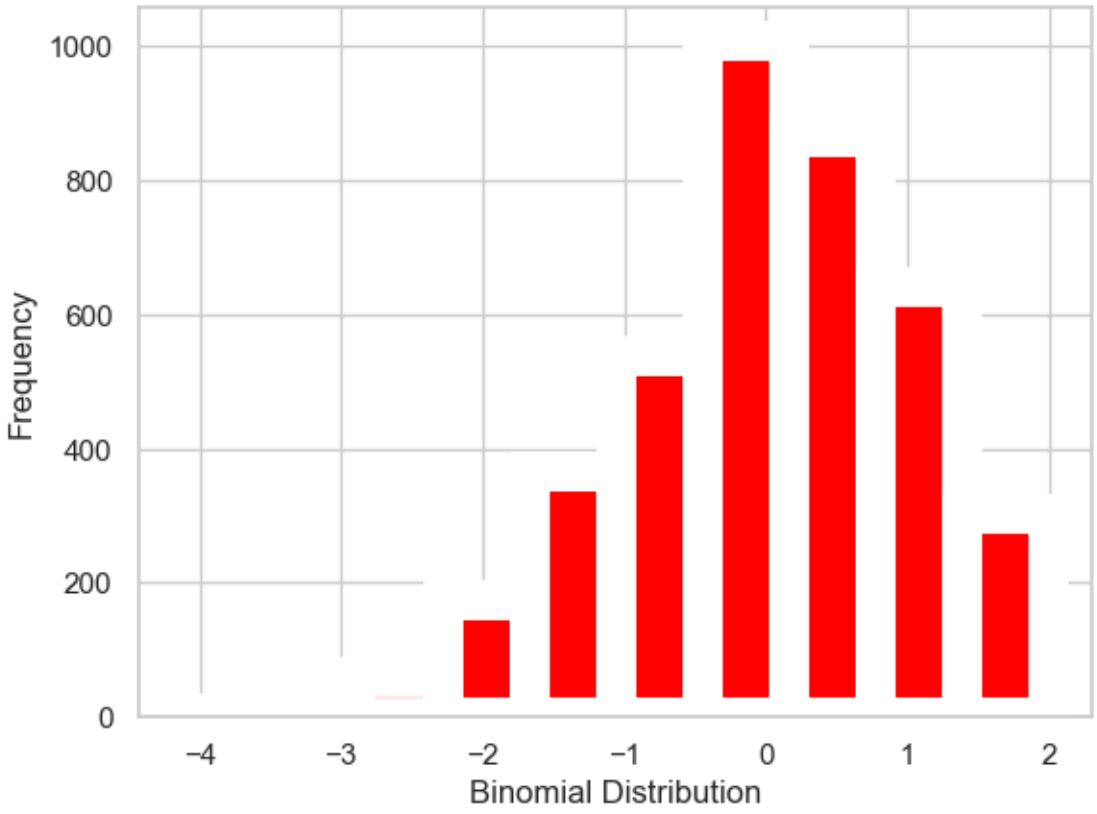
[162]: [Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]



[163]:
```
# Binomial Distribution for conscientiousness

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_conscientiousness = sns.
 ↪distplot(df_data['conscientiousness'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_conscientiousness.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2359769432.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

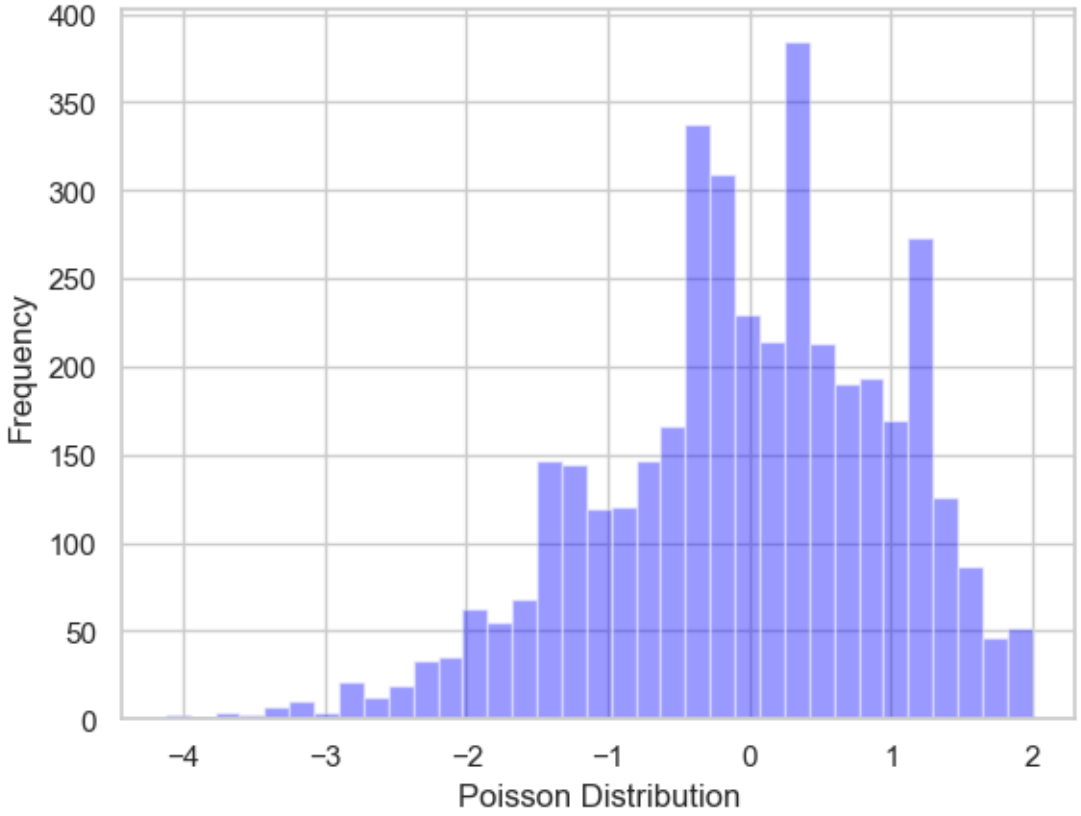
```
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_conscientiousness =
sns.distplot(df_data['conscientiousness'],
```



[164]:
```
# Poisson Distribution for conscientiousness

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

poisson_distribution_conscientiousness = sns.
 ↪distplot(df_data['conscientiousness'],
                  kde=False,
                  color='blue')
poisson_distribution_conscientiousness.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')
```

```
plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\1451433305.py:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_conscientiousness =
sns.distplot(df_data['conscientiousness'],
```

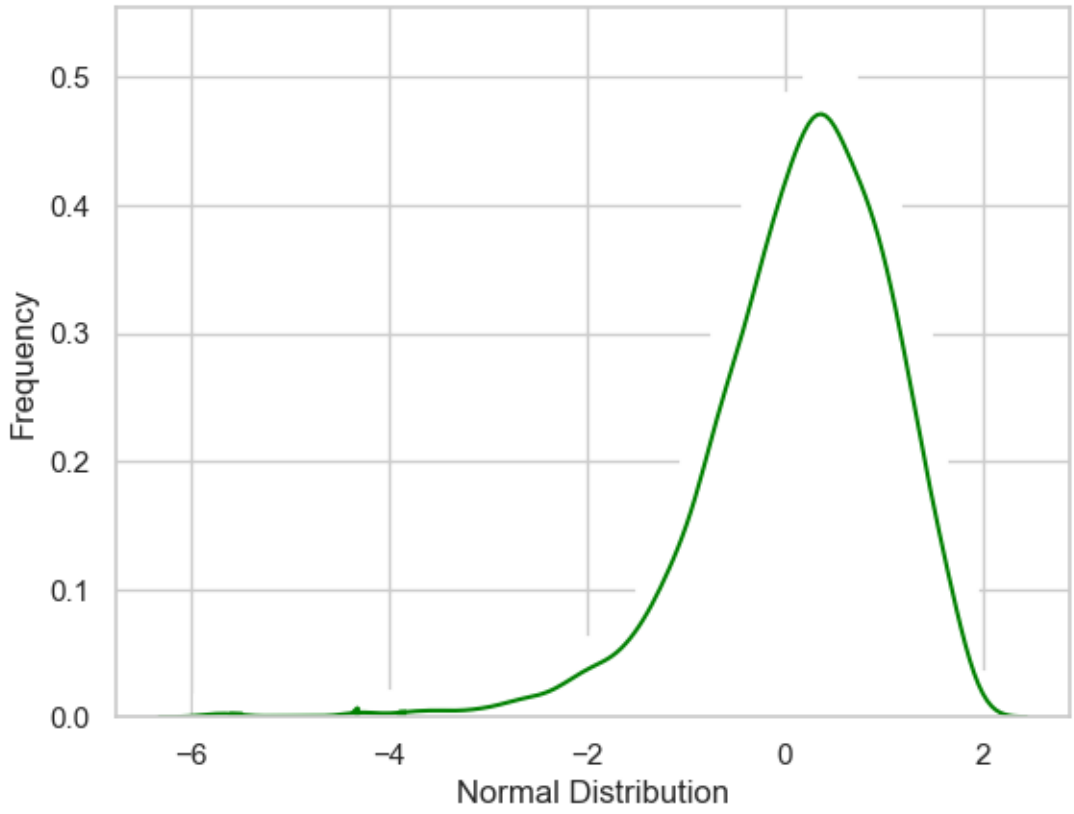


[165]:
```
# Normal Distribution for agreeableness

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_agreeableness= sns.distplot(df_data['agreeableness'],
```

```
                    bins=50,
                    kde=True,
                    color='green',
                    hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_agreeableness.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

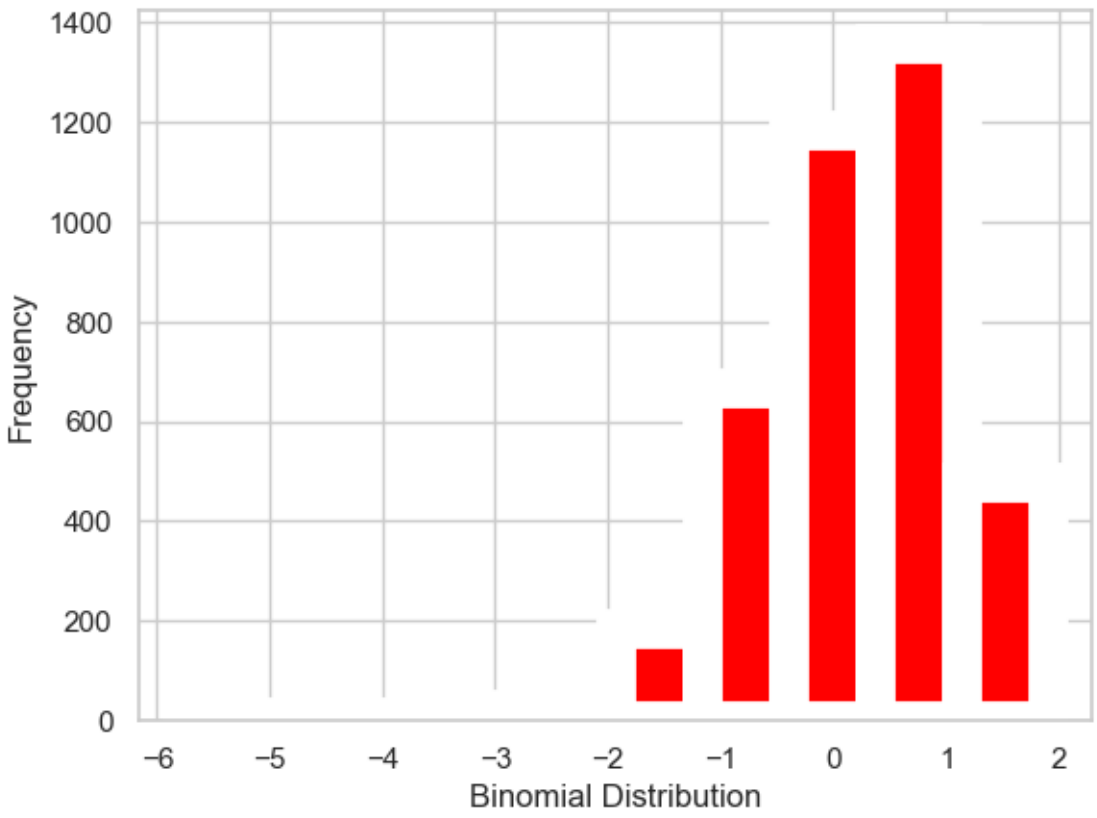
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2985260873.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_agreeableness= sns.distplot(df_data['agreeableness'],
```

[165]: [Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]



[166]:
```
# Binomial Distribution for agreeableness

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_agreeableness = sns.distplot(df_data['agreeableness'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_agreeableness.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

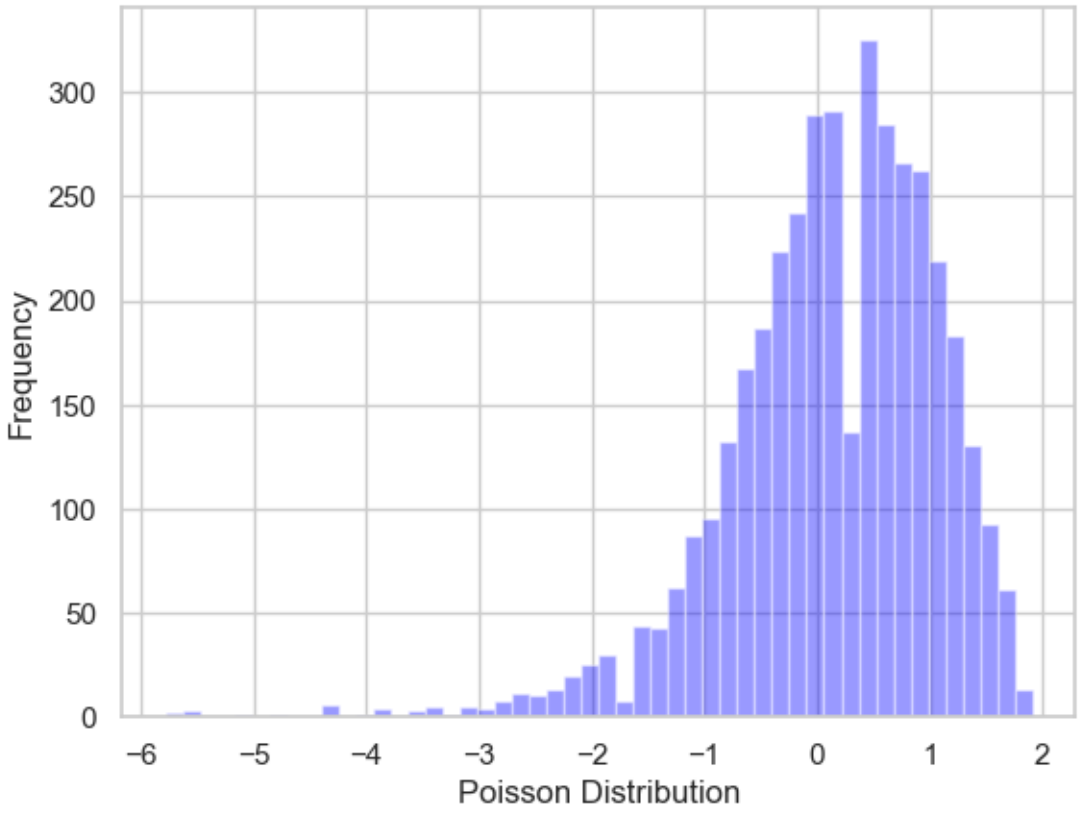
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\288167759.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_agreeableness = sns.distplot(df_data['agreeableness'],
```



[167]:
```
# Poisson Distribution for agreeableness

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_agreeableness = sns.distplot(df_data['agreeableness'],
                  kde=False,
                  color='blue')
poisson_distribution_agreeableness.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\173311890.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_agreeableness = sns.distplot(df_data['agreeableness'],
```

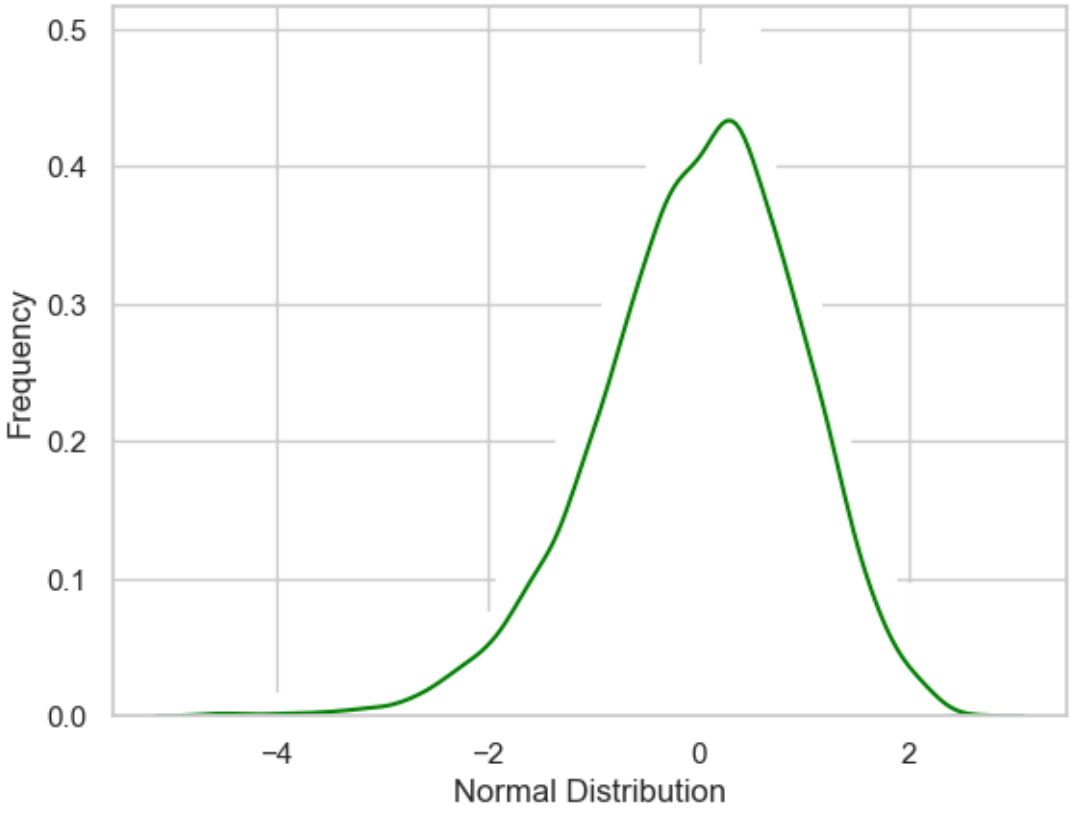


[168]:
```
# Normal Distribution for extraversion

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_extraversion= sns.distplot(df_data['extraversion'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_extraversion.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\111868112.py:6: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_extraversion= sns.distplot(df_data['extraversion'],
```

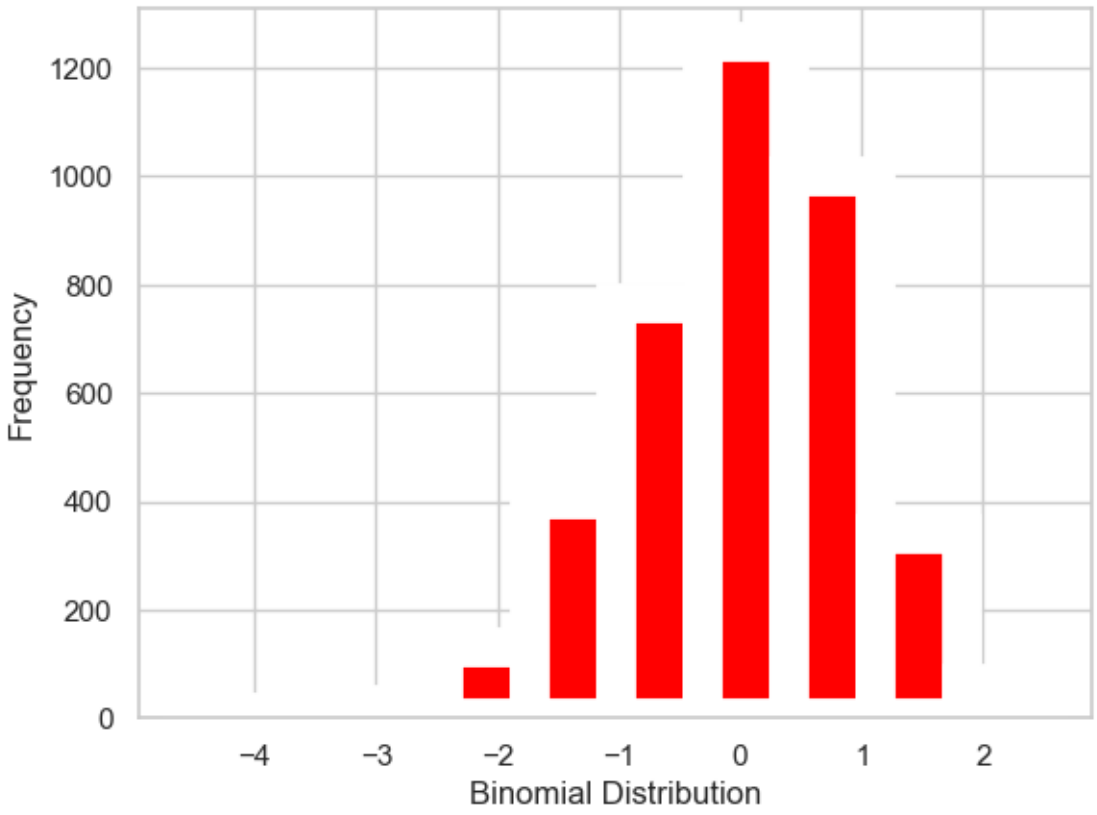
[168]: `[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]`



[169]:
```
# Binomial Distribution for extraversion

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_extraversion = sns.distplot(df_data['extraversion'],
                 bins=10,
                 kde=False,
                 color='red',
```

```
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_extraversion.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

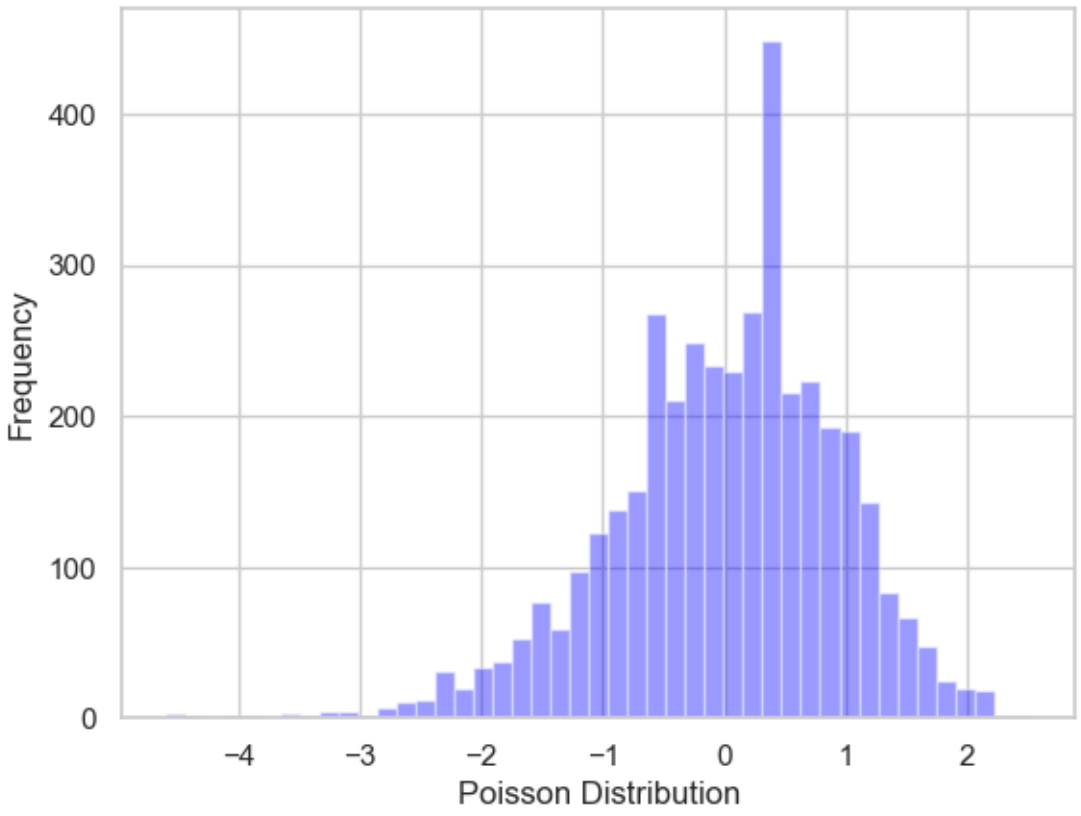
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3106836883.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_extraversion = sns.distplot(df_data['extraversion'],
```



[170]:
```
# Poisson Distribution for extraversion

import seaborn as sns
```

```
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_extraversion = sns.distplot(df_data['extraversion'],
                  kde=False,
                  color='blue')
poisson_distribution_extraversion.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

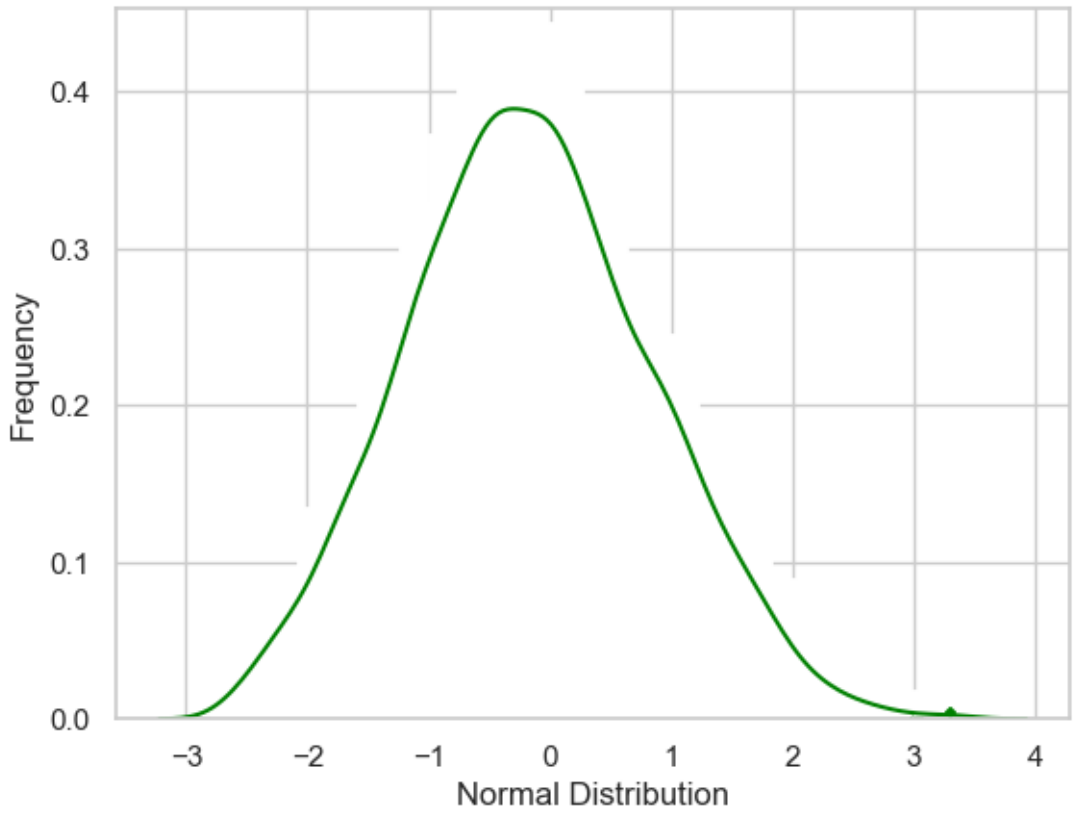
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\534928021.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_extraversion = sns.distplot(df_data['extraversion'],
```



[171]:
```
# Normal Distribution for nueroticism

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
                  bins=50,
                  kde=True,
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_nueroticism.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3469608490.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
```

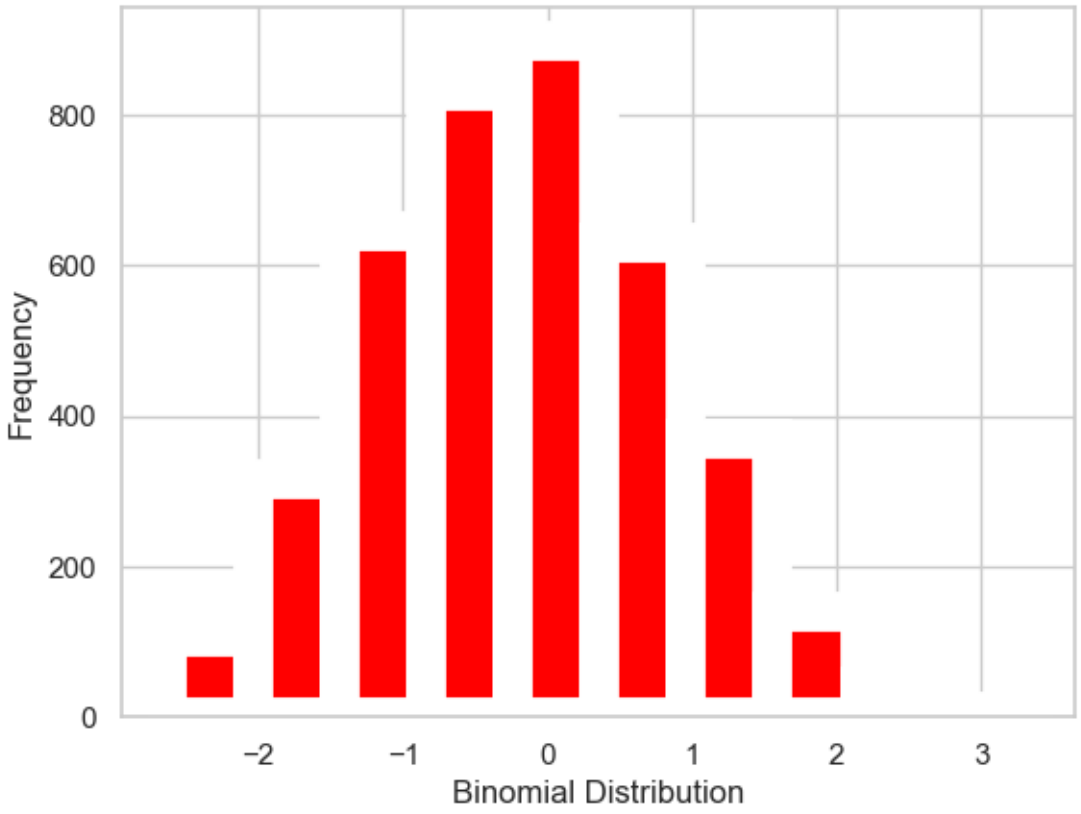
[171]:
```
[Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]
```



[172]:
```
# Binomial Distribution for nueroticism

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_nueroticism.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

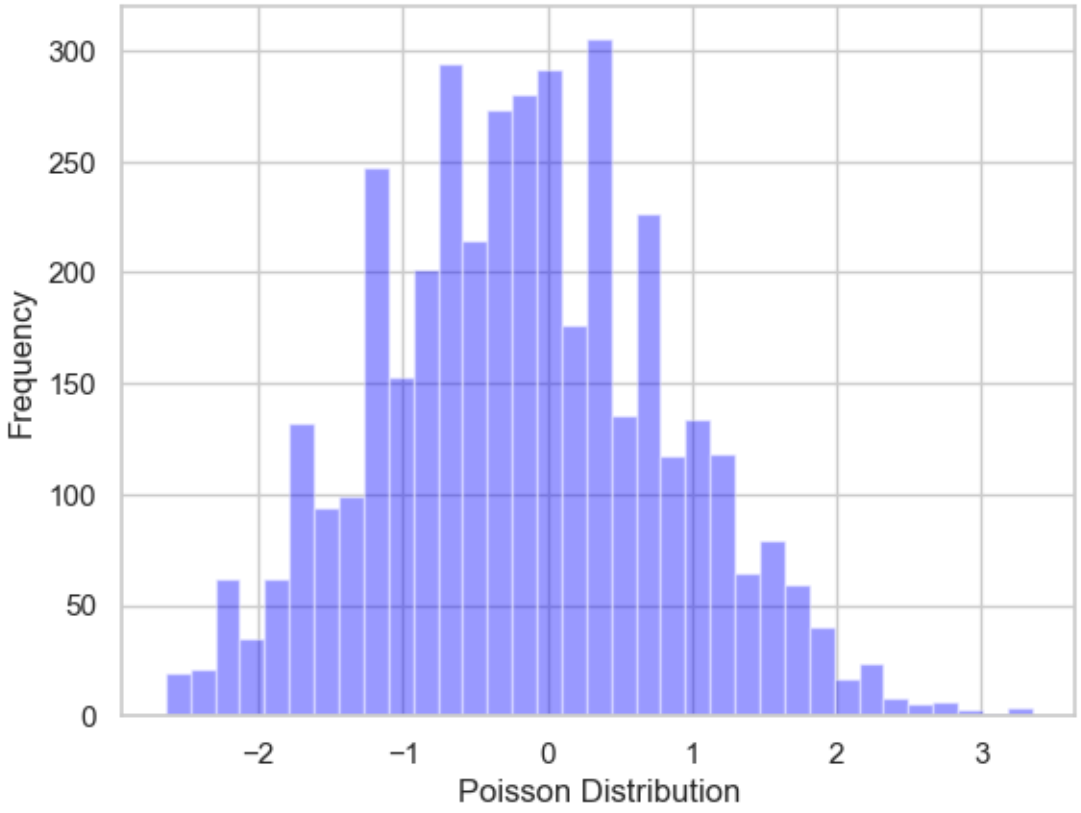
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\2910340560.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
```



[173]:
```
# Poisson Distribution for neuroticism

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
                 kde=False,
                 color='blue')
poisson_distribution_nueroticism.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\677355635.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_nueroticism = sns.distplot(df_data['nueroticism'],
```

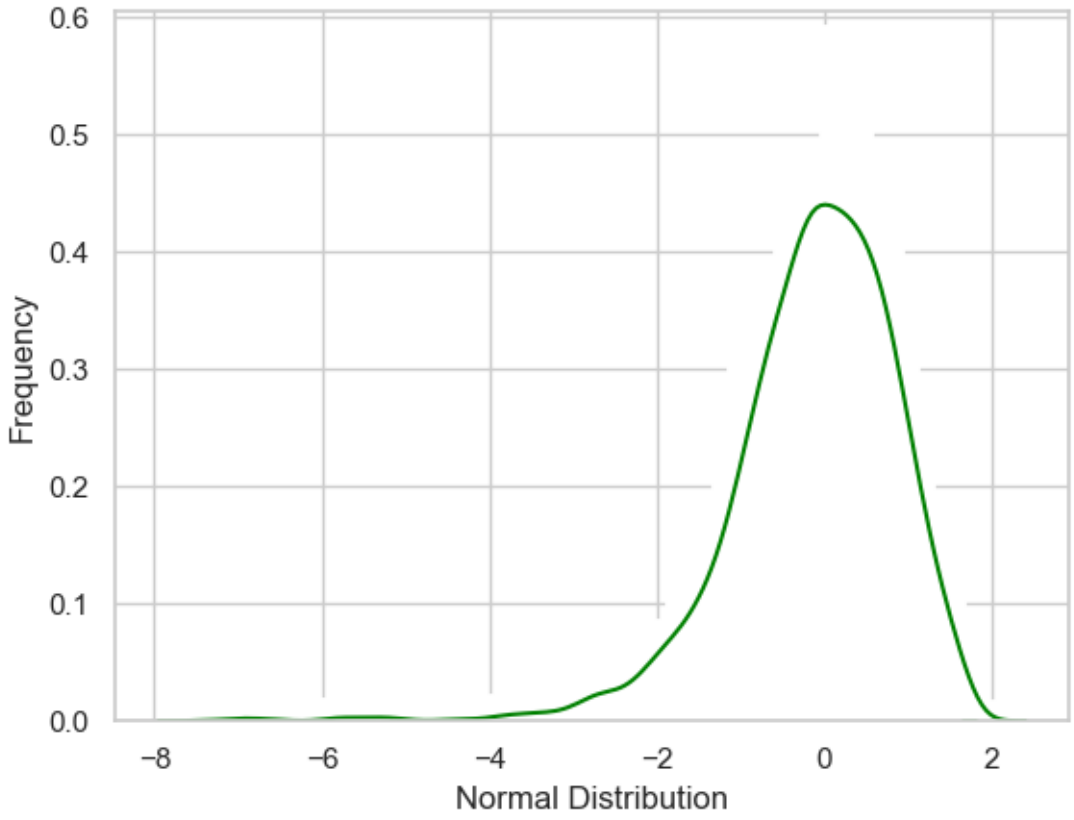


[174]:
```
# Normal Distribution for open_ness_experience

import scipy.stats as stats
import seaborn as sns
import matplotlib.pyplot as plt
normal_distribution_openess_to_experience = sns.
 ↪distplot(df_data['openess_to_experience'],
                 bins=50,
                 kde=True,
```

```
                  color='green',
                  hist_kws={"linewidth": 15,'alpha':1})
normal_distribution_openess_to_experience.set(xlabel='Normal Distribution',␣
 ↪ylabel='Frequency')
```

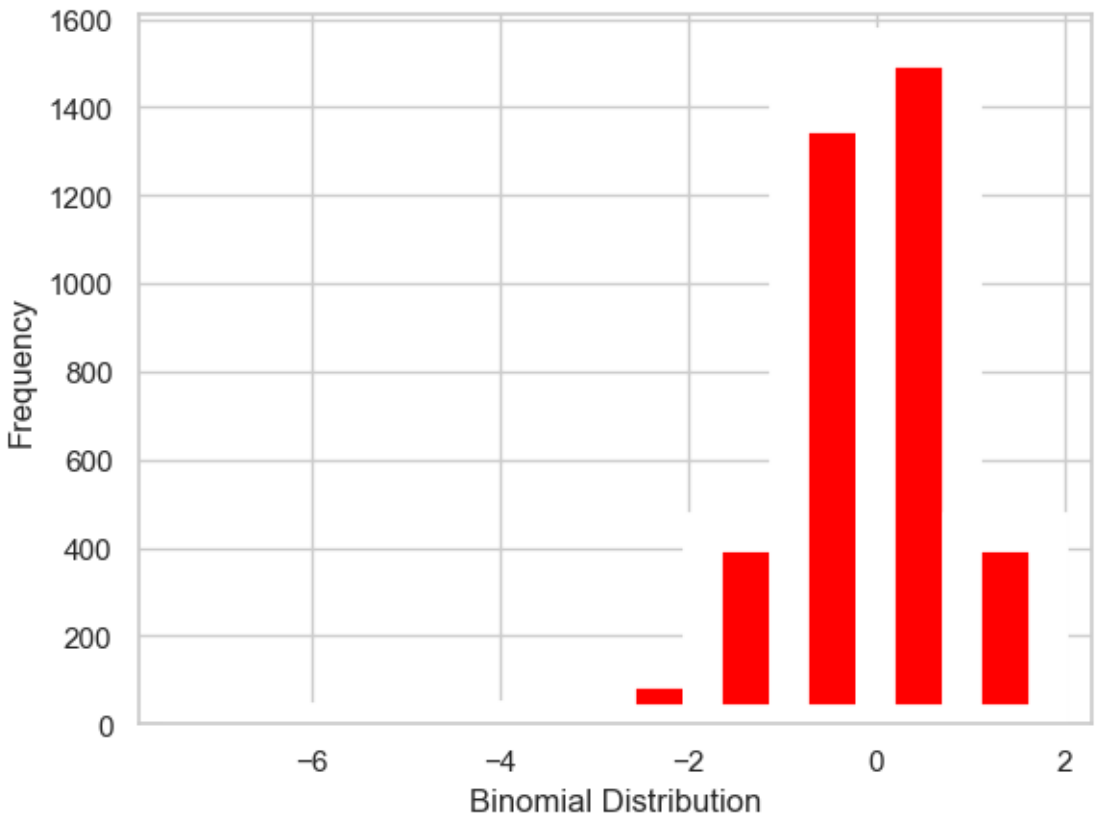
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3753357938.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  normal_distribution_openess_to_experience =
sns.distplot(df_data['openess_to_experience'],
```

[174]: [Text(0.5, 0, 'Normal Distribution'), Text(0, 0.5, 'Frequency')]



[175]:
```
# Binomial Distribution for openess_to_experience

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
binomial_distribution_openess_to_experience = sns.
 ↪distplot(df_data['openess_to_experience'],
                  bins=10,
                  kde=False,
                  color='red',
                  hist_kws={"linewidth": 15, 'alpha': 1})
binomial_distribution_openess_to_experience.set(xlabel='Binomial Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

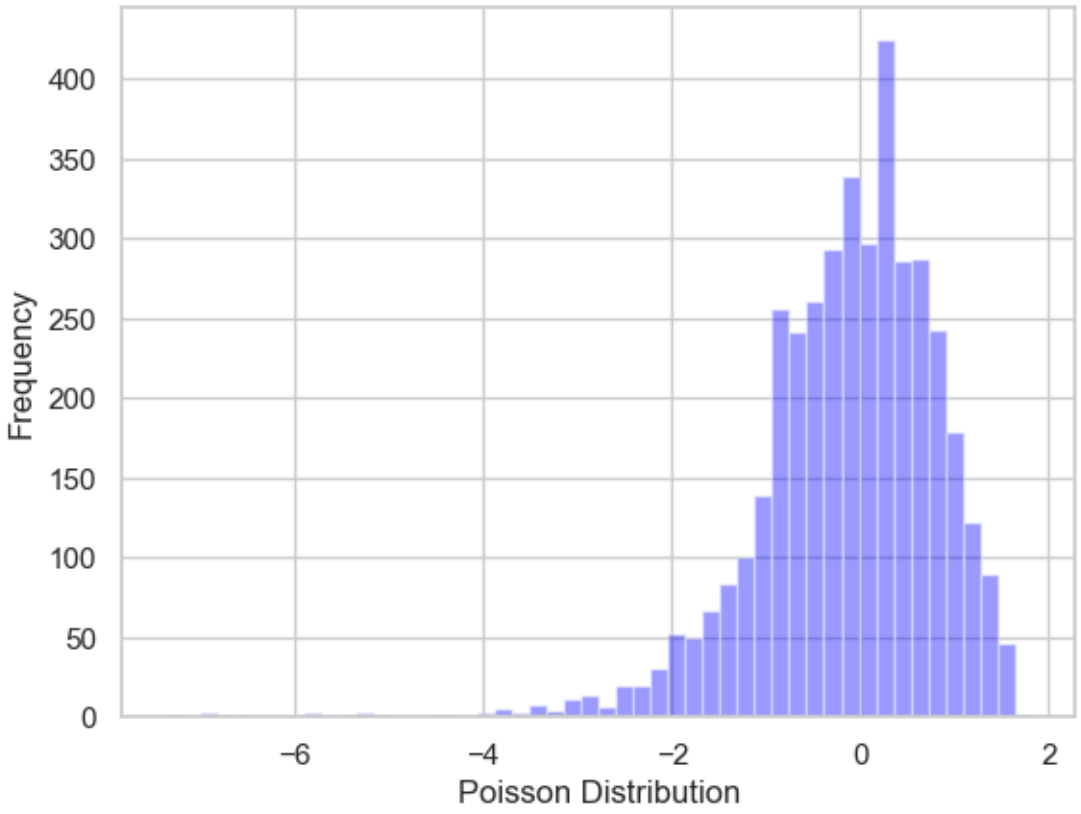
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3244029191.py:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  binomial_distribution_openess_to_experience =
sns.distplot(df_data['openess_to_experience'],
```



[176]:
```
# Poisson Distribution for openess_to_experience

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# plotting a histogram
poisson_distribution_openess_to_experience = sns.
 ↪distplot(df_data['openess_to_experience'],
                 kde=False,
                 color='blue')
poisson_distribution_openess_to_experience.set(xlabel='Poisson Distribution',␣
 ↪ylabel='Frequency')

plt.show()
```

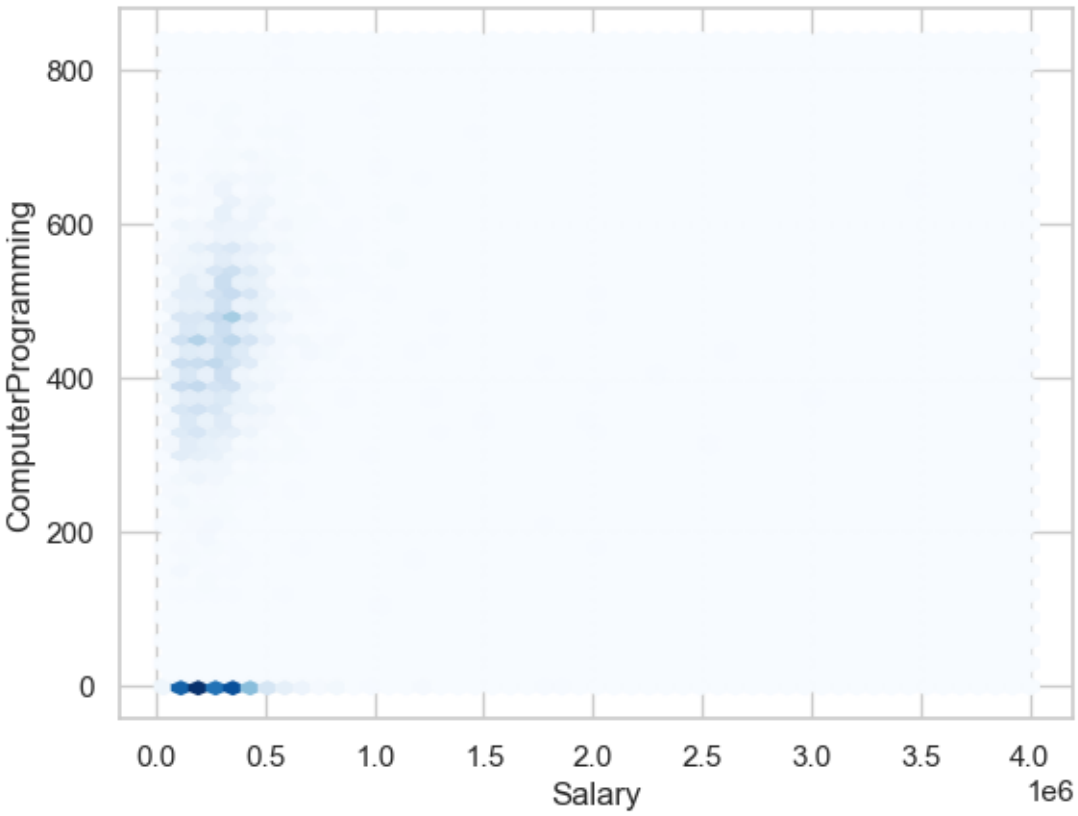
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_13628\3052673081.py:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

```
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  poisson_distribution_openess_to_experience =
sns.distplot(df_data['openess_to_experience'],
```



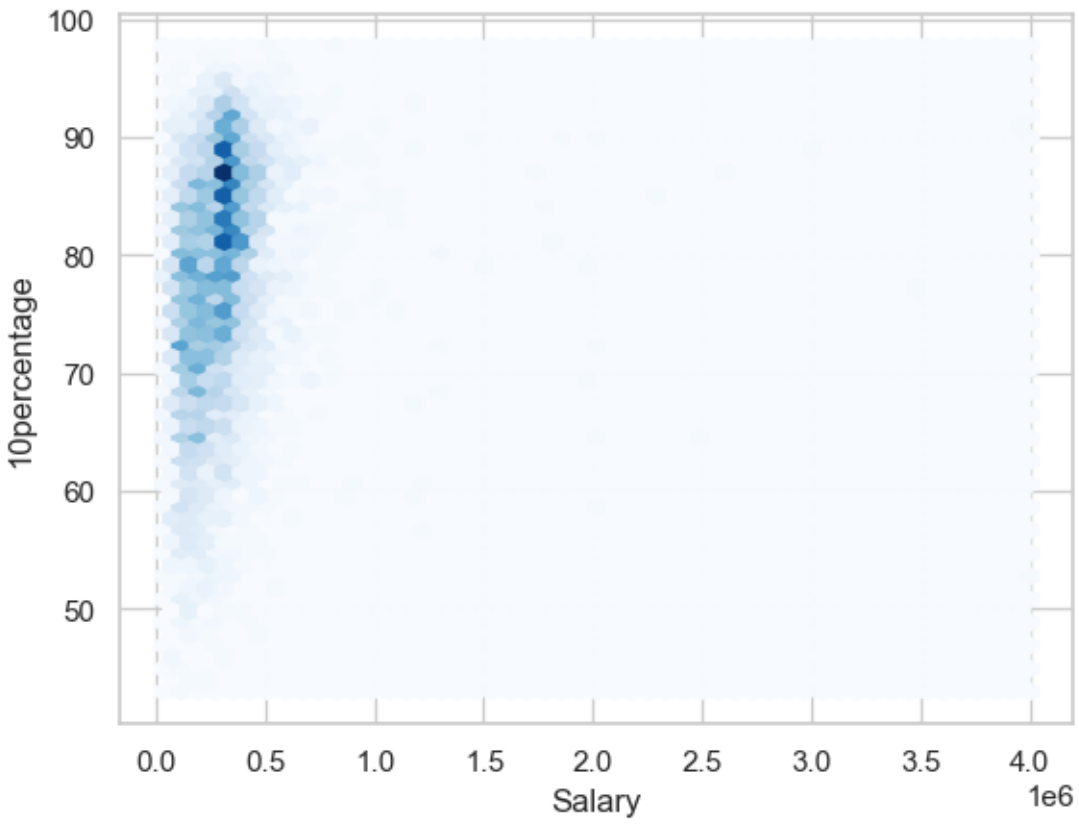
[177]:
```
plt.hexbin(df_data['Salary'], df_data['ComputerProgramming'], gridsize = 50,␣
  ↪cmap ='Blues')
plt.xlabel('Salary')
plt.ylabel('ComputerProgramming')
```

[177]:
```
Text(0, 0.5, 'ComputerProgramming')
```



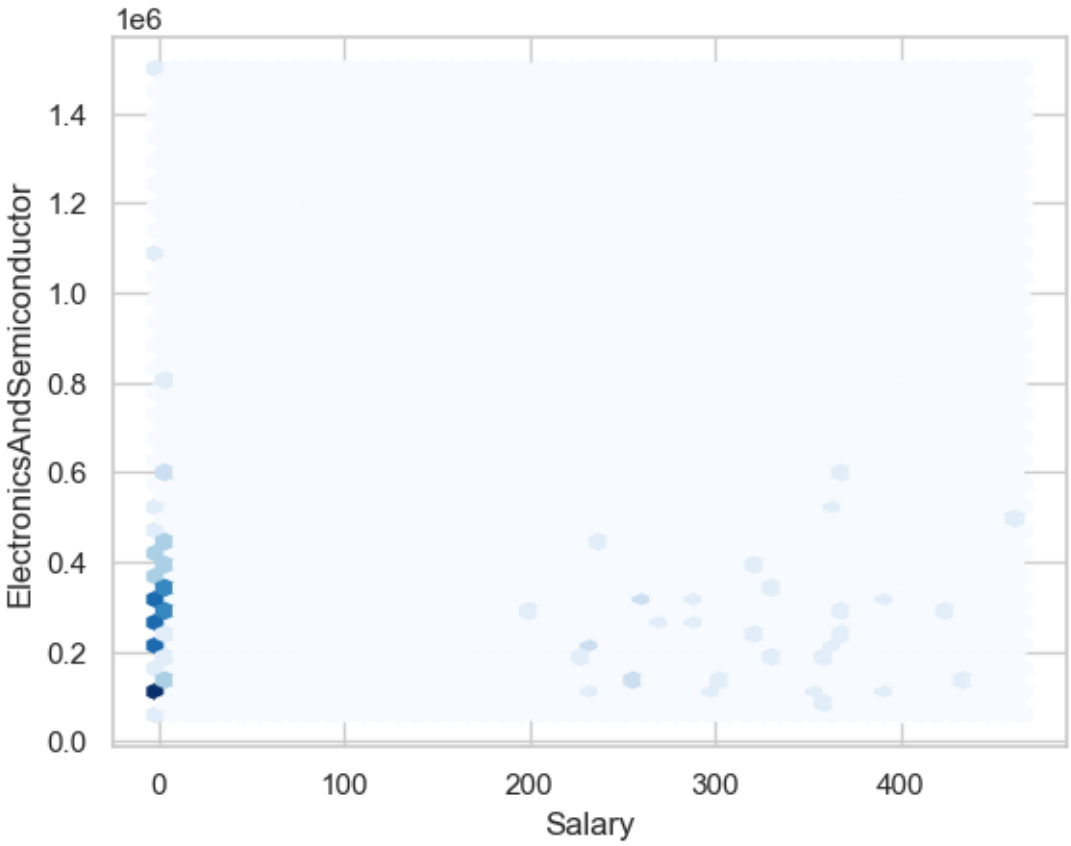
[178]:
```
import matplotlib.pyplot as plt
plt.hexbin(df_data['Salary'], df_data['10percentage'], gridsize = 50, cmap␣
↪='Blues')
plt.xlabel('Salary')
plt.ylabel('10percentage')
```

[178]:
```
Text(0, 0.5, '10percentage')
```



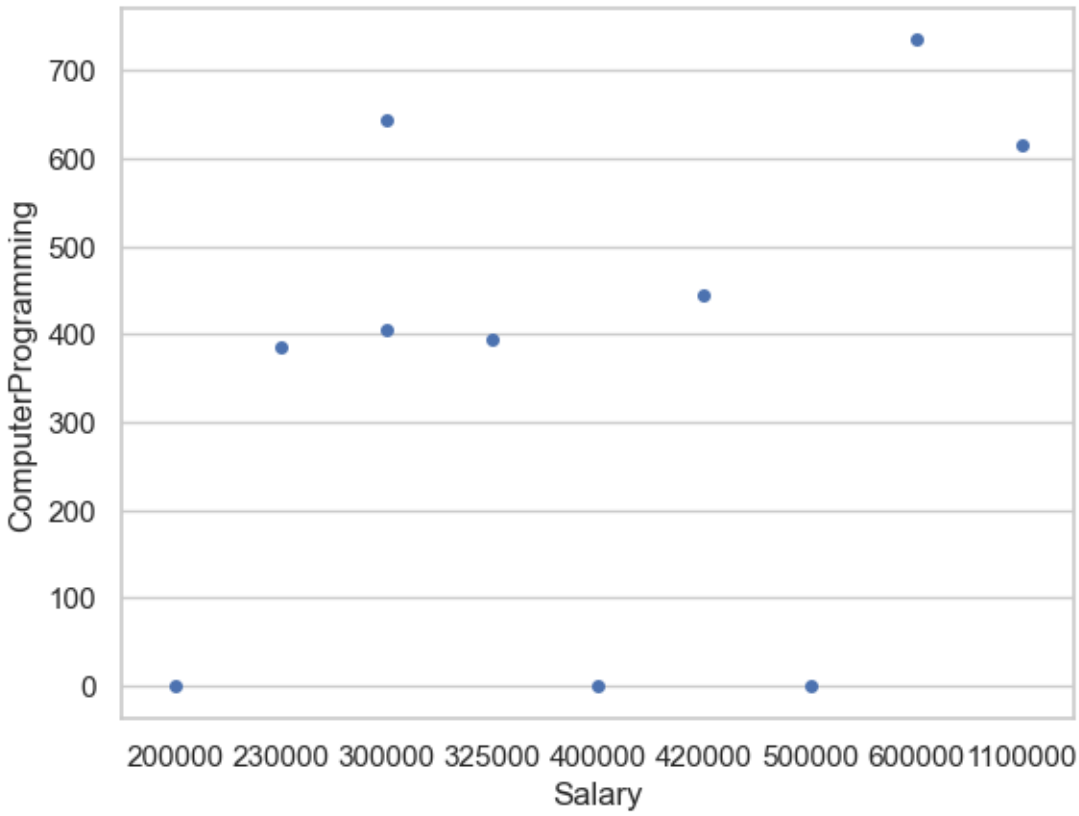
[179]:
```
plt.hexbin(x=df_data['ElectronicsAndSemicon'].head(100),y=df_data['Salary'].
  ↪head(100),gridsize=50,cmap='Blues')
plt.xlabel('Salary')
plt.ylabel('ElectronicsAndSemiconductor')
```

[179]:
```
Text(0, 0.5, 'ElectronicsAndSemiconductor')
```



[180]:
```
sns.set(style='whitegrid')
sns.swarmplot(x="Salary",
                  y="ComputerProgramming",
                  data=df_data.head(10),orient="vertical")
```

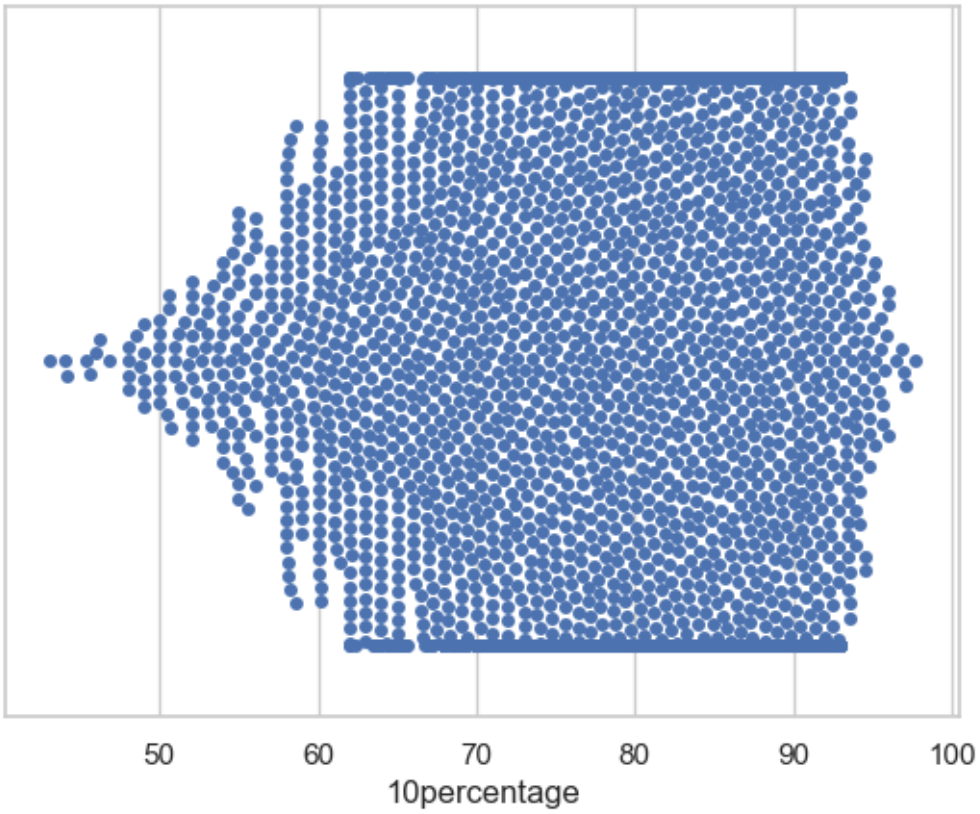
[180]: <Axes: xlabel='Salary', ylabel='ComputerProgramming'>



[182]:
```
sns.set(style="whitegrid")
sns.swarmplot(x=df_data["10percentage"])
```

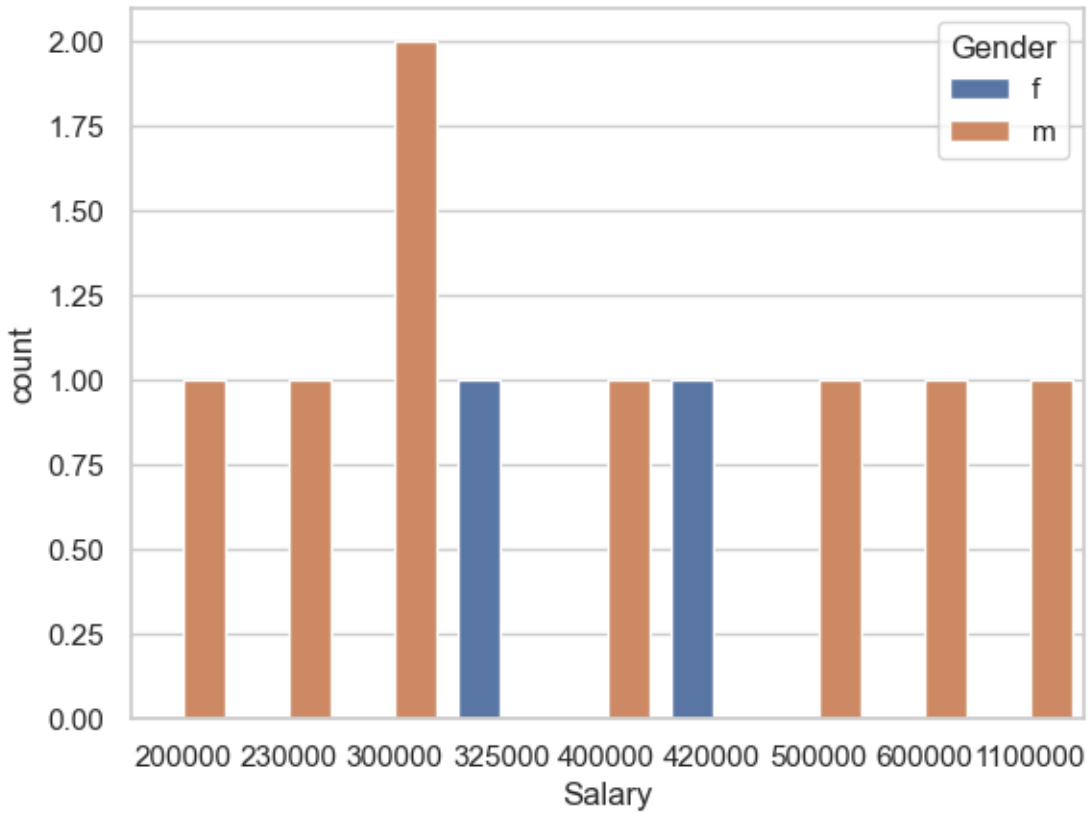
[182]: <Axes: xlabel='10percentage'>

```
C:\Users\madhu\anaconda\Lib\site-packages\seaborn\categorical.py:3544:
UserWarning: 53.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\madhu\anaconda\Lib\site-packages\seaborn\categorical.py:3544:
UserWarning: 53.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



[181]:
```
sns.countplot(x=df_data['Salary'].head(10),
              hue=df_data['Gender'].head(10),orient='h')
```

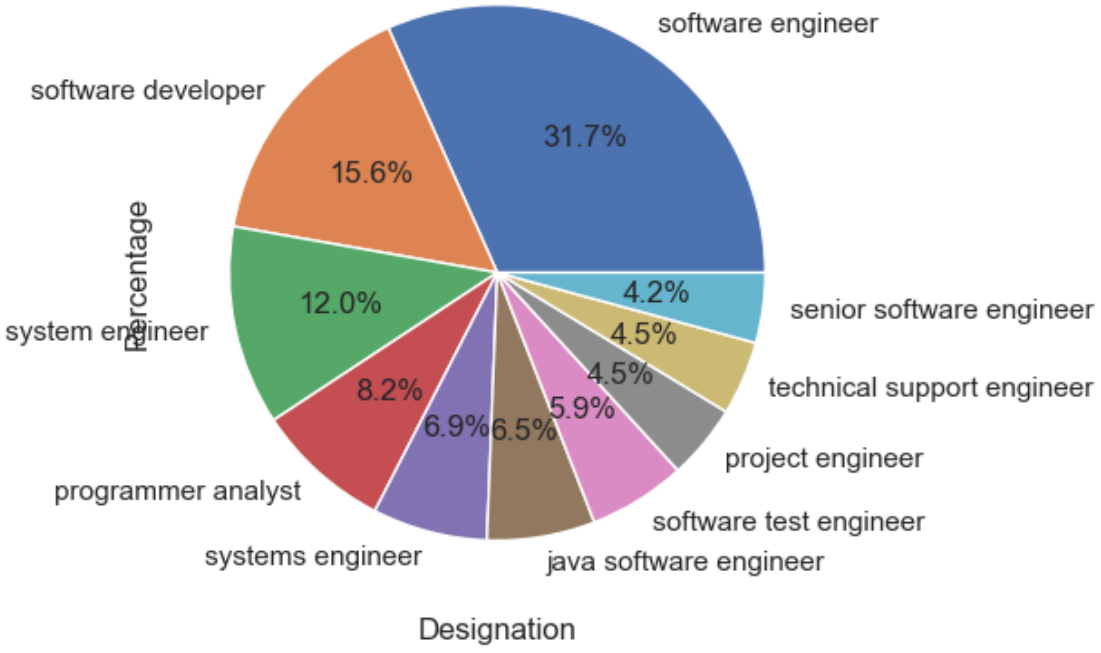
[181]: <Axes: xlabel='Salary', ylabel='count'>



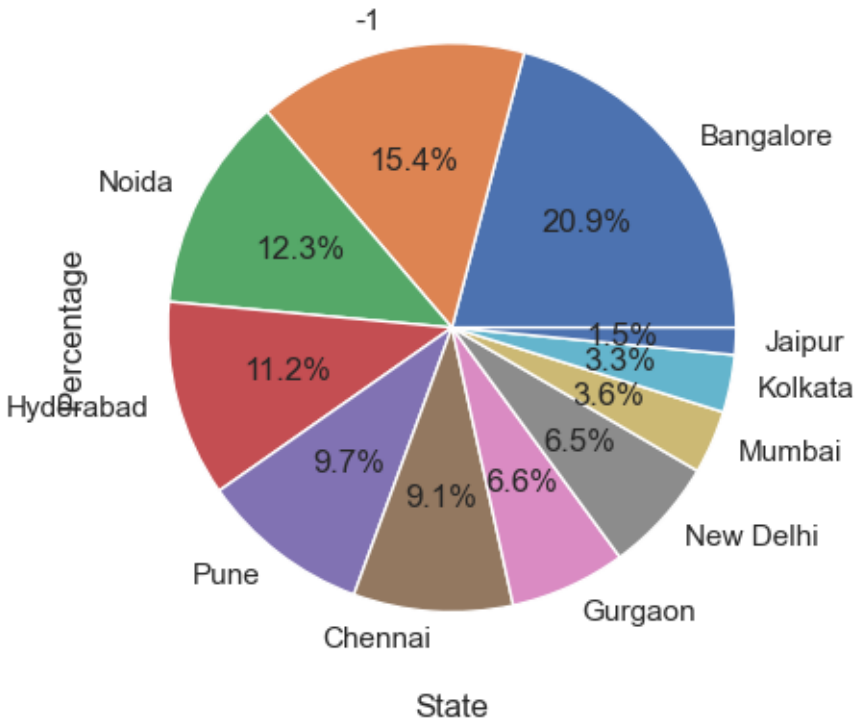
[182]:
```
# Find the correlation between Salary and Computer Programming

x=df_data['Salary']
y=df_data['ComputerProgramming']
np.corrcoef(x,y)
```

[182]:
```
array([[1.        , 0.1156648],
       [0.1156648, 1.       ]])
```

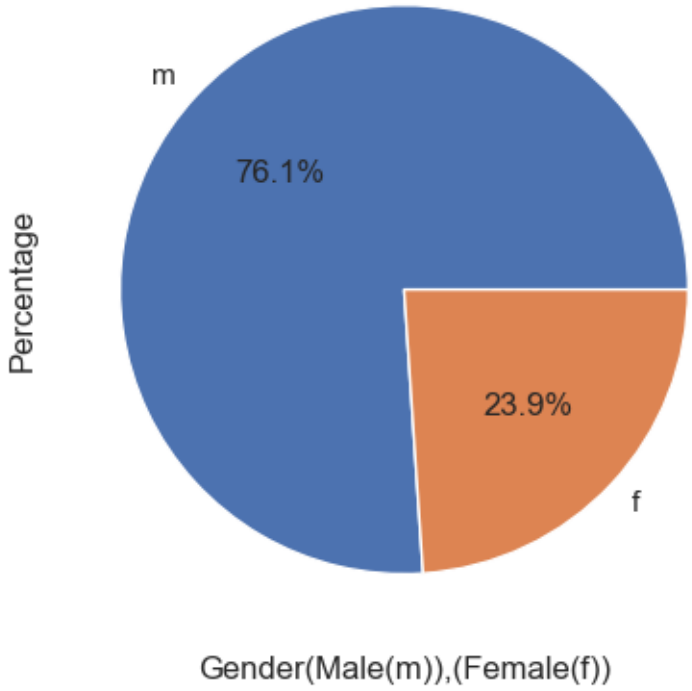
[183]:
```
# Pieplot for the Designation column of dataset

import matplotlib.pyplot as plt

x = df_data['Designation'].value_counts().head(10)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Designation')
plt.ylabel('Percentage')
plt.show()
```



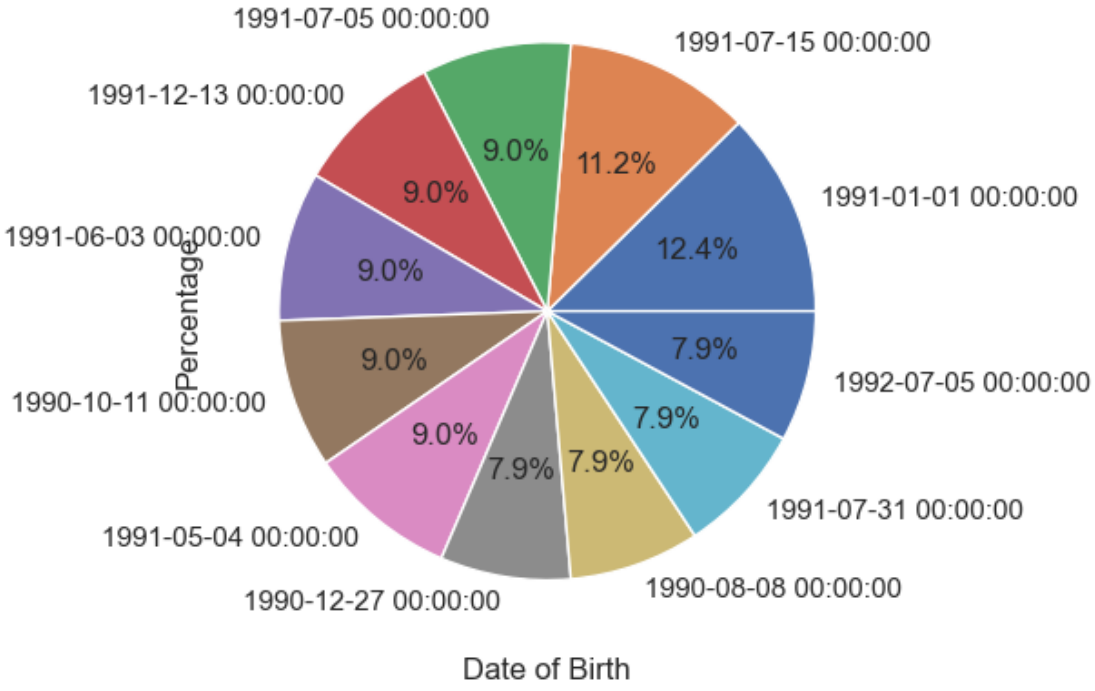
[184]:
```
# Pieplot for the JobCity column of dataset

import matplotlib.pyplot as plt

x = df_data['JobCity'].value_counts().head(11)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('State')
plt.ylabel('Percentage')
plt.show()
```



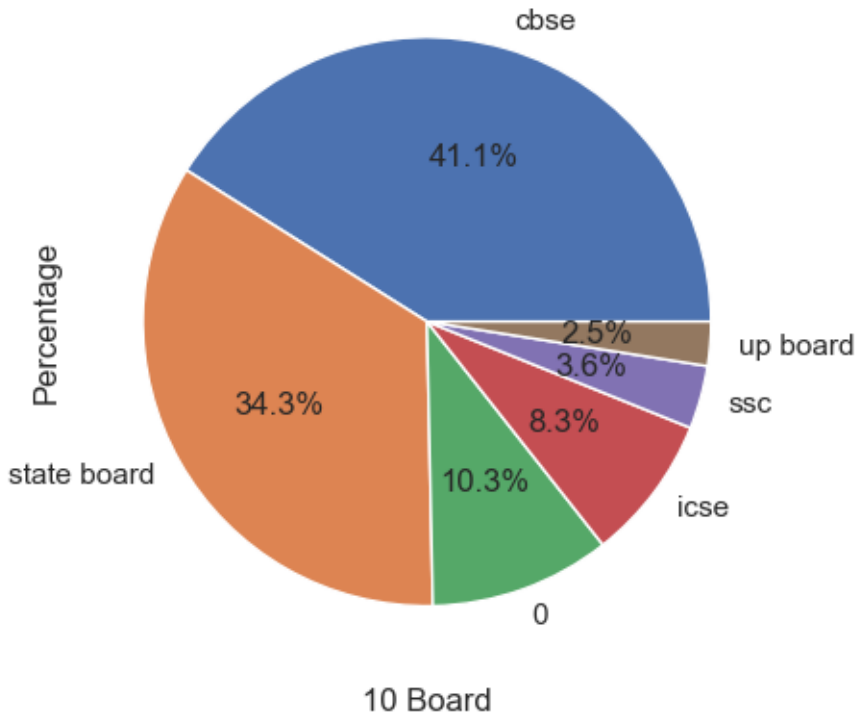
[185]:
```
# Pieplot for the JobCity column of dataset

import matplotlib.pyplot as plt

x = df_data['Gender'].value_counts().head(11)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Gender(Male(m)),(Female(f))')
plt.ylabel('Percentage')
plt.show()
```



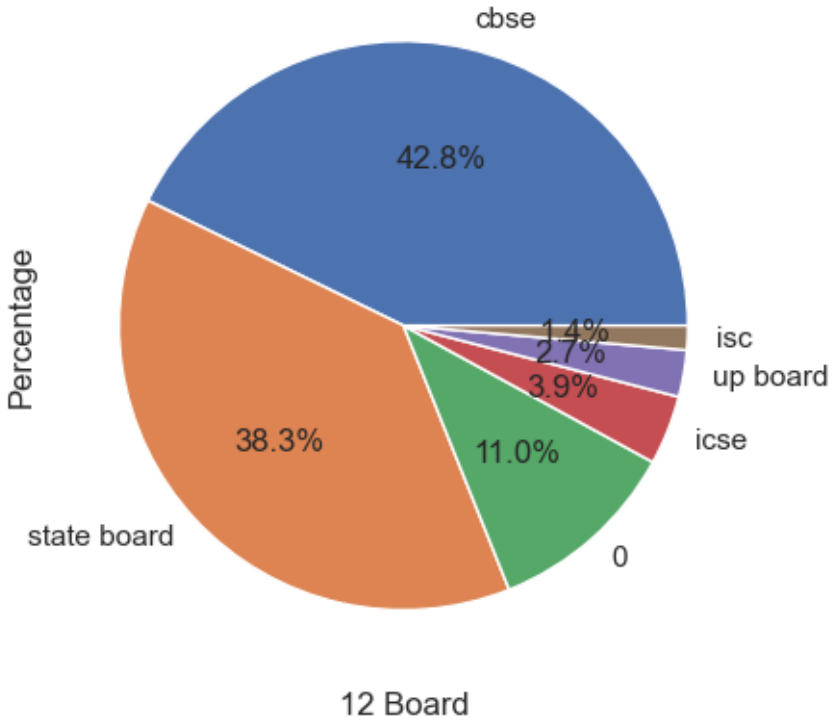
[186]:
```
# Pieplot for the DOB column of dataset

import matplotlib.pyplot as plt

x = df_data['DOB'].value_counts().head(11)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel("Date of Birth")
plt.ylabel("Percentage")
plt.show()
```



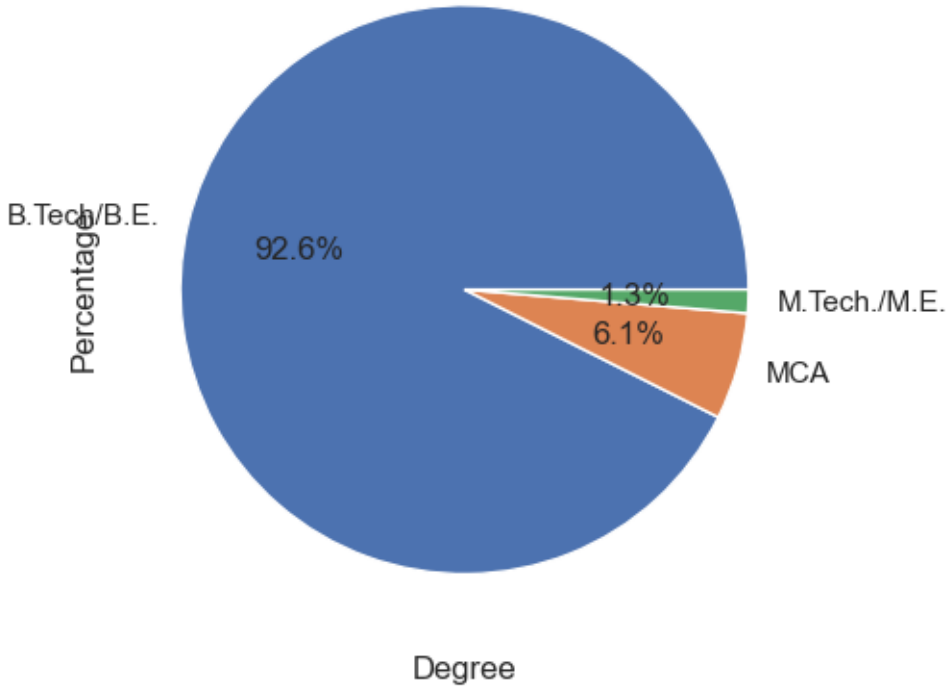
[187]:
```
# Pieplot for the 10board column of dataset

import matplotlib.pyplot as plt

x = df_data['10board'].value_counts().head(6)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('10 Board')
plt.ylabel('Percentage')
plt.show()
```



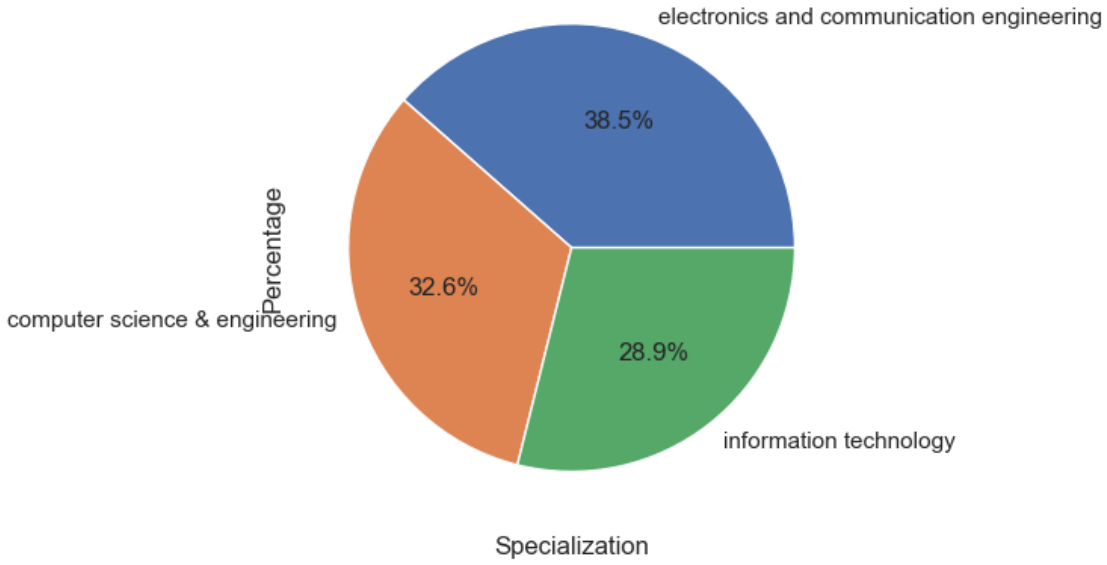
[188]:
```
# Pieplot for the 12board column of dataset

import matplotlib.pyplot as plt

x = df_data['12board'].value_counts().head(6)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('12 Board')
plt.ylabel('Percentage')
plt.show()
```



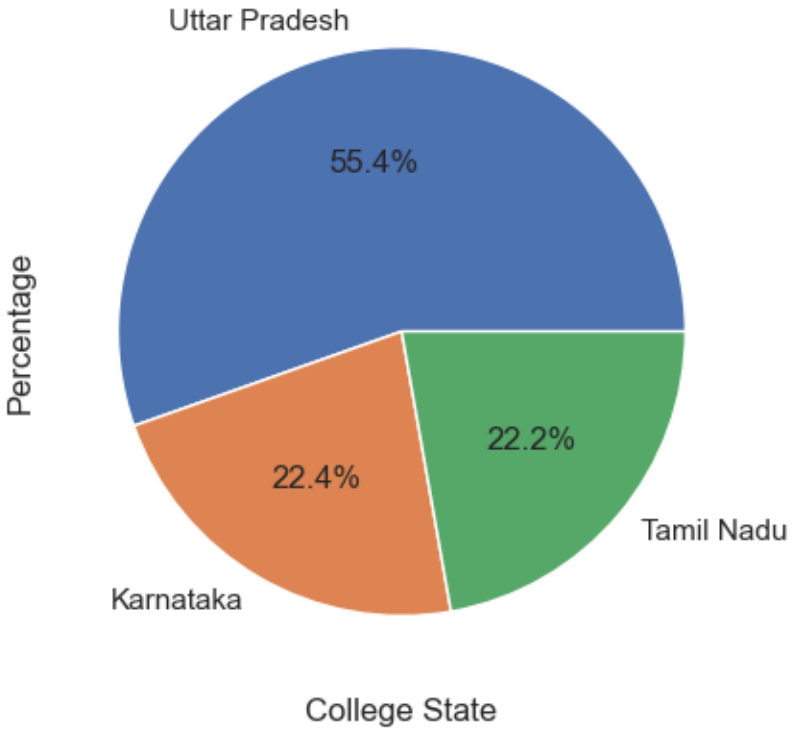
[189]:
```
# Pieplot for the Degree column of dataset

import matplotlib.pyplot as plt

x = df_data['Degree'].value_counts().head(3)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')

plt.xlabel('Degree')
plt.ylabel('Percentage')
plt.show()
```



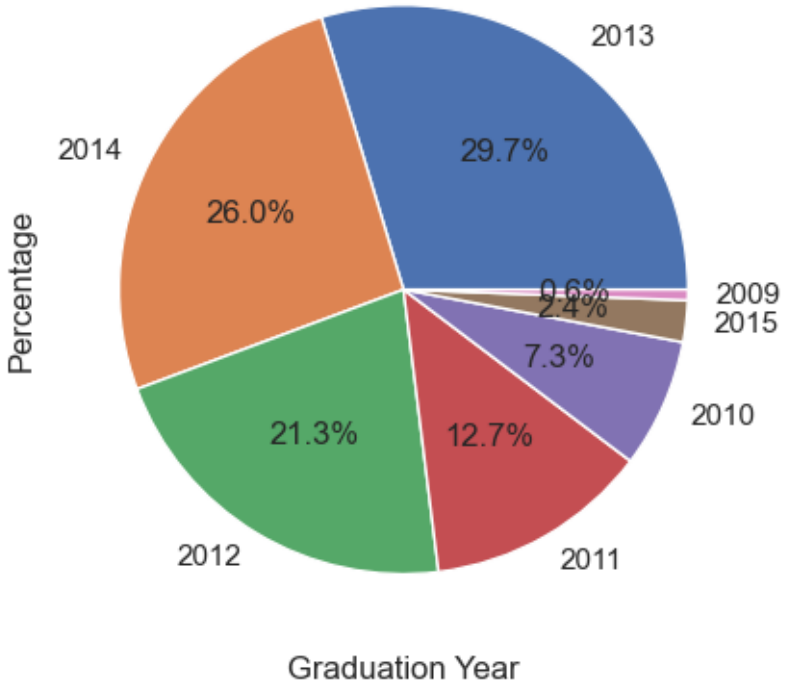
[190]:
```
# Pieplot for the Specialization column of dataset

import matplotlib.pyplot as plt

x = df_data['Specialization'].value_counts().head(3)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Specialization')
plt.ylabel('Percentage')
plt.show()
```



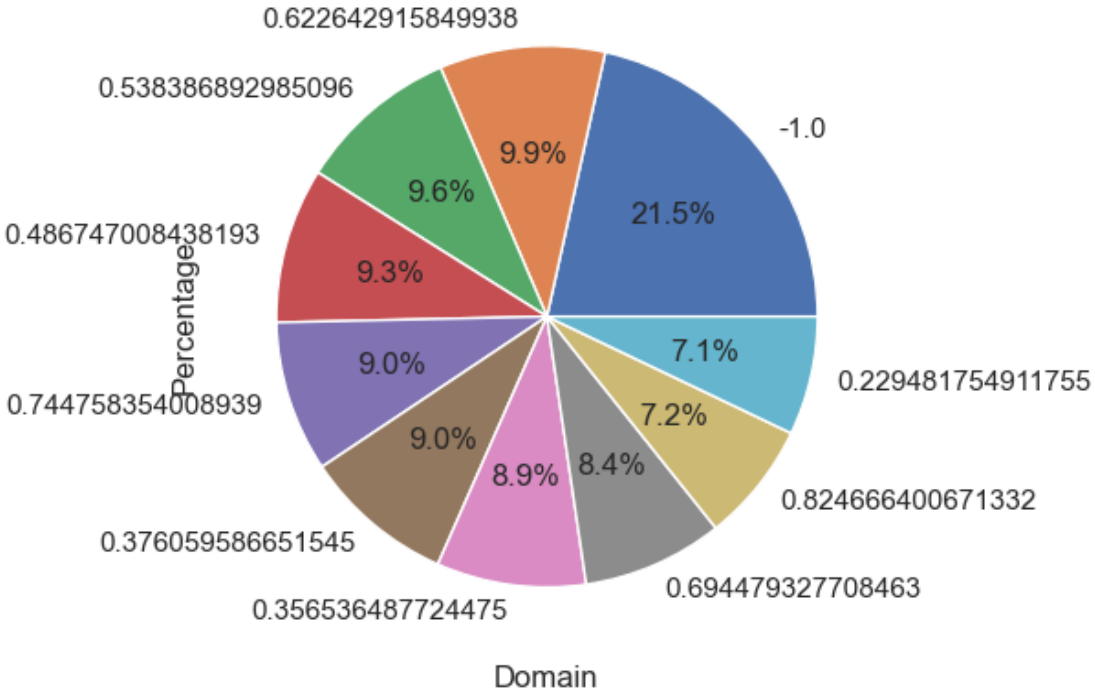
[191]:
```
# Pieplot for the Degree College State of dataset

import matplotlib.pyplot as plt

x = df_data['CollegeState'].value_counts().head(3)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')

plt.xlabel('College State')
plt.ylabel('Percentage')
plt.show()
```



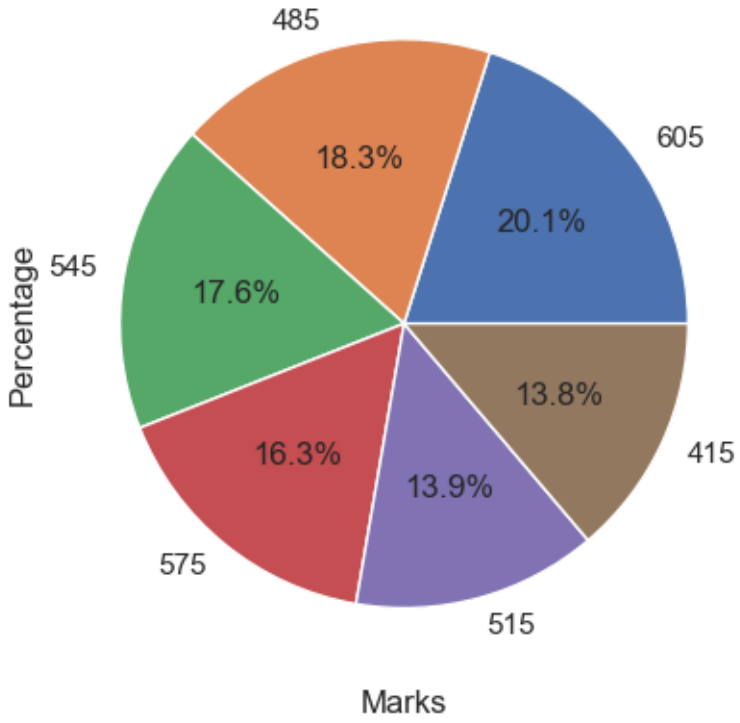
[192]:
```
# Pieplot for the Degree Graduation Year of dataset

import matplotlib.pyplot as plt

x = df_data['GraduationYear'].value_counts().head(7)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')

plt.xlabel('Graduation Year')
plt.ylabel('Percentage')
plt.show()
```



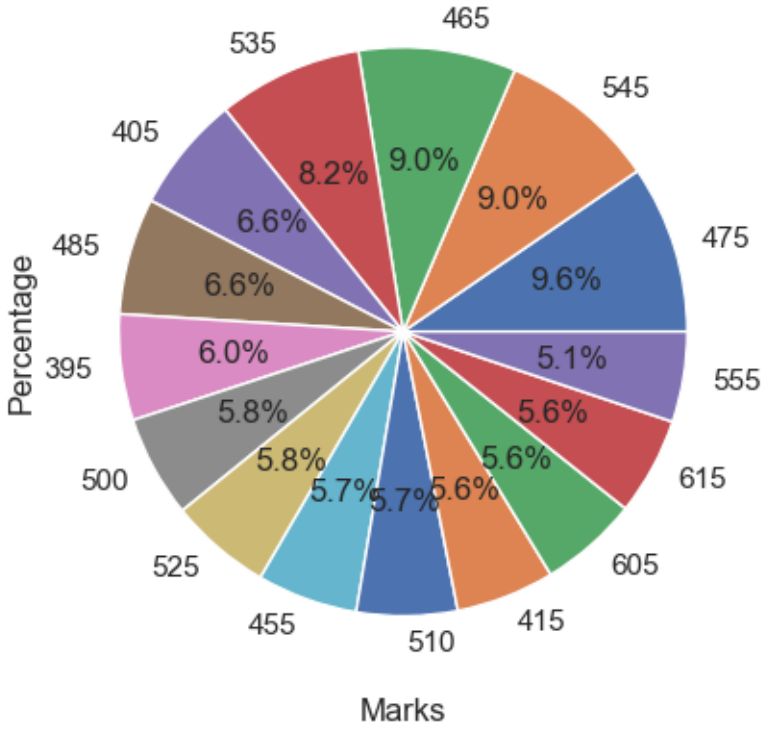
[193]:
```
# Pieplot for the Domian column

import matplotlib.pyplot as plt

x = df_data['Domain'].value_counts().head(10)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Domain')
plt.ylabel('Percentage')
plt.show()
```



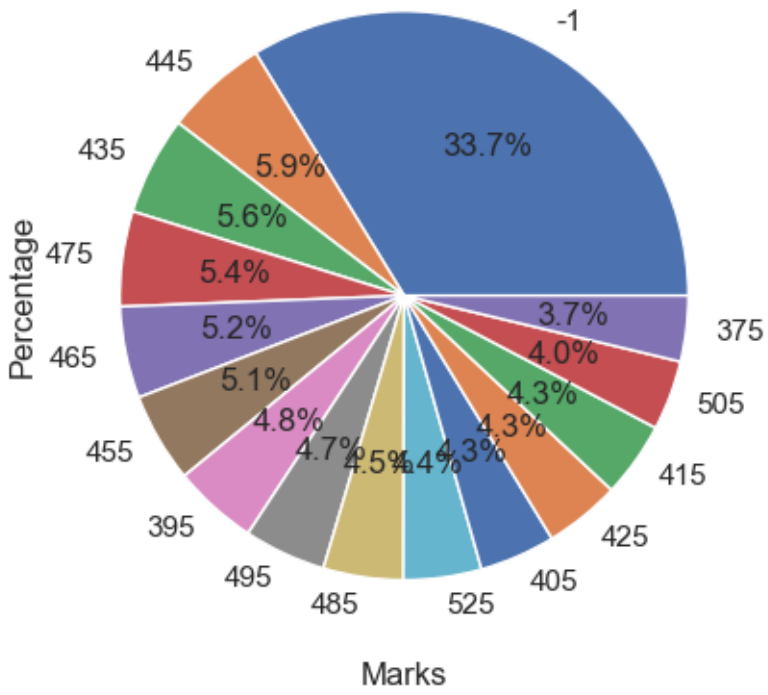
[194]:
```
# Pieplot for the Degree Graduation Year of dataset

import matplotlib.pyplot as plt

x = df_data['Quant'].value_counts().head(6)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Marks')
plt.ylabel('Percentage')
plt.show()
```



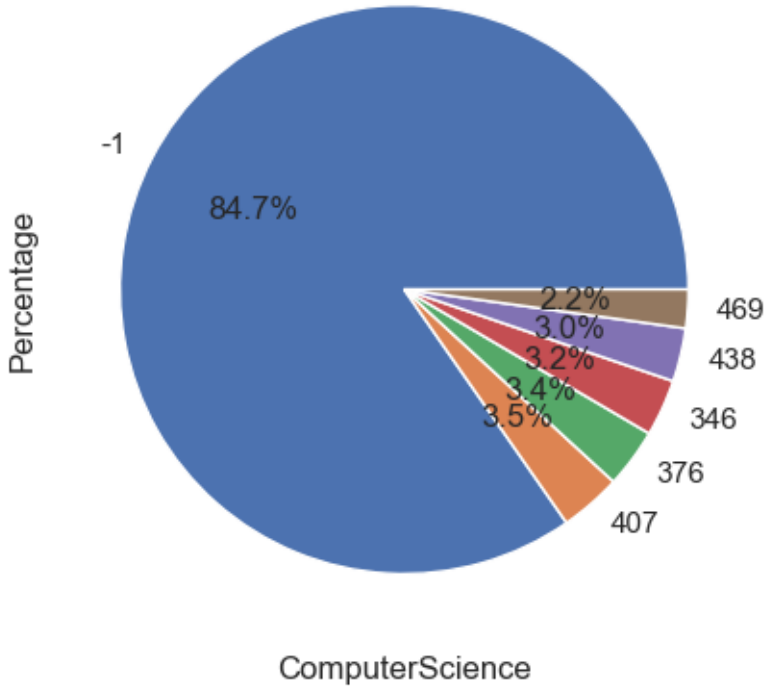
[195]:
```
# Pieplot for the English

import matplotlib.pyplot as plt

x = df_data['English'].value_counts().head(15)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Marks')
plt.ylabel('Percentage')
plt.show()
```



[196]:
```
# Pieplot for the Computer Programming

import matplotlib.pyplot as plt

x = df_data['ComputerProgramming'].value_counts().head(15)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('Marks')
plt.ylabel('Percentage')
plt.show()
```



[197]:
```
# Pieplot for the Computer Science

import matplotlib.pyplot as plt

x = df_data['ComputerScience'].value_counts().head(6)
plt.pie(x.values,
        labels=x.index,
        autopct='%1.1f%%')
plt.xlabel('ComputerScience')
plt.ylabel('Percentage')
plt.show()
```

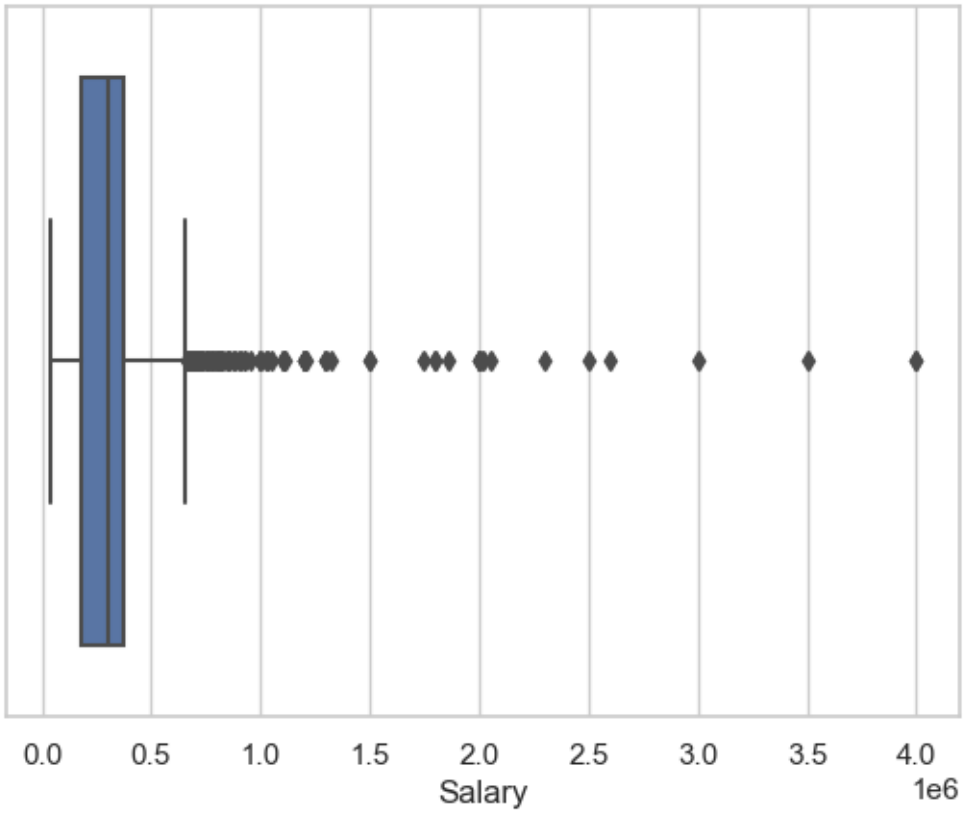


[198]:
```
df_data['Designation'][(df_data['Salary']) > 2000000]
```

[198]:
```
500          application developer
779      assistant system engineer
1237      senior software engineer
1239            operations analyst
1517     software engineer trainee
1759                    programmer
1982       software test engineer
2030           software developer
2182          automation engineer
Name: Designation, dtype: object
```

[199]:
```
sns.boxplot(x=df_data["Salary"])
```

[199]:
```
<Axes: xlabel='Salary'>
```



[200]:
```
df_data['Salary'].median()
```

[200]:
```
300000.0
```

[201]:
```
df_data[df_data["Salary"] > 500000]
```

[201]:
```
     Unnamed: 0      ID   Salary        DOJ                 DOL  \
3          train  267447  1100000 2011-07-01             present
8          train  552574   600000 2013-07-01             present
10         train   87291   600000 2011-04-01  2015-04-01 00:00:00
30         train  192703   530000 2011-12-01             present
57         train  294700   525000 2012-03-01             present
…          …      …       …          …                  …
3946       train   48419   650000 2013-08-01  2015-04-01 00:00:00
3961       train  230702   700000 2011-07-01  2014-09-01 00:00:00
3979       train  212055   550000 2013-07-01  2014-04-01 00:00:00
3991       train  230873   630000 2011-07-01  2014-10-01 00:00:00
3992       train  344407   800000 2014-04-01  2015-04-01 00:00:00

                    Designation    JobCity Gender        DOB  10percentage  \
3     senior software engineer    Gurgaon      m 1989-12-05         85.60
```

```
8          electrical engineer      Noida      m 1991-09-17         90.00
10        senior php developer  Bangalore      m 1989-06-24         88.60
30            systems engineer  Hyderabad      m 1989-10-04         84.00
57               test engineer  Hyderabad      f 1989-08-15         89.88
…                          …       …    …        …           …
3946  senior software engineer  Bangalore      m 1986-04-03         73.00
3961         planning engineer    Rajpura      m 1987-12-27         84.20
3979         software engineer  Bangalore      m 1989-07-22         69.16
3991           systems analyst  Bangalore      m 1990-05-20         80.00
3992                   manager     Rajkot      m 1990-06-22         73.00

      … ComputerScience  MechanicalEngg  ElectricalEngg TelecomEngg  \
3     …              -1              -1              -1          -1
8     …              -1              -1              -1          -1
10    …              -1              -1              -1          -1
30    …              -1              -1              -1          -1
57    …              -1              -1              -1         420
…   …              …               …               …          …
3946  …              -1              -1              -1          -1
3961  …              -1              -1              -1          -1
3979  …              -1              -1              -1          -1
3991  …              -1              -1              -1         393
3992  …              -1              -1              -1          -1

      CivilEngg  conscientiousness agreeableness extraversion  nueroticism  \
3            -1             0.0464        0.3448      -0.3440     -0.40780
8            -1            -0.1590        0.5454      -0.6048     -0.74150
10           -1             0.0464        0.8128       0.0914      0.17980
30           -1            -0.4173        0.6568      -0.0537     -1.23030
57           -1             0.3555       -0.2793      -0.6343     -0.29020
…         …                 …             …           …          …
3946         -1            -1.0355        0.6568       0.0914      0.41480
3961        460            -1.3447        0.0328      -2.3759     -0.99530
3979         -1            -0.5719        0.5008      -0.4891      0.41480
3991         -1            -1.3447        0.5008      -1.6502     -0.05520
3992        480             0.3555       -0.9033       0.9623      0.64983

      openess_to_experience
3                   -0.9194
8                   -0.2859
10                   0.0284
30                  -0.9194
57                  -0.2875
…                     …
3946                 0.5024
3961                 0.3444
3979                -1.2354
```

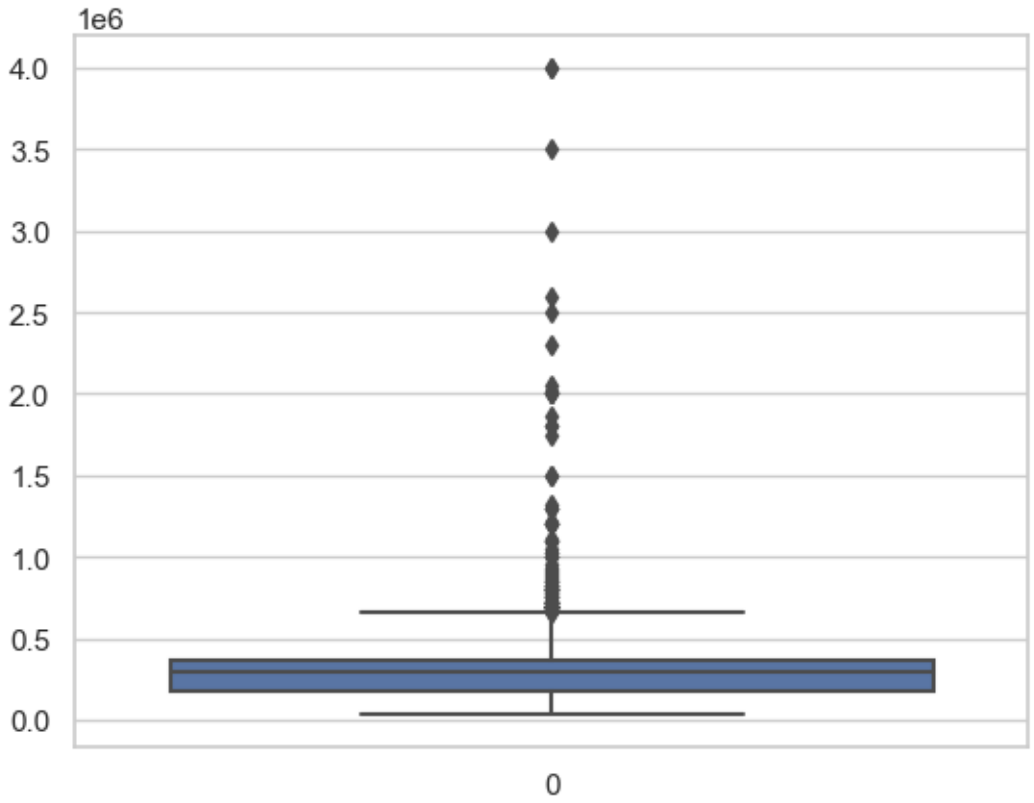
```
3991                 0.0284
3992                -0.4229

[315 rows x 39 columns]
```

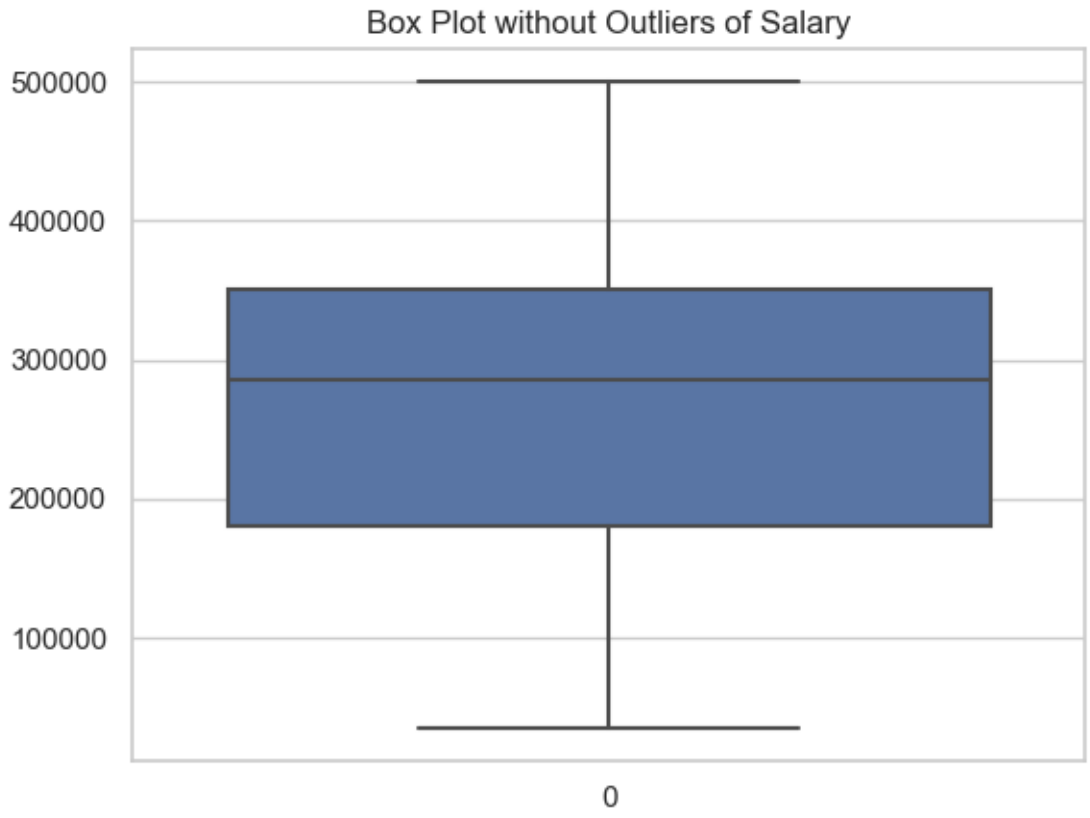
[202]:
```
df_data[df_data["Salary"] > 500000]
```

[202]:
```
     Unnamed: 0      ID   Salary        DOJ                 DOL  \
3         train  267447  1100000 2011-07-01              present
8         train  552574   600000 2013-07-01              present
10        train   87291   600000 2011-04-01  2015-04-01 00:00:00
30        train  192703   530000 2011-12-01              present
57        train  294700   525000 2012-03-01              present
…           …       …        …          …                   …
3946      train   48419   650000 2013-08-01  2015-04-01 00:00:00
3961      train  230702   700000 2011-07-01  2014-09-01 00:00:00
3979      train  212055   550000 2013-07-01  2014-04-01 00:00:00
3991      train  230873   630000 2011-07-01  2014-10-01 00:00:00
3992      train  344407   800000 2014-04-01  2015-04-01 00:00:00

                   Designation    JobCity Gender        DOB  10percentage  \
3     senior software engineer    Gurgaon      m 1989-12-05         85.60
8          electrical engineer      Noida      m 1991-09-17         90.00
10        senior php developer  Bangalore      m 1989-06-24         88.60
30            systems engineer  Hyderabad      m 1989-10-04         84.00
57               test engineer  Hyderabad      f 1989-08-15         89.88
…                          …        …    …         …           …
3946  senior software engineer  Bangalore      m 1986-04-03         73.00
3961         planning engineer    Rajpura      m 1987-12-27         84.20
3979         software engineer  Bangalore      m 1989-07-22         69.16
3991           systems analyst  Bangalore      m 1990-05-20         80.00
3992                   manager     Rajkot      m 1990-06-22         73.00

      … ComputerScience  MechanicalEngg  ElectricalEngg TelecomEngg  \
3     …               -1              -1              -1          -1
8     …               -1              -1              -1          -1
10    …               -1              -1              -1          -1
30    …               -1              -1              -1          -1
57    …               -1              -1              -1         420
…   …               …               …               …          …
3946  …               -1              -1              -1          -1
3961  …               -1              -1              -1          -1
3979  …               -1              -1              -1          -1
3991  …               -1              -1              -1         393
3992  …               -1              -1              -1          -1

      CivilEngg  conscientiousness agreeableness extraversion  nueroticism  \
```

```
3            -1             0.0464        0.3448      -0.3440     -0.40780
8            -1            -0.1590        0.5454      -0.6048     -0.74150
10           -1             0.0464        0.8128       0.0914      0.17980
30           -1            -0.4173        0.6568      -0.0537     -1.23030
57           -1             0.3555       -0.2793      -0.6343     -0.29020
…          …                …             …           …           …
3946         -1            -1.0355        0.6568       0.0914      0.41480
3961        460            -1.3447        0.0328      -2.3759     -0.99530
3979         -1            -0.5719        0.5008      -0.4891      0.41480
3991         -1            -1.3447        0.5008      -1.6502     -0.05520
3992        480             0.3555       -0.9033       0.9623      0.64983

      openess_to_experience
3                   -0.9194
8                   -0.2859
10                   0.0284
30                  -0.9194
57                  -0.2875
…                     …
3946                 0.5024
3961                 0.3444
3979                -1.2354
3991                 0.0284
3992                -0.4229

[315 rows x 39 columns]
```

[203]:
```
import matplotlib.pyplot as plt
def removal_box_plot(df, column, threshold):
    sns.boxplot(df[column])
    plt.show()

    removed_outliers = df[df[column] <= threshold]

    sns.boxplot(removed_outliers[column])
    plt.title(f'Box Plot without Outliers of {column}')
    plt.show()
    return removed_outliers


threshold_value = 5000000

no_outliers = removal_box_plot(df_data, 'Salary', 500000)
```

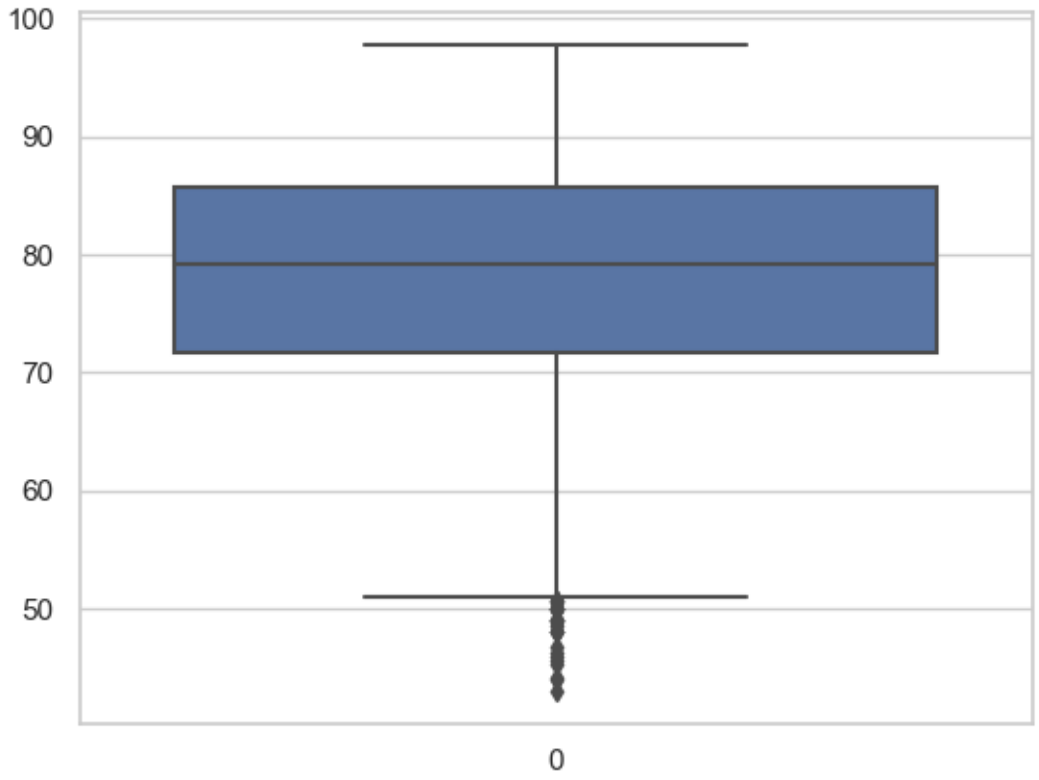

1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0

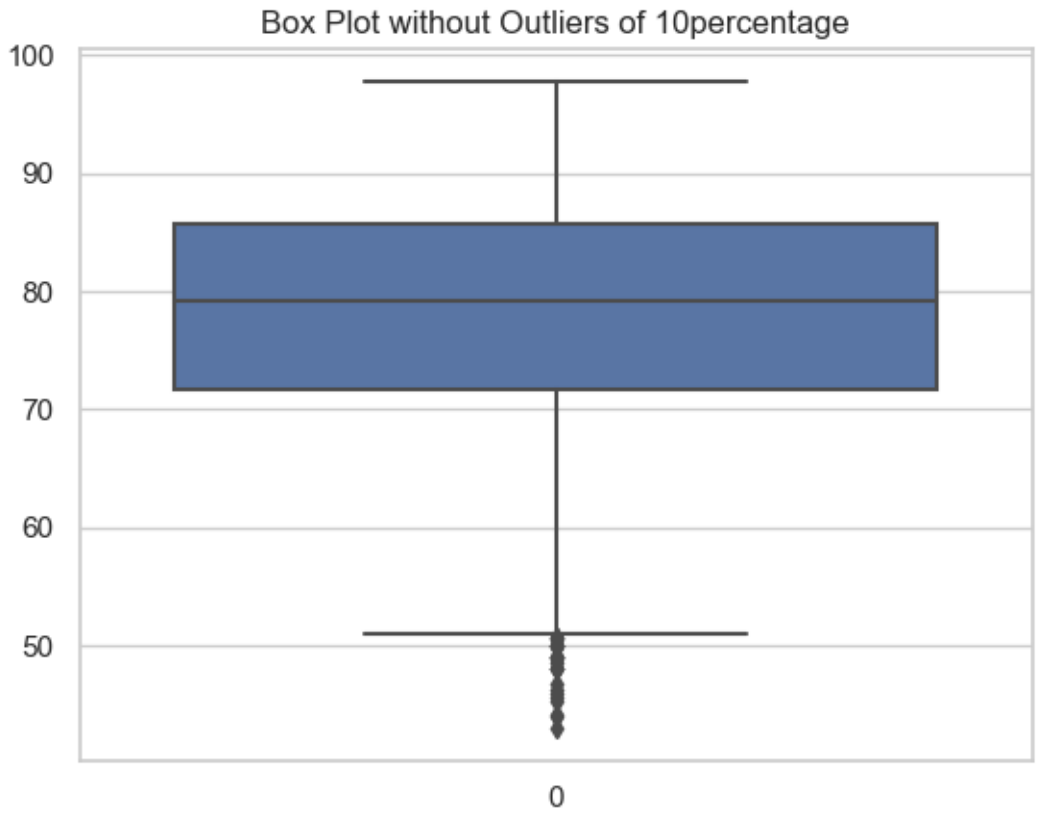


[204]:
```
df_data['Salary'].median()
```

[204]:
```
300000.0
```

[205]:
```
import matplotlib.pyplot as plt
def removal_box_plot(df, column, threshold):
    sns.boxplot(df[column])
    plt.show()

    removed_outliers = df[df[column] <= threshold]

    sns.boxplot(removed_outliers[column])
    plt.title(f'Box Plot without Outliers of {column}')
    plt.show()
    return removed_outliers


threshold_value = 5000000

no_outliers = removal_box_plot(df_data, '10percentage', 500000)
```


100
90
80
70
60
50
0



[206]:
```
df_data['10percentage'].mean()
```

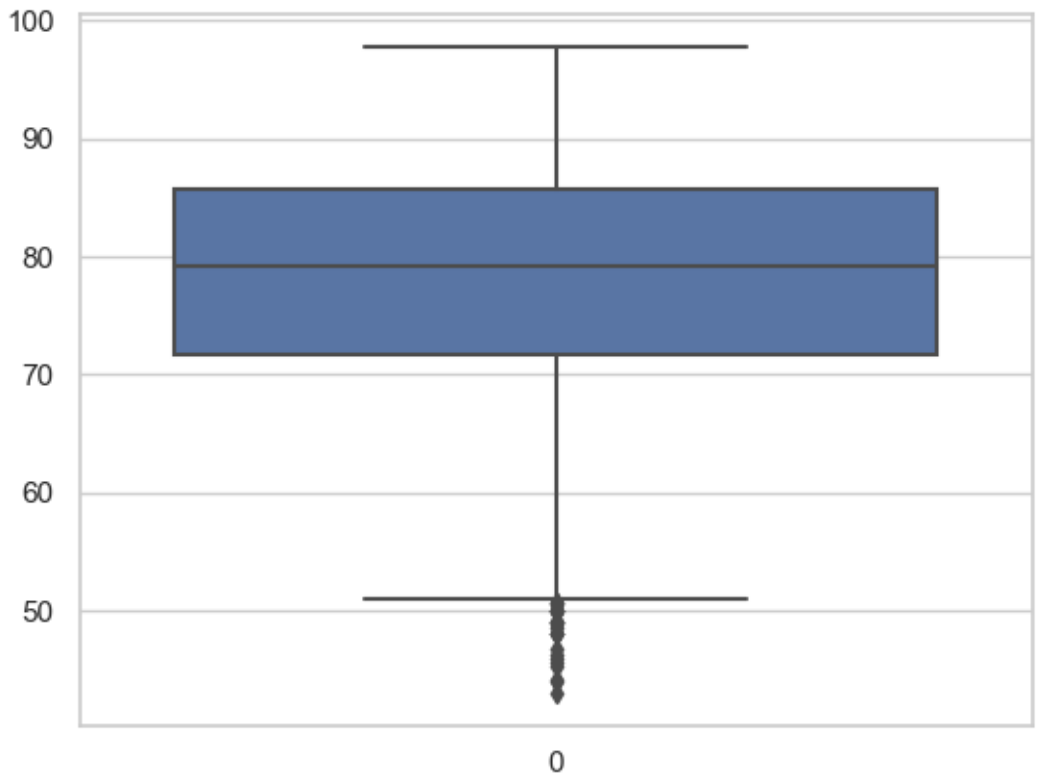
[206]: 77.9254427213607

[207]:
```
df_data['10percentage'].max()
```

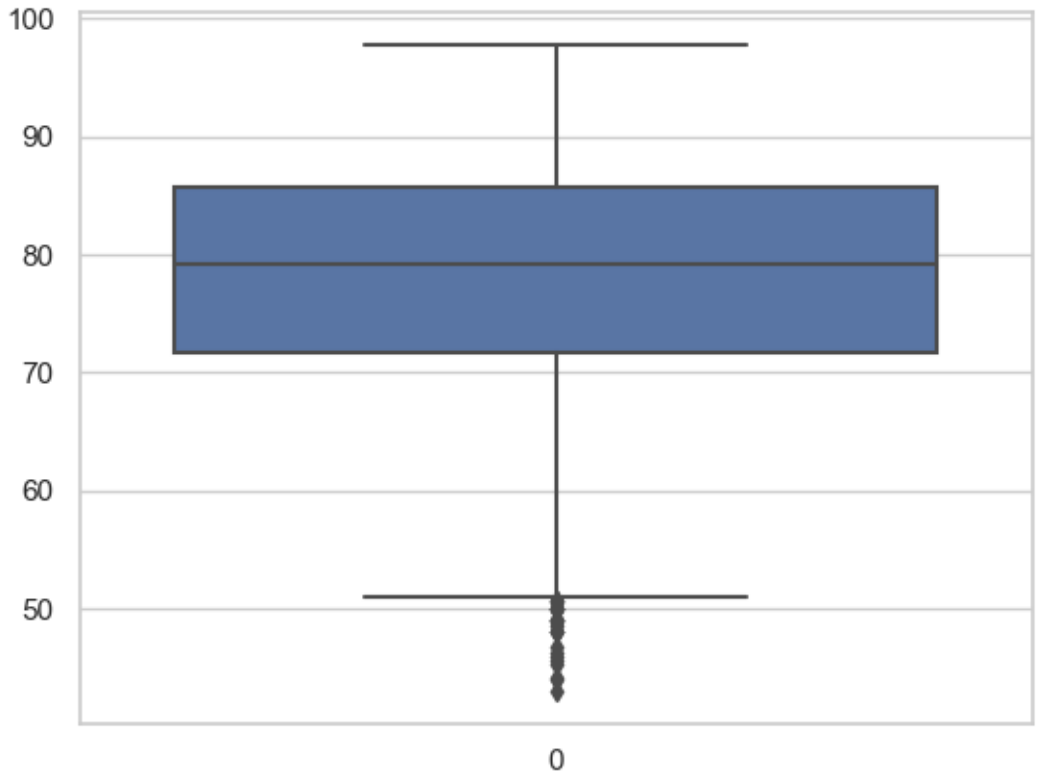
[207]: 97.76

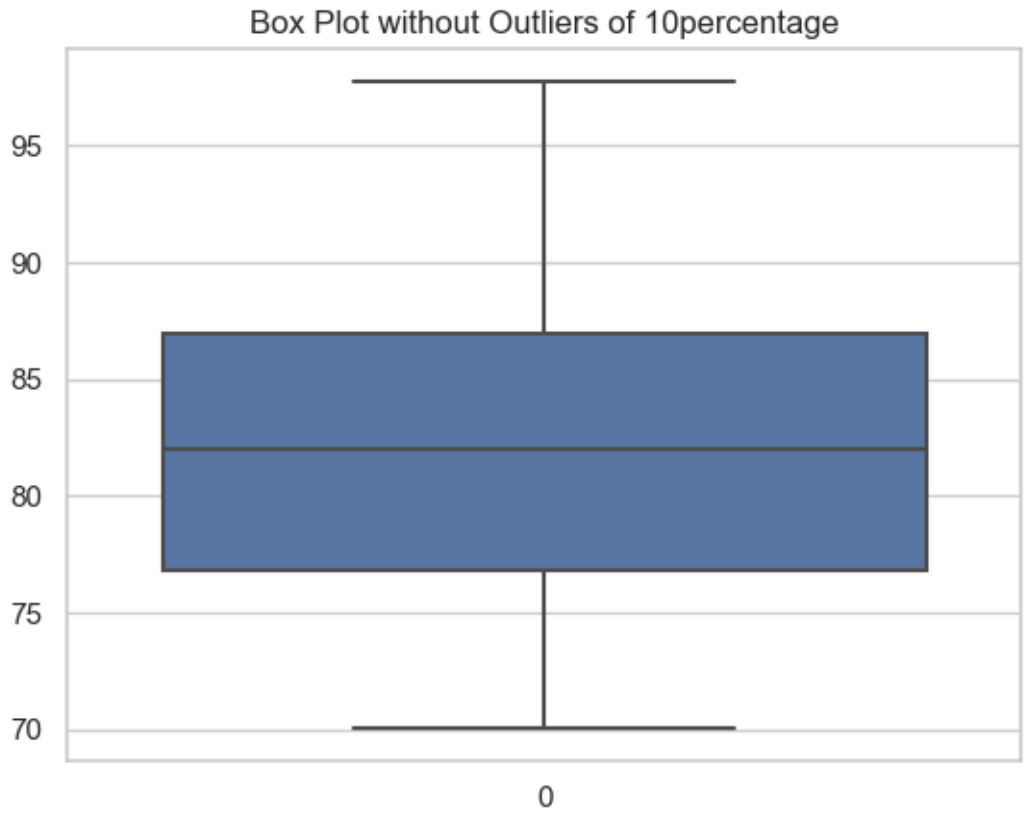
[208]:
```
sns.set(style='whitegrid')
sns.boxplot(df_data['10percentage'])
```

[208]: <Axes: >

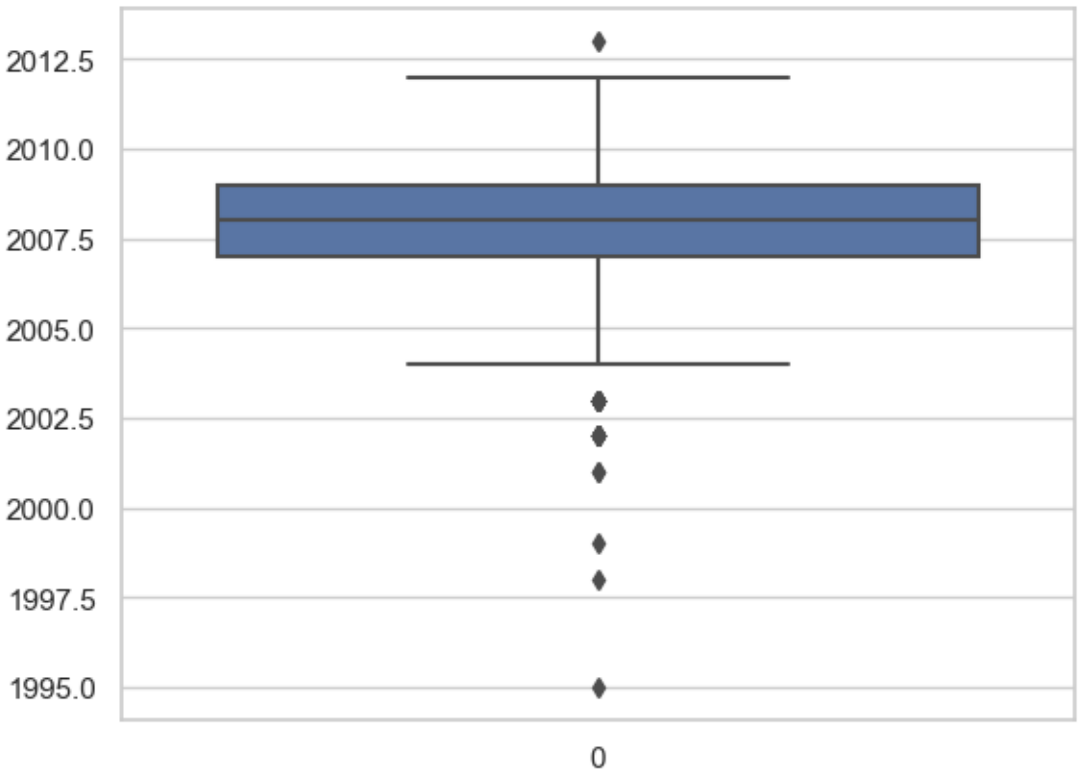


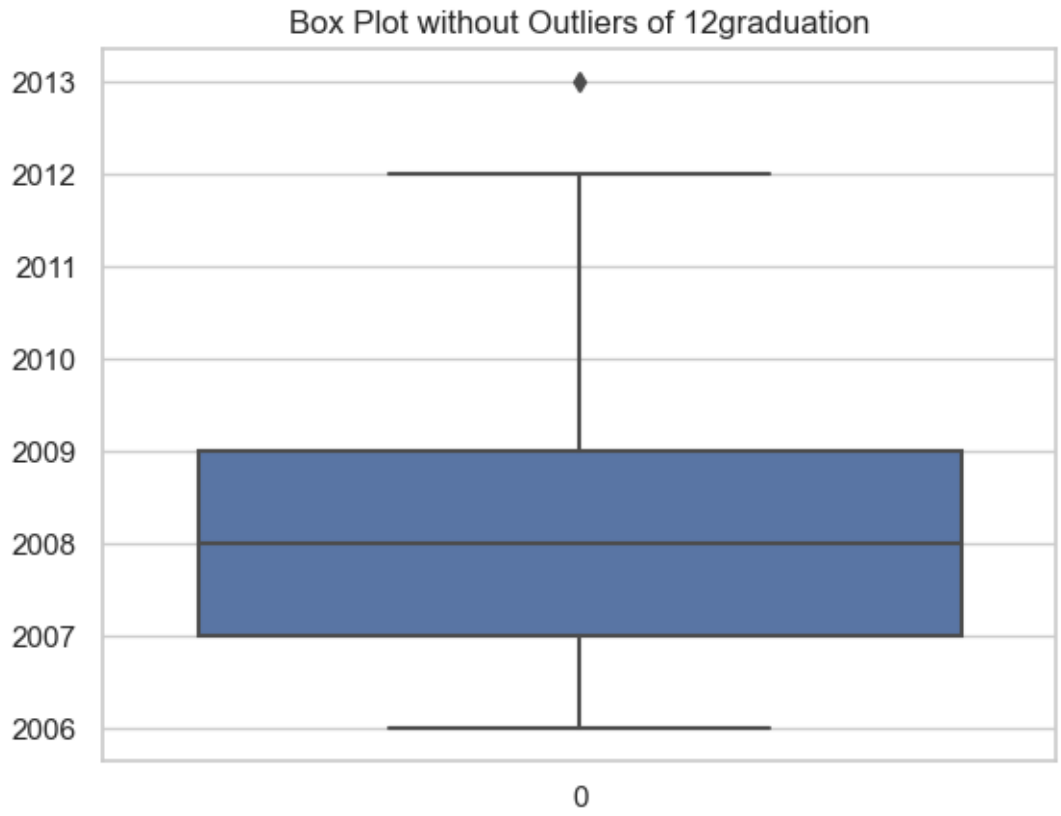
[209]:
```
import matplotlib.pyplot as plt
def removal_box_plot(df, column, threshold):
    sns.boxplot(df[column])
    plt.show()

    removed_outliers = df[df[column] > threshold]

    sns.boxplot(removed_outliers[column])
    plt.title(f'Box Plot without Outliers of {column}')
    plt.show()
    return removed_outliers


threshold_value = 70

no_outliers = removal_box_plot(df_data, '10percentage', 70)
```


100
90
80
70
60
50
0



[210]:
```
import matplotlib.pyplot as plt
def removal_box_plot(df, column, threshold):
    sns.boxplot(df[column])
    plt.show()

    removed_outliers = df[df[column] > threshold]

    sns.boxplot(removed_outliers[column])
    plt.title(f'Box Plot without Outliers of {column}')
    plt.show()
    return removed_outliers


threshold_value = 2005

no_outliers = removal_box_plot(df_data, '12graduation', 2005)
```


2012.5
2010.0
2007.5
2005.0
2002.5
2000.0
1997.5
1995.0
0



[211]:
```
df_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
```