IMPORTING USEFUL LIBRARIES

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
```

EXTRACTING DATA

In [2]:

```
df=pd.read_csv(r"C:\Users\mitra\Desktop\INNOMATICS(MITRABHANU PANDA)\INTERNSHIP\PROJECTS\
1ST PROJECT\data.xlsx - Sheet1.csv")
```

In [3]:

```
df
```

Out[3]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	Con
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.30	...	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.40	...	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.00	...	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.60	...	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.00	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	...	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	...	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	...	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	...	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	...	

3998 rows × 39 columns

KNOW ABOUT THE DATASET

To see first 5 rows of the dataset

To see first 5 rows of the dataset

In [4]:

```
df.head()
```

Out[4]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScien
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	...	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	...	
2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	...	
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	...	
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	...	

5 rows × 39 columns

To see last 5 rows of the dataset

In [5]:

```
df.tail()
```

Out[5]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	Com
3993	train	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	...	
3994	train	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	...	
3995	train	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	...	
3996	train	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	...	
3997	train	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	...	

5 rows × 39 columns

To know about the shape of the database i.e. rows,columns

In [6]:

```
df.shape
```

Out[6]:

```
(3998, 39)
```

To know all the column's name of the datasets.

In [7]:

```
df.columns
```

Out[7]:

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

To know about the column's name with thier data type, how many null value present inside each columns, how many rows in this dataset, how many memory usegae

In [8]:

```
df.info()
```

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

To know about how many null values present inside each columns

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
Unnamed: 0               0
ID                       0
Salary                   0
DOJ                      0
DOL                      0
Designation              0
JobCity                  0
Gender                   0
DOB                      0
10percentage             0
10board                  0
12graduation             0
12percentage             0
12board                  0
CollegeID                0
CollegeTier              0
Degree                   0
Specialization           0
collegeGPA               0
CollegeCityID            0
CollegeCityTier          0
CollegeState             0
GraduationYear           0
English                  0
Logical                  0
Quant                    0
Domain                   0
ComputerProgramming      0
ElectronicsAndSemicon    0
ComputerScience          0
MechanicalEngg           0
ElectricalEngg           0
TelecomEngg              0
CivilEngg                0
conscientiousness        0
agreeableness            0
extraversion             0
nueroticism              0
openess_to_experience    0
dtype: int64
```

To know about Total null values present inside a dataset

In [10]:

```
df.isnull().sum().sum()
```

Out[10]:

```
0
```

In [11]:

```
df.columns
```

Out[11]:

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

In [12]:

```
# Delete the first column i.e. 'Unnamed: 0' because this column is not required for our a
nalysis
df.drop(["Unnamed: 0"],axis=1,inplace=True)
```

In [13]:

```
df
```

Out[13]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	C
0	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.30	board ofsecondary education,ap	...	
1	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.40	cbse	...	
2	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.00	cbse	...	
3	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.60	cbse	...	
4	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.00	cbse	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	47916	280000.0	10/1/11 0:00	10/1/12 0:00	software engineer	New Delhi	m	4/15/87 0:00	52.09	cbse	...	
3994	752781	100000.0	7/1/13 0:00	7/1/13 0:00	technical writer	Hyderabad	f	8/27/92 0:00	90.00	state board	...	
3995	355888	320000.0	7/1/13 0:00	present	associate software engineer	Bangalore	m	7/3/91 0:00	81.86	bse,odisha	...	
3996	947111	200000.0	7/1/14 0:00	1/1/15 0:00	software developer	Asifabadbanglore	f	3/20/92 0:00	78.72	state board	...	
3997	324966	400000.0	2/1/13 0:00	present	senior systems engineer	Chennai	f	2/26/91 0:00	70.60	cbse	...	

3998 rows × 38 columns

In [14]:

```
# In this dataset DOJ & DOB is in object but it should be in Date time format
df["DOJ"]=pd.to_datetime(df["DOJ"])
```

In [15]:

```
df["DOB"]=pd.to_datetime(df["DOB"])
```

In [16]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

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

In [17]:

```
# Convert the "present value" of DOL column to NaN.
df["DOL"]=df["DOL"].apply(lambda x: np.nan  if x=="present" else x)
```

In [18]:

```
df["DOL"]
```

Out[18]:

```
0                 NaN
1                 NaN
2                 NaN
3                 NaN
4         3/1/15 0:00
             ...     
3993     10/1/12 0:00
3994      7/1/13 0:00
3995              NaN
3996      1/1/15 0:00
3997              NaN
Name: DOL, Length: 3998, dtype: object
```

In [19]:

```
# Fill the NaN value to Today's date
# Fill the NaN value to Today's date.
df["DOL"]=df["DOL"].fillna(pd.to_datetime('today').date())
```

In [20]:

```
df["DOL"]
```

Out[20]:

```
0          2024-02-17
1          2024-02-17
2          2024-02-17
3          2024-02-17
4         3/1/15 0:00
              ...
3993    10/1/12 0:00
3994     7/1/13 0:00
3995      2024-02-17
3996     1/1/15 0:00
3997      2024-02-17
Name: DOL, Length: 3998, dtype: object
```

In [21]:

```
# Convert the data type to Datetime
df["DOL"]=pd.to_datetime(df["DOL"])
```

In [22]:

```
df["DOL"]
```

Out[22]:

```
0      2024-02-17
1      2024-02-17
2      2024-02-17
3      2024-02-17
4      2015-03-01
          ...
3993   2012-10-01
3994   2013-07-01
3995   2024-02-17
3996   2015-01-01
3997   2024-02-17
Name: DOL, Length: 3998, dtype: datetime64[ns]
```

In [23]:

```
df
```

Out[23]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Compu
0	203097	420000.0	2012-06-01	2024-02-17	senior quality engineer	Bangalore	f	1990-02-19	84.30	board ofsecondary education,ap	...	
1	579905	500000.0	2013-09-01	2024-02-17	assistant manager	Indore	m	1989-10-04	85.40	cbse	...	
2	810601	325000.0	2014-06-01	2024-02-17	systems engineer	Chennai	f	1992-08-03	85.00	cbse	...	
3	267447	1100000.0	2011-07-01	2024-02-17	senior software engineer	Gurgaon	m	1989-12-05	85.60	cbse	...	
4	343523	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.00	cbse	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	47916	280000.0	2011-10-01	2012-10-01	software engineer	New Delhi	m	1987-04-15	52.09	cbse	...	

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Compu
3994	752781	100000.0	2013-07-01	2013-07-01	technical writer	Hyderabad	f	1992-08-27	[illegible]	state board	...	
3995	355888	320000.0	2013-07-01	2024-02-17	associate software engineer	Bangalore	m	1991-07-03	81.86	bse,odisha	...	
3996	947111	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20	78.72	state board	...	
3997	324966	400000.0	2013-02-01	2024-02-17	senior systems engineer	Chennai	f	1991-02-26	70.60	cbse	...	

3998 rows × 38 columns

To know how many Object data type columns are there in the dataset

In [24]:

```
df.select_dtypes("object")
```

Out[24]:

	Designation	JobCity	Gender	10board	12board	Degree	Specialization	CollegeState
0	senior quality engineer	Bangalore	f	board ofsecondary education,ap	board of intermediate education,ap	B.Tech/B.E.	computer engineering	Andhra Pradesh
1	assistant manager	Indore	m	cbse	cbse	B.Tech/B.E.	electronics and communication engineering	Madhya Pradesh
2	systems engineer	Chennai	f	cbse	cbse	B.Tech/B.E.	information technology	Uttar Pradesh
3	senior software engineer	Gurgaon	m	cbse	cbse	B.Tech/B.E.	computer engineering	Delhi
4	get	Manesar	m	cbse	cbse	B.Tech/B.E.	electronics and communication engineering	Uttar Pradesh
...	...	...	...	...	...	...	...	...
3993	software engineer	New Delhi	m	cbse	cbse	B.Tech/B.E.	information technology	Haryana
3994	technical writer	Hyderabad	f	state board	state board	B.Tech/B.E.	electronics and communication engineering	Telangana
3995	associate software engineer	Bangalore	m	bse,odisha	chse,odisha	B.Tech/B.E.	computer engineering	Orissa
3996	software developer	Asifabadbanglore	f	state board	state board	B.Tech/B.E.	computer science & engineering	Karnataka
3997	senior systems engineer	Chennai	f	cbse	cbse	B.Tech/B.E.	information technology	Tamil Nadu

3998 rows × 8 columns

To know how many numeric data type columns are there in the dataset

In [25]:

```
df.select_dtypes(["int64","float64"])
```

Out[25]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	Colle
0	203097	420000.0	84.30	2007	95.80	1141	2	78.00	1141	
1	579905	500000.0	85.40	2007	85.00	5807	2	70.06	5807	
2	810601	325000.0	85.00	2010	68.20	64	2	70.00	64	
3	267447	1100000.0	85.60	2007	83.60	6920	1	74.64	6920	
4	343523	200000.0	78.00	2008	76.80	11368	2	73.90	11368	
...	...	...	...	...	...	...	...	...	...	
3993	47916	280000.0	52.09	2006	55.50	6268	2	61.50	6268	
3994	752781	100000.0	90.00	2009	93.00	4883	2	77.30	4883	
3995	355888	320000.0	81.86	2008	65.50	9786	2	70.00	9786	
3996	947111	200000.0	78.72	2010	69.88	979	2	70.42	979	
3997	324966	400000.0	70.60	2008	68.00	6609	2	68.00	6609	

3998 rows × 27 columns

To know how many Datetime data type columns are there in the dataset

In [26]:

```
df.select_dtypes("datetime64[ns]")
```

Out[26]:

	DOJ	DOL	DOB
0	2012-06-01	2024-02-17	1990-02-19
1	2013-09-01	2024-02-17	1989-10-04
2	2014-06-01	2024-02-17	1992-08-03
3	2011-07-01	2024-02-17	1989-12-05
4	2014-03-01	2015-03-01	1991-02-27
...	...	...	...
3993	2011-10-01	2012-10-01	1987-04-15
3994	2013-07-01	2013-07-01	1992-08-27
3995	2013-07-01	2024-02-17	1991-07-03
3996	2014-07-01	2015-01-01	1992-03-20
3997	2013-02-01	2024-02-17	1991-02-26

3998 rows × 3 columns

To describe about the Numerical Columns

In [27]:

```
df.describe()
```

Out[27]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	Colle
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	399

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	Colle
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	1.925713	71.486171	515
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	0.262270	8.167338	480
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	1.000000	6.450000	
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	2.000000	66.407500	49
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	2.000000	71.720000	387
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	2.000000	76.327500	881
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	2.000000	99.930000	1840

8 rows × 27 columns

OUTLIER DETECTION & TREATMENT

In [28]:

```
df.select_dtypes(["int64","float64"]).columns
```

Out[28]:

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

In [29]:

```
column=['Salary', '10percentage', '12percentage', 'collegeGPA', 'English', 'Logical', 'Qu
ant','conscientiousness', 'agreeableness', 'extraversion',
       'nueroticism', 'openess_to_experience']
```

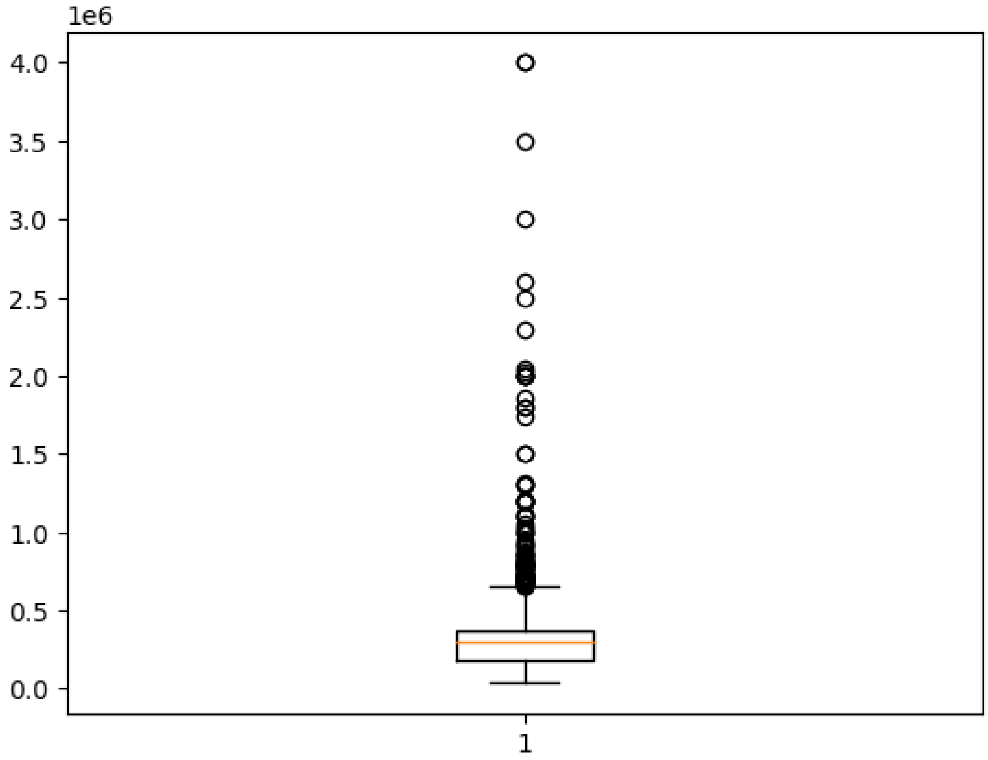
In [30]:

```
def outlier_treatment(dcol):
    for i in dcol:
        print("Describe: ")
        print(df[i].describe())
        print("*****************************************")
        print()
        plt.boxplot(df[i])
        plt.show()
        print()
        print("*****************************************")
        print("Skewness: ",df[i].skew())
        print("*****************************************")
        print()
        q1=df[i].quantile(0.25)
        print("First Quartile: ",q1)
        print()
        q3=df[i].quantile(0.75)
        print("Third Quartile: ",q3)
        print("*****************************************")
        print()
        iqr=q3-q1
        print("InterQuartile Range: ",iqr)
        print()
        print("*****************************************")
        lower=q1-(1.5*iqr)
        print("Lower Limit", lower)
        print()
        upper=q3+(1.5*iqr)
        print("upper Limit", upper)
        print()
        print("****************************************************")
        print("Shape: ")
        print(df[(df[i]<lower) | (df[i]>upper)].shape)
        df[i]=df[i].apply(lambda x :  lower if x<lower  else   upper if x>upper else x)
        print("****************************************")
        print(df[(df[i]<lower) | (df[i]>upper)].shape)
        print()
        print("****************************************")
        print()
        plt.boxplot(df[i])
        plt.show()
        print()
```

In [31]:

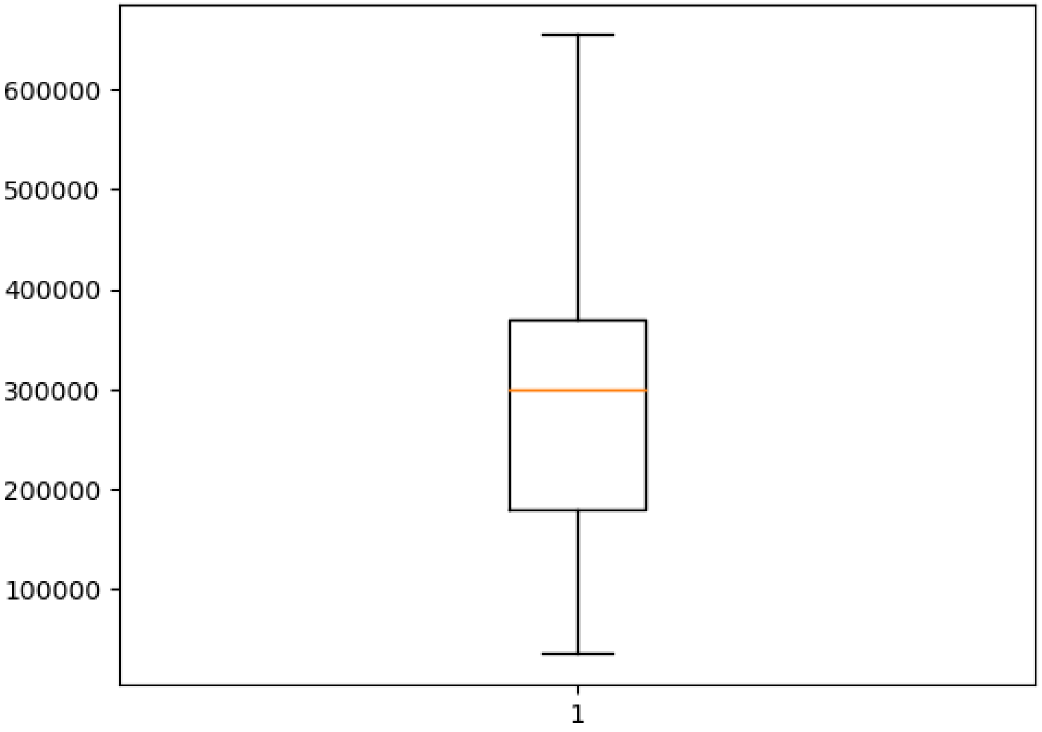
```
outlier_treatment(column)
```

```
Describe:
count    3.998000e+03
mean     3.076998e+05
std      2.127375e+05
min      3.500000e+04
25%      1.800000e+05
50%      3.000000e+05
75%      3.700000e+05
max      4.000000e+06
Name: Salary, dtype: float64
****************************************
```

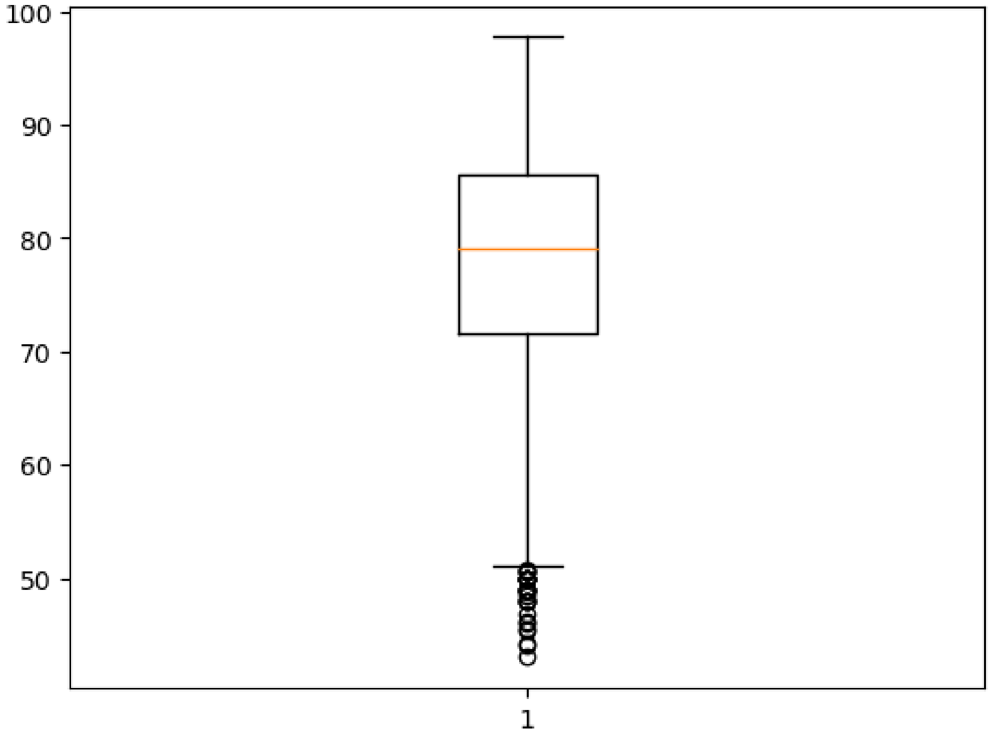


```
****************************************
Skewness:  6.451081166224832
****************************************

First Quartile:  180000.0

Third Quartile:  370000.0
****************************************

InterQuartile Range:  190000.0

****************************************
Lower Limit -105000.0

upper Limit 655000.0
```

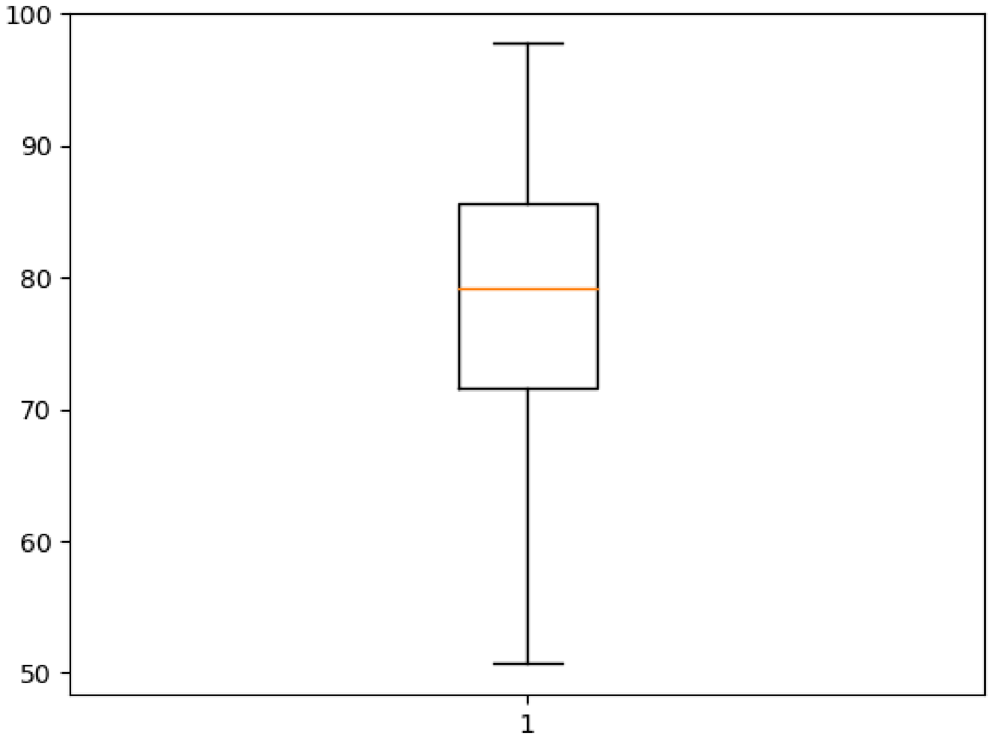
```
***************************************************
Shape:
(109, 38)
****************************************
(0, 38)

****************************************
```



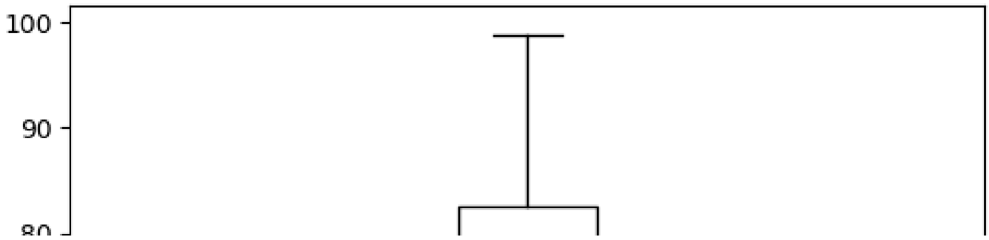
```
Describe:
count    3998.000000
mean       77.925443
std         9.850162
min        43.000000
25%        71.680000
50%        79.150000
75%        85.670000
max        97.760000
Name: 10percentage, dtype: float64
****************************************
```

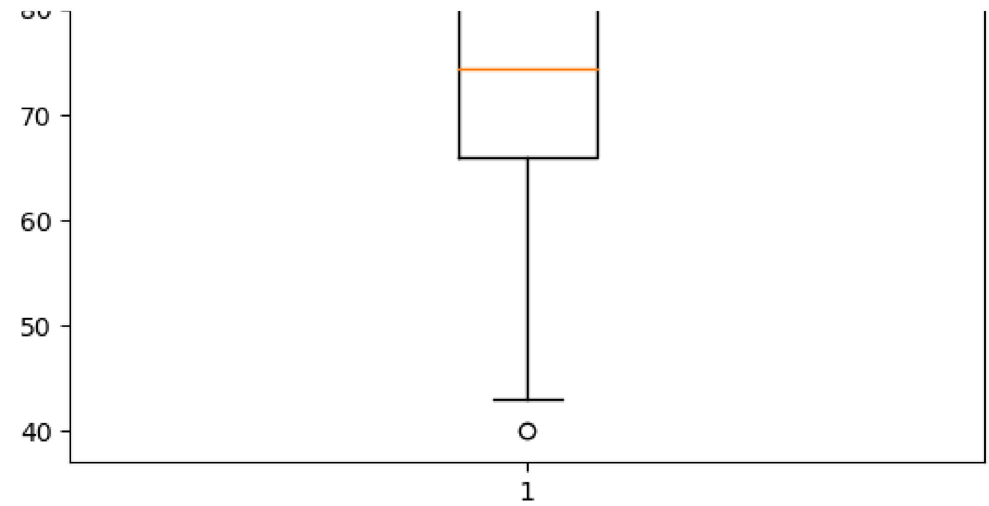


```
****************************************
Skewness:  -0.5910185081648047
****************************************

First Quartile:  71.68

Third Quartile:  85.67
****************************************

InterQuartile Range:  13.989999999999995

****************************************
Lower Limit 50.695000000000014

upper Limit 106.655

***************************************************
Shape:
(30, 38)
****************************************
(0, 38)

****************************************
```

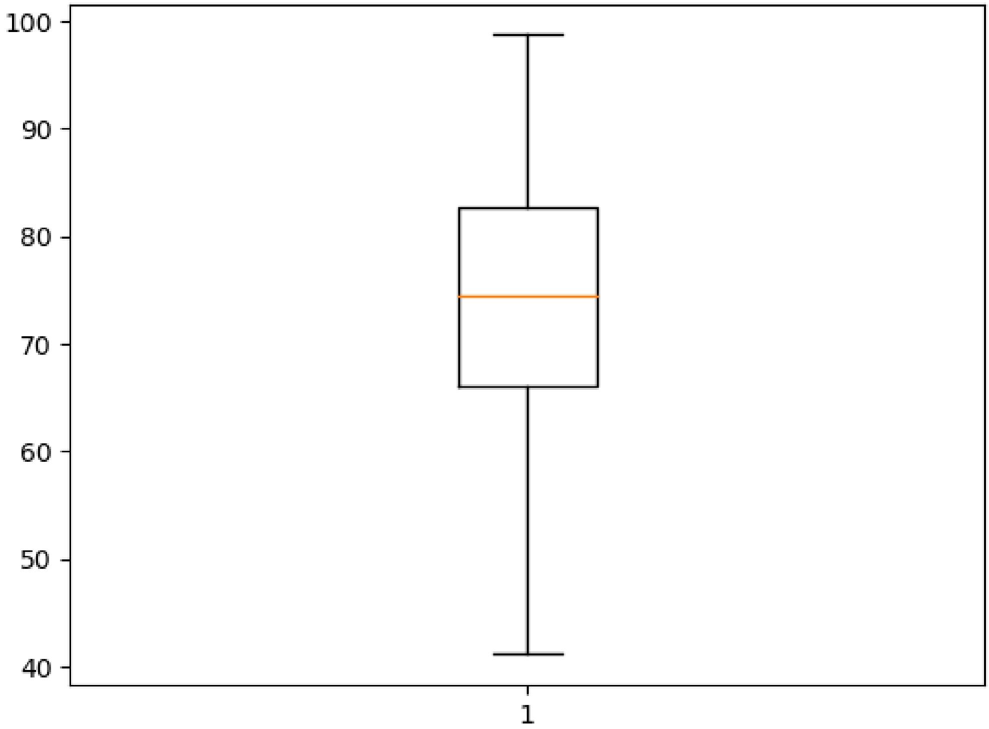


```
Describe:
count    3998.000000
mean       74.466366
std        10.999933
min        40.000000
25%        66.000000
50%        74.400000
75%        82.600000
max        98.700000
Name: 12percentage, dtype: float64
****************************************
```

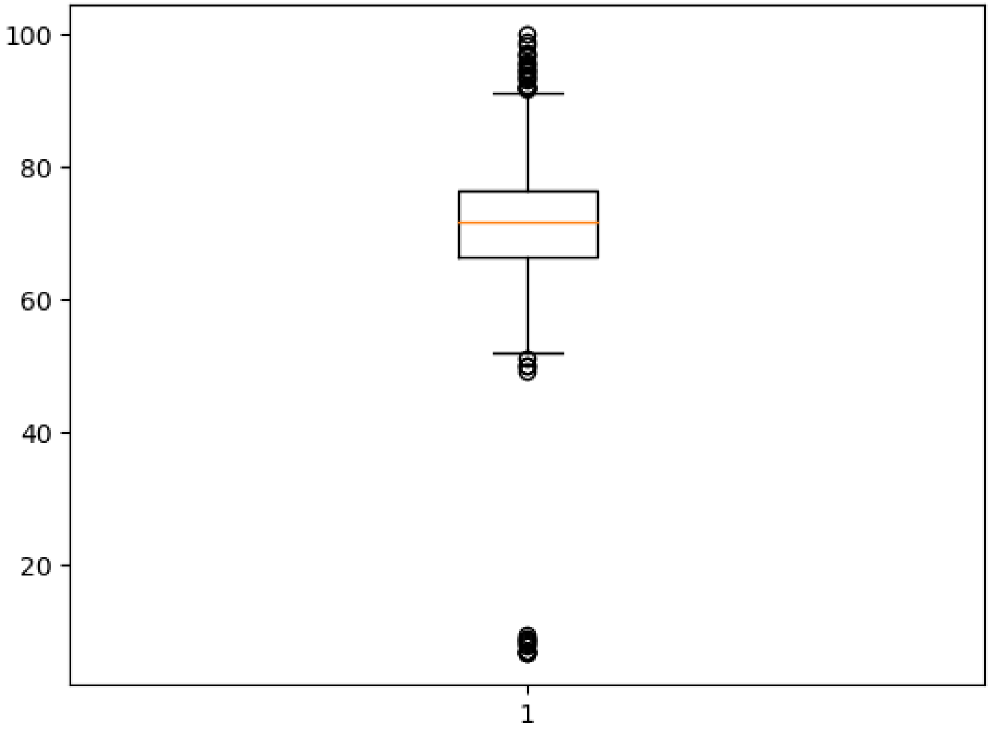




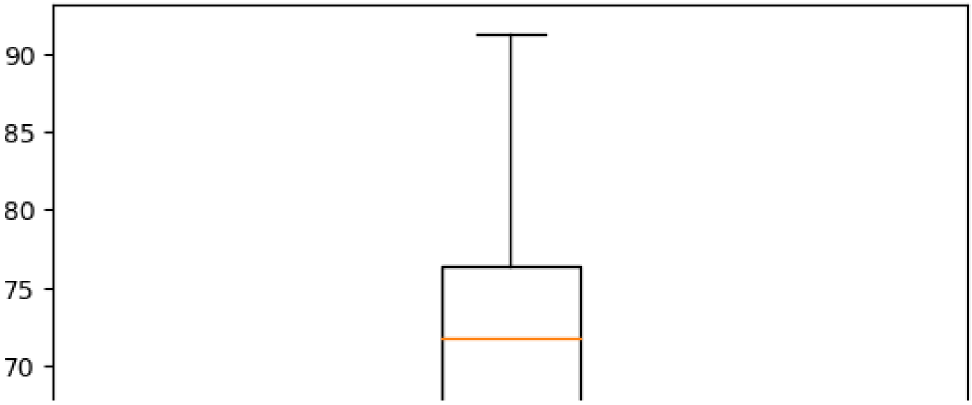
```
****************************************
Skewness:  -0.03260741437482245
****************************************

First Quartile:  66.0

Third Quartile:  82.6
****************************************

InterQuartile Range:  16.599999999999994

****************************************
Lower Limit 41.10000000000001

upper Limit 107.49999999999999

***************************************************
Shape:
(1, 38)
****************************************
(0, 38)

****************************************
```

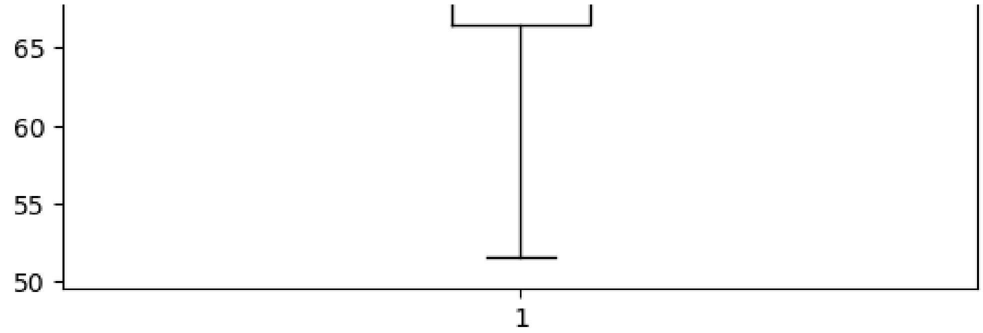


```
Describe:
count    3998.000000
mean       71.486171
std         8.167338
min         6.450000
min         6.450000
25%        66.407500
50%        71.720000
75%        76.327500
max        99.930000
Name: collegeGPA, dtype: float64
****************************************
```

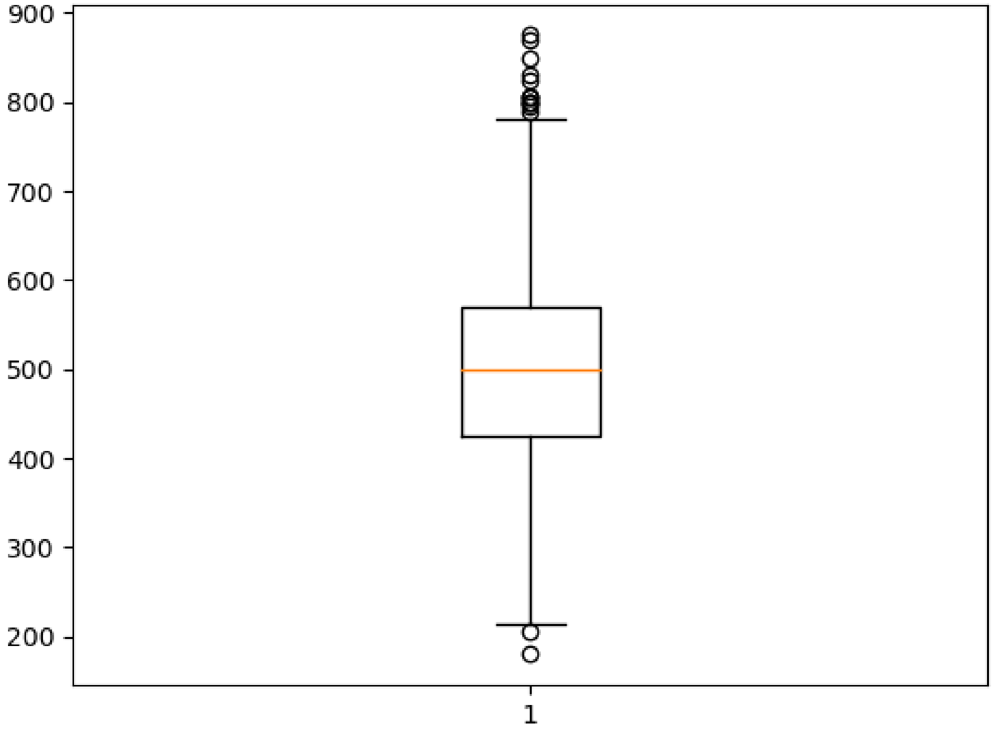


```
****************************************
Skewness:  -1.2492091640381637
****************************************

First Quartile:  66.4075

Third Quartile:  76.3275
****************************************

InterQuartile Range:  9.920000000000002

****************************************
Lower Limit 51.527499999999996

upper Limit 91.20750000000001

***************************************************
Shape:
(38, 38)
****************************************
(0, 38)

****************************************
```



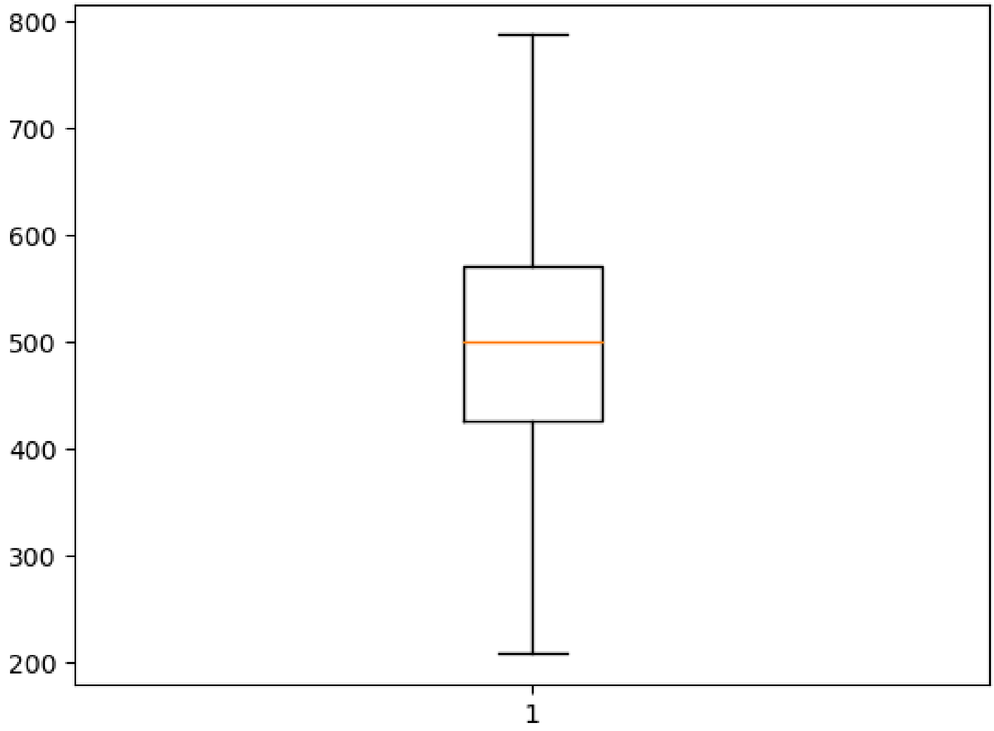


```
Describe:
count    3998.000000
mean      501.649075
std       104.940021
min       180.000000
25%       425.000000
50%       500.000000
75%       570.000000
max       875.000000
Name: English, dtype: float64
****************************************
```

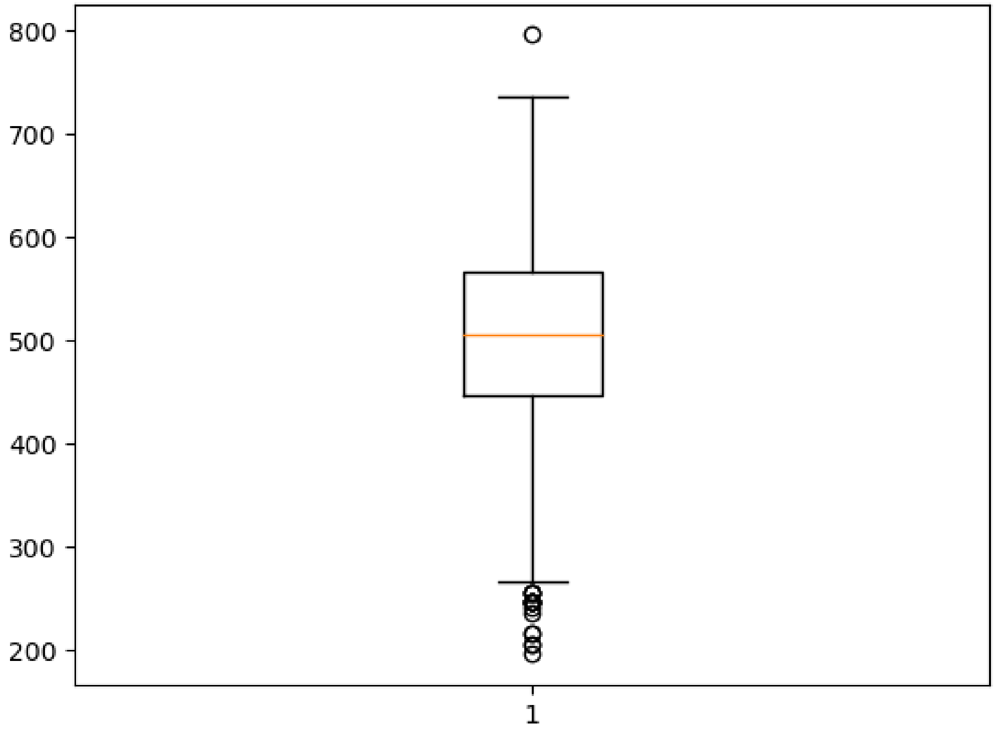


```
****************************************
Skewness:  0.1919970174188361
****************************************

First Quartile:  425.0

Third Quartile:  570.0
****************************************

InterQuartile Range:  145.0

****************************************
Lower Limit 207.5

upper Limit 787.5

**************************************************
Shape:
(15, 38)
****************************************
(0, 38)
```

```
****************************************
```

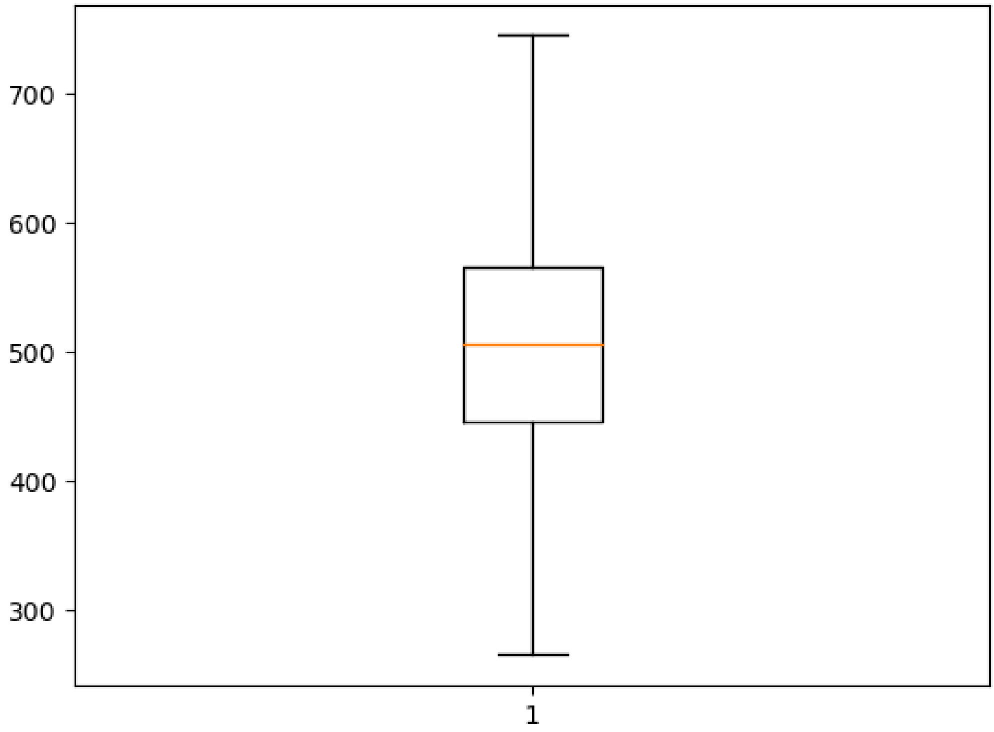


```
Describe:
count    3998.000000
mean      501.598799
std        86.783297
min       195.000000
25%       445.000000
50%       505.000000
75%       565.000000
max       795.000000
Name: Logical, dtype: float64
****************************************
```

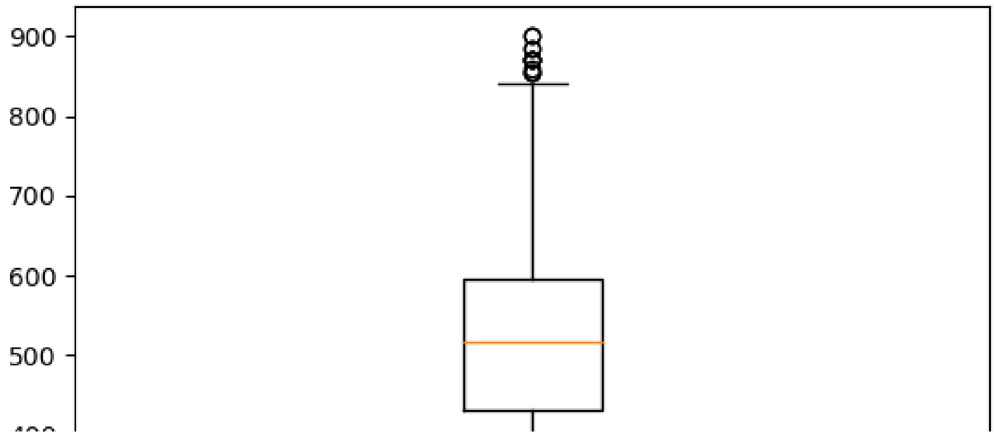


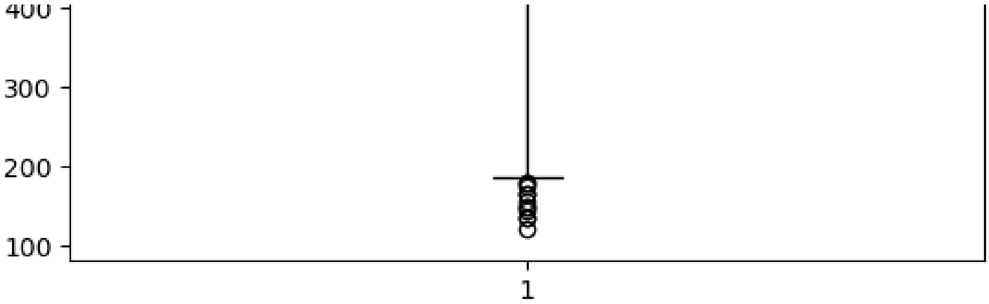
```
****************************************
Skewness:  -0.21660181091305136
****************************************

First Quartile:  445.0
```

```
Third Quartile:  565.0
****************************************

InterQuartile Range:  120.0

****************************************
Lower Limit 265.0

upper Limit 745.0

***************************************************
Shape:
(18, 38)
****************************************
(0, 38)

****************************************
```

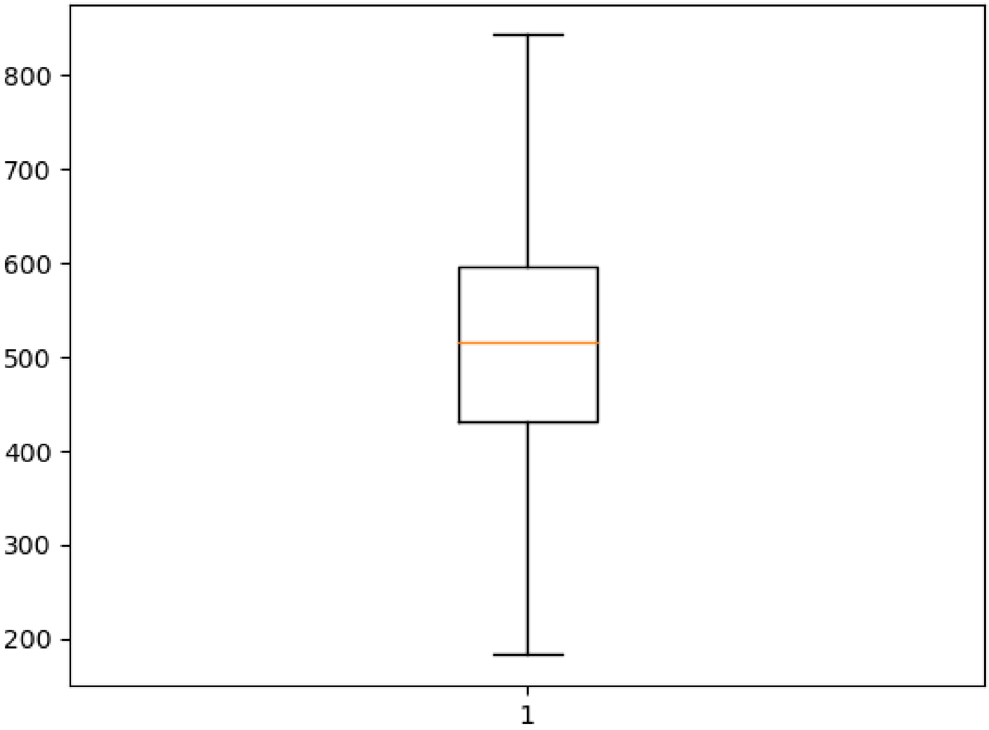


```
Describe:
count    3998.000000
mean      513.378189
std       122.302332
min       120.000000
25%       430.000000
50%       515.000000
75%       595.000000
max       900.000000
Name: Quant, dtype: float64
****************************************
```

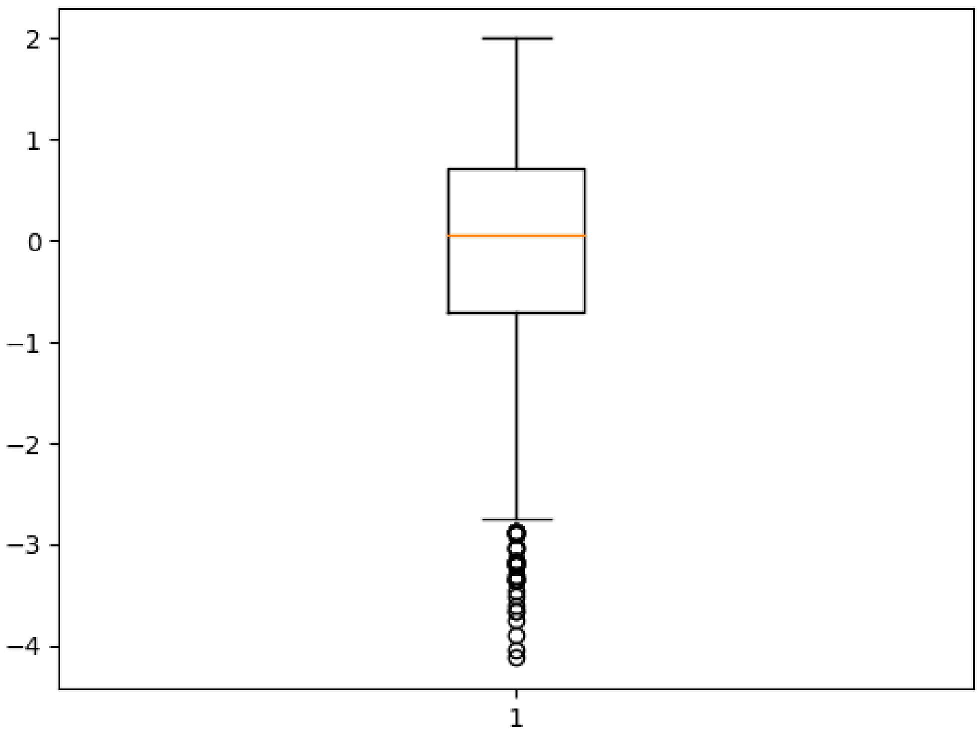




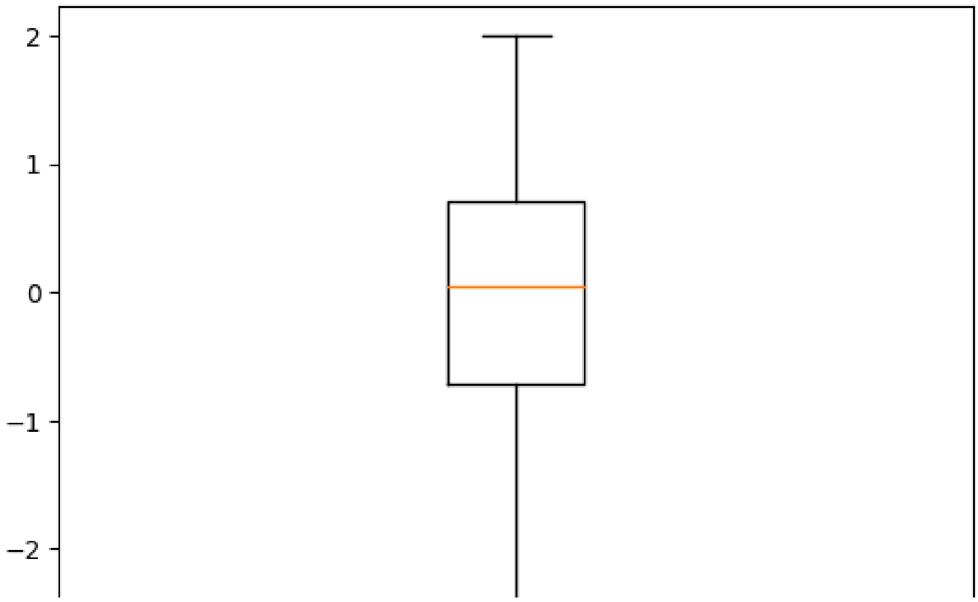
```
****************************************
Skewness:  -0.01939903459277611
****************************************

First Quartile:  430.0

Third Quartile:  595.0
****************************************

InterQuartile Range:  165.0

****************************************
Lower Limit 182.5

upper Limit 842.5

***************************************************
Shape:
(25, 38)
****************************************
(0, 38)

****************************************
```

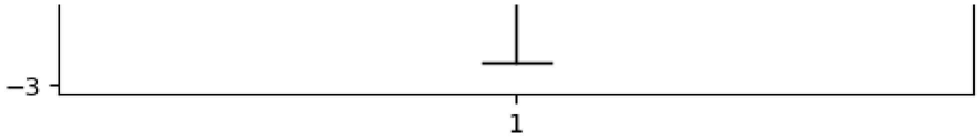


```
Describe:
count    3998.000000
mean       -0.037831
std         1.028666
min        -4.126700
25%        -0.713525
50%         0.046400
75%         0.702700
max         1.995300
Name: conscientiousness, dtype: float64
****************************************
```

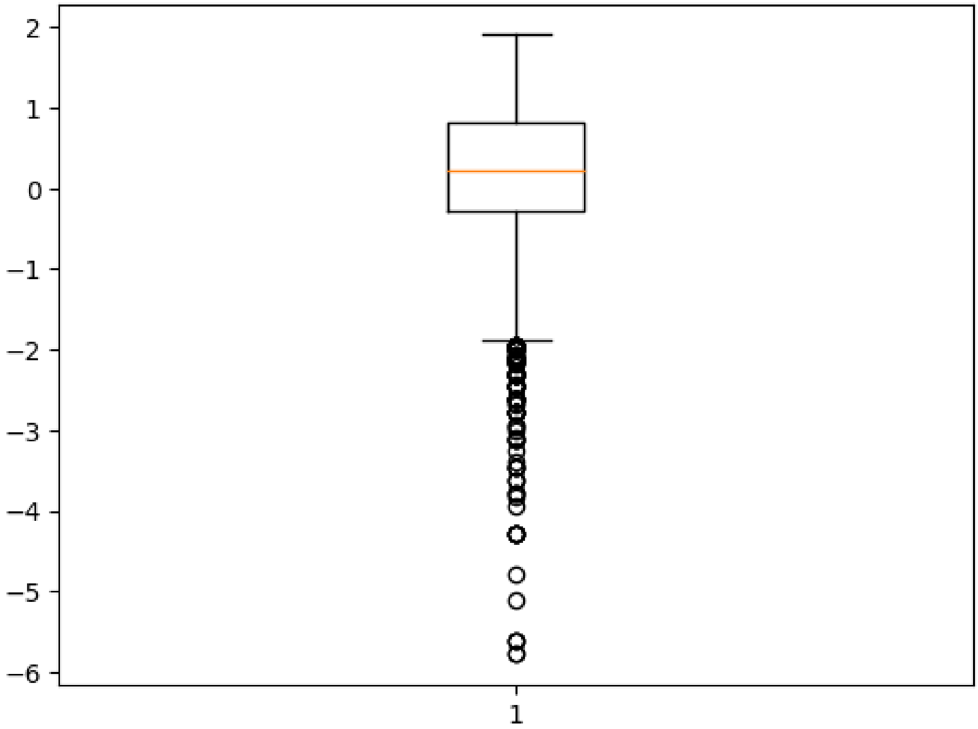


```
****************************************
Skewness:  -0.5270033403119497
****************************************

First Quartile:  -0.7135250000000001

Third Quartile:  0.7027
****************************************

InterQuartile Range:  1.416225

****************************************
Lower Limit -2.8378625000000004

upper Limit 2.8270375000000003

**************************************************
Shape:
(39, 38)
****************************************
(0, 38)

****************************************
```

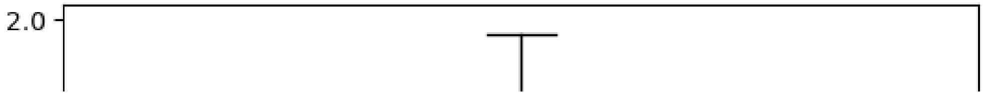


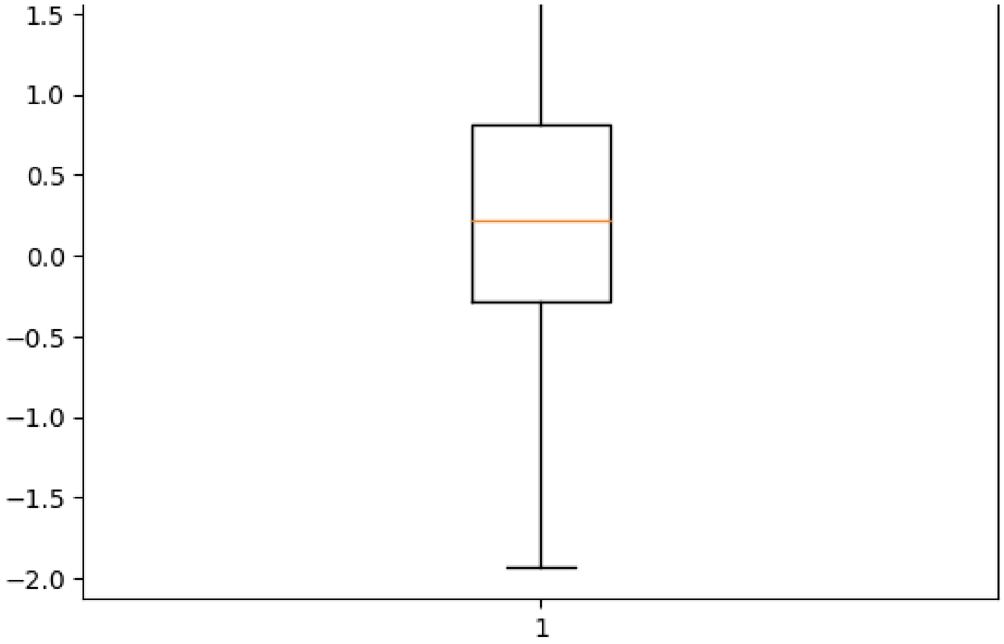


```
Describe:
count    3998.000000
mean        0.146496
std         0.941782
min        -5.781600
25%        -0.287100
50%         0.212400
75%         0.812800
max         1.904800
Name: agreeableness, dtype: float64
****************************************
```

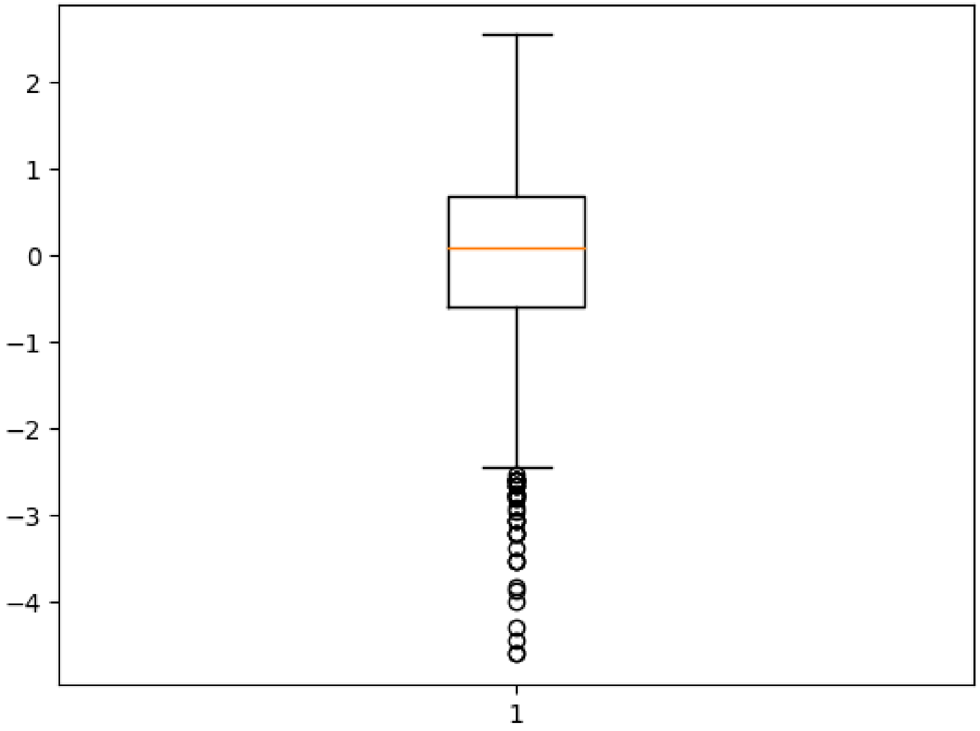


```
****************************************
Skewness:  -1.2049152493551414
****************************************

First Quartile:  -0.2871

Third Quartile:  0.8128
****************************************

InterQuartile Range:  1.0998999999999999

****************************************
Lower Limit -1.93695

upper Limit 2.46265

*****************************************************
Shape:
(123, 38)
****************************************
(0, 38)

****************************************
```



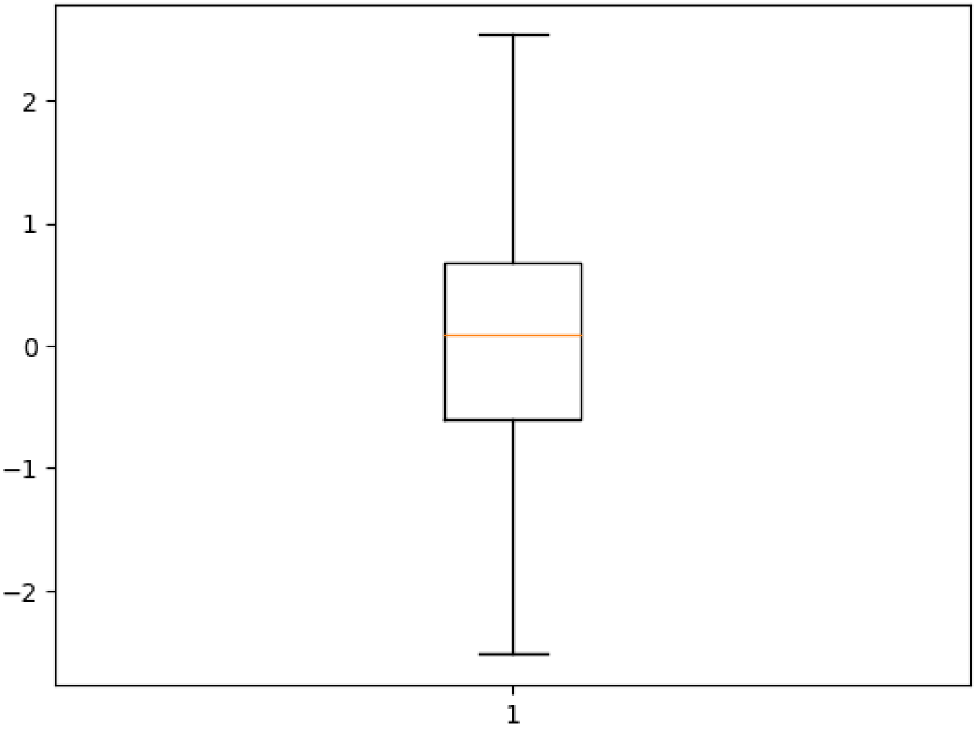


```
Describe:
count    3998.000000
mean        0.002763
std         0.951471
min        -4.600900
25%        -0.604800
50%         0.091400
75%         0.672000
max         2.535400
Name: extraversion, dtype: float64
****************************************
```

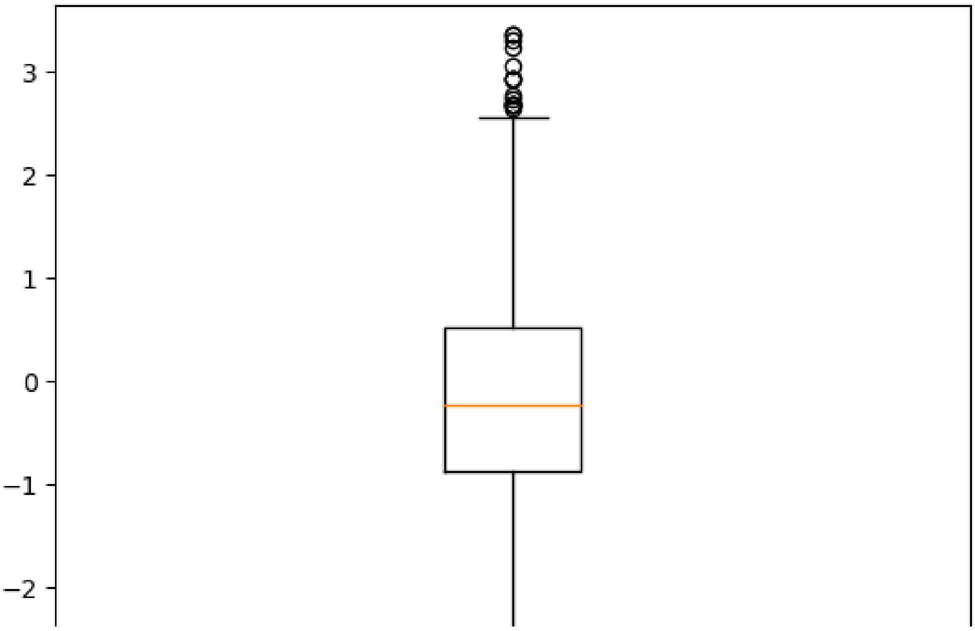


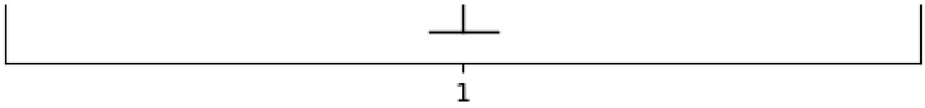
```
****************************************
Skewness:  -0.5232667810368843
****************************************

First Quartile:  -0.6048

Third Quartile:  0.672
****************************************

InterQuartile Range:  1.2768000000000002

****************************************
```

```
Lower Limit -2.5200000000000005

upper Limit 2.5872

***************************************************
Shape:
(40, 38)
****************************************
(0, 38)

****************************************
```

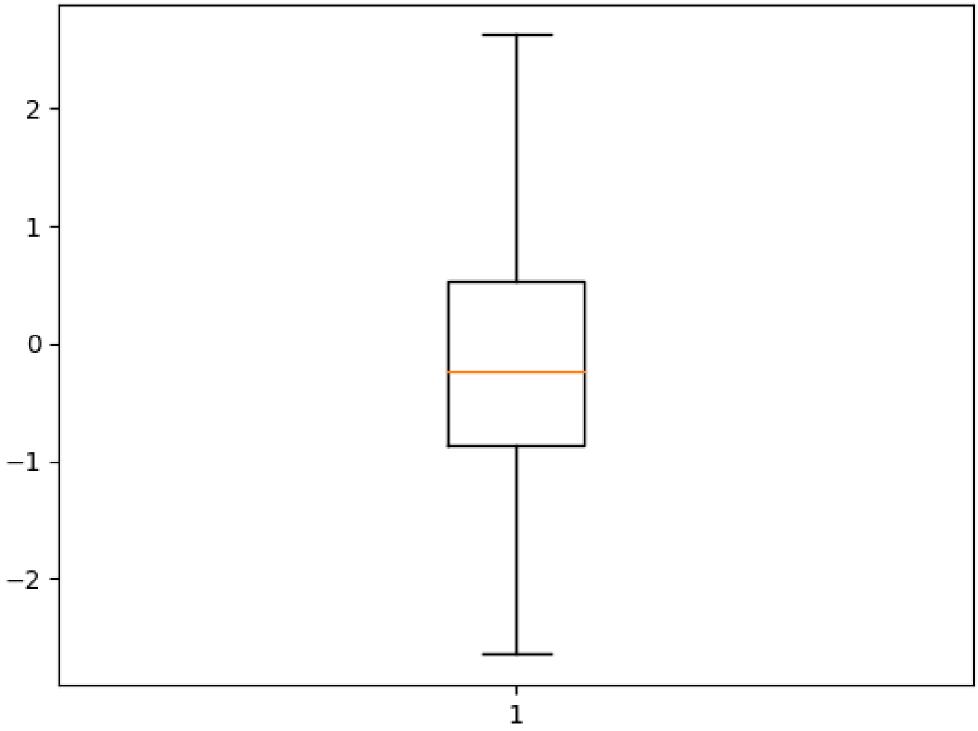


```
Describe:
count    3998.000000
mean       -0.169033
std         1.007580
min        -2.643000
25%        -0.868200
50%        -0.234400
75%         0.526200
max         3.352500
Name: nueroticism, dtype: float64
****************************************
```

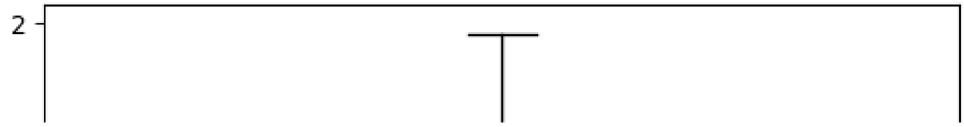


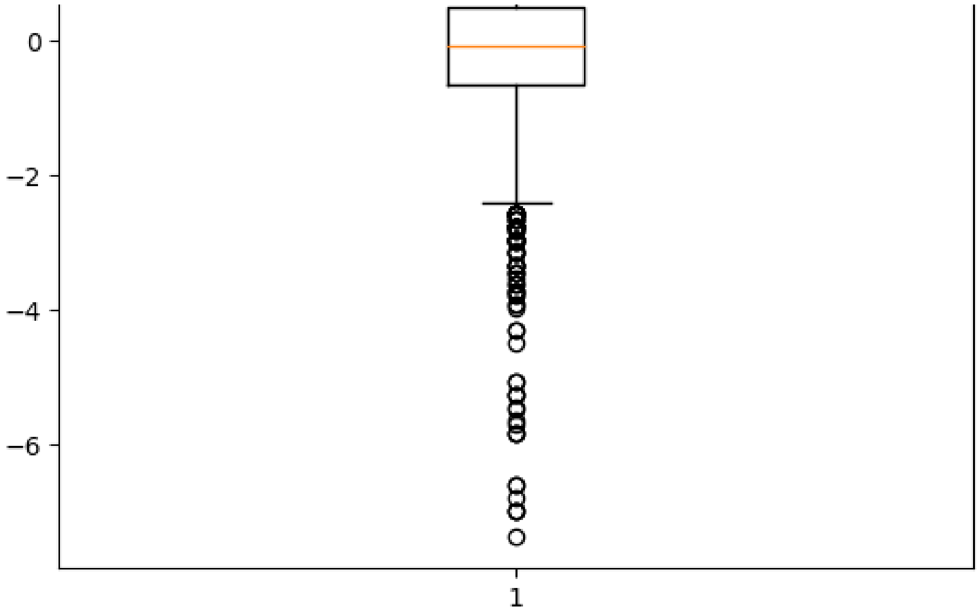


```
****************************************
Skewness:  0.1657096849156382
****************************************

First Quartile:  -0.8682

Third Quartile:  0.5262
****************************************

InterQuartile Range:  1.3944

****************************************
Lower Limit -2.9598

upper Limit 2.6178

***************************************************
Shape:
(15, 38)
****************************************
(0, 38)

****************************************
```

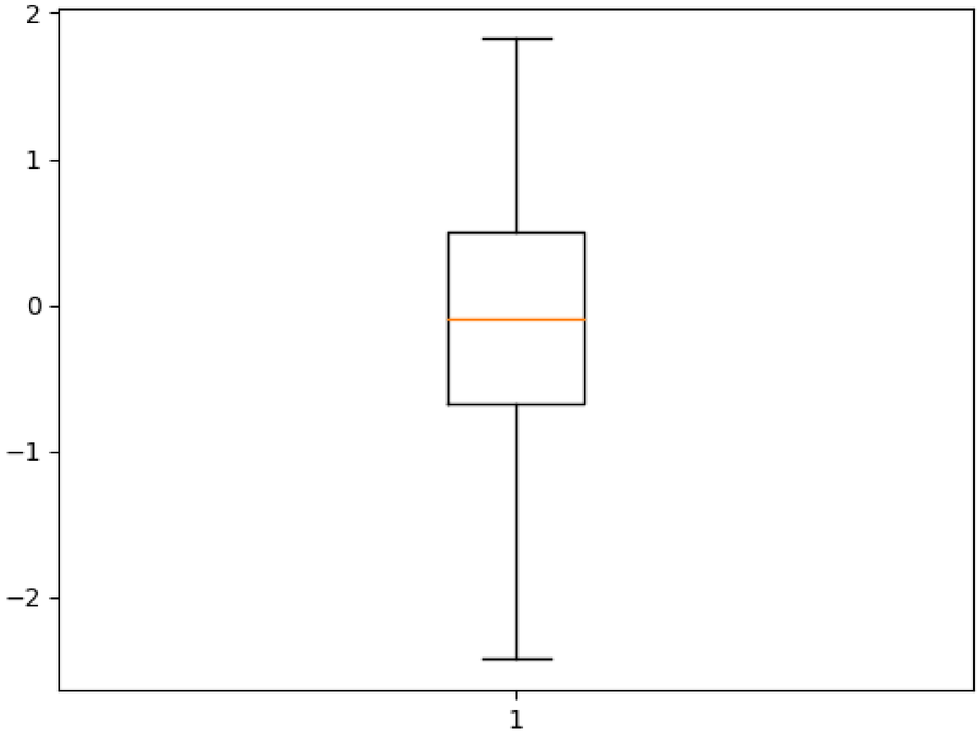


```
Describe:
count    3998.000000
mean       -0.138110
std         1.008075
min        -7.375700
25%        -0.669200
50%        -0.094300
75%         0.502400
max         1.822400
Name: openess_to_experience, dtype: float64
****************************************
```





```
****************************************
Skewness:  -1.5069620137292778
****************************************

First Quartile:  -0.6692

Third Quartile:  0.5024
****************************************

InterQuartile Range:  1.1716

****************************************
Lower Limit -2.4266

upper Limit 2.2598000000000003

***************************************************
Shape:
(95, 38)
****************************************
(0, 38)

****************************************
```



Data is importatnt for us,so we can not remove any data for our analysis.And here we can not do any outlier treat our columns,because so many columns having student's present or not in exam.

UNIVARIATE ANALYSIS

NUMERIC DATA TYPE

In [32]:

```
df.select_dtypes(["int64","float64"]).columns
```

Out[32]:

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

SALARY

In [33]:

```
# Minimum Salary
df["Salary"].min()
```

Out[33]:

```
35000.0
```

In [34]:

```
# Maximum Salary
df["Salary"].max()
```

Out[34]:

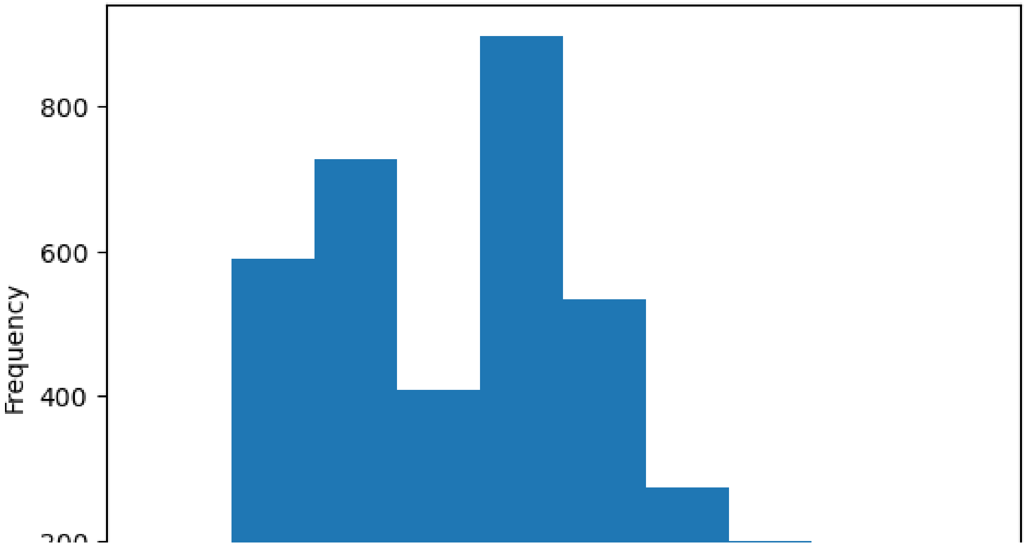
```
655000.0
```

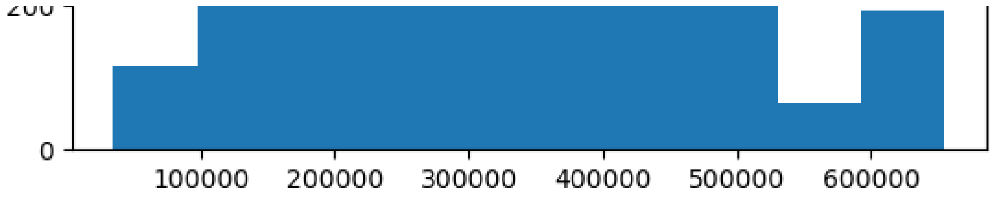
In [35]:

```
df["Salary"].plot(kind="hist")
```

Out[35]:

```
<AxesSubplot:ylabel='Frequency'>
```





10percentage

In [36]:

```
# Minimum Percentage in 10th Board
df['10percentage'].min()
```

Out[36]:

```
50.695000000000014
```

In [37]:

```
# Maximum Percentage in 10th Board
df['10percentage'].max()
```

Out[37]:

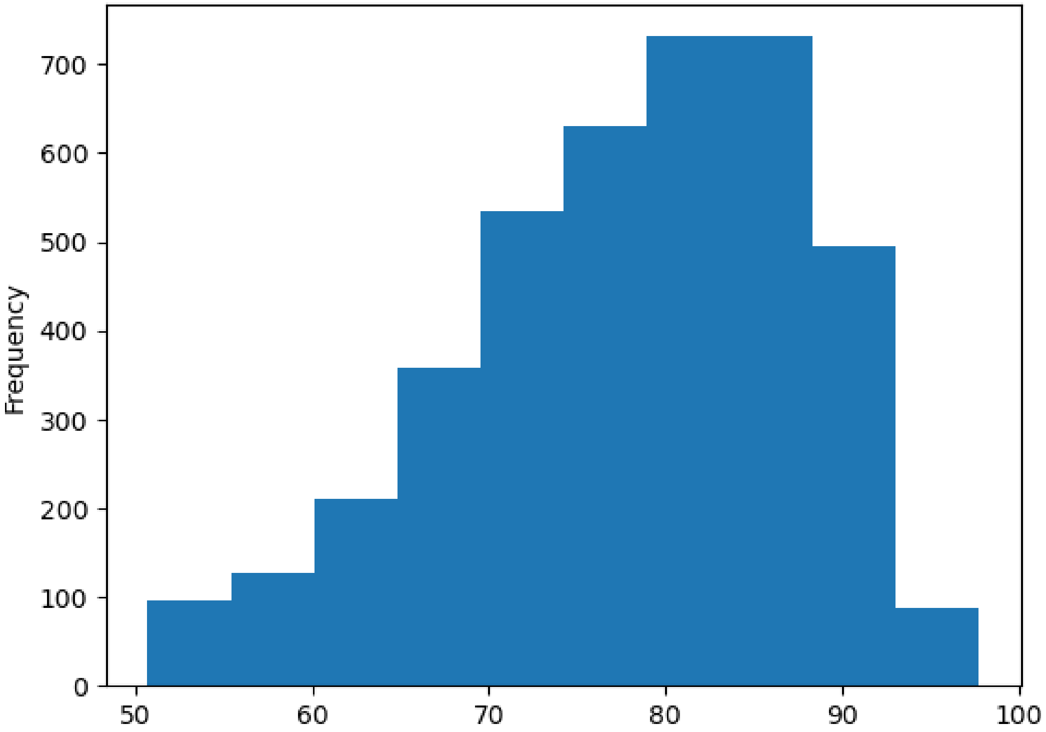
```
97.76
```

In [38]:

```
# It is looking like so many students were kept less marks
df['10percentage'].plot(kind="hist")
```

Out[38]:

```
<AxesSubplot:ylabel='Frequency'>
```



12percentage

In [39]:

```
# Minimum Mark
df["12percentage"].min()
```

Out[39]:

```
41.10000000000001
```

In [40]:

```
# Maximum Mark
df["12percentage"].max()
```

Out[40]:

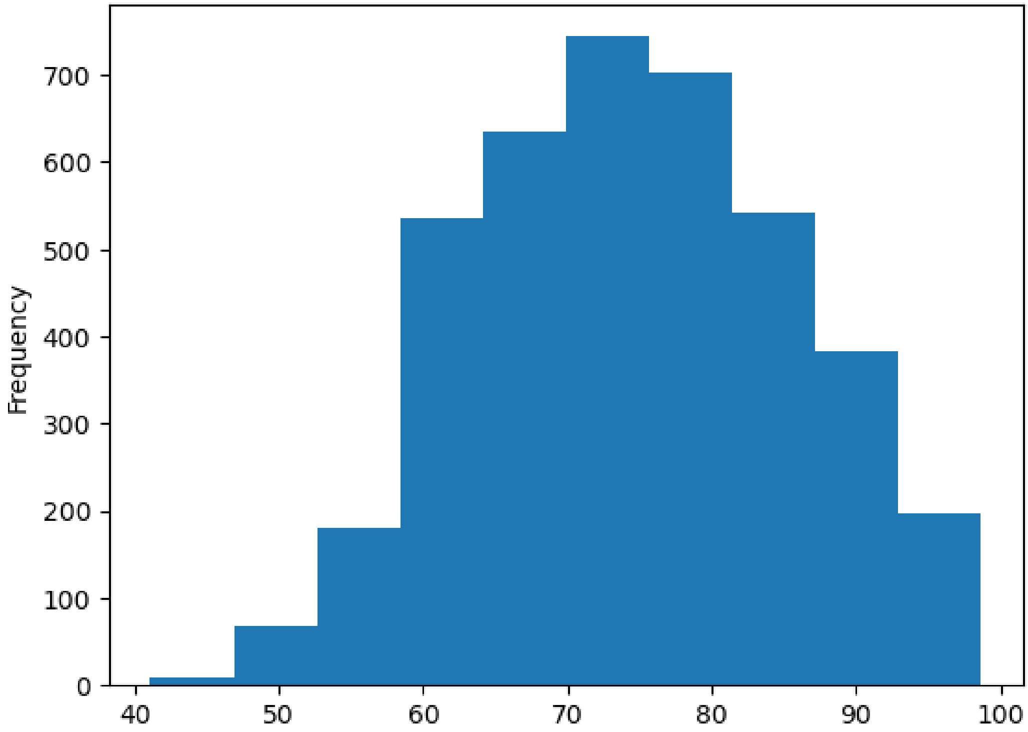
```
98.7
```

In [41]:

```
# So many students lies in between 60-100%
df["12percentage"].plot(kind="hist")
```

Out[41]:

```
<AxesSubplot:ylabel='Frequency'>
```



12graduation

In [42]:

```
# So many student were gave the exam in the year 2009
df['12graduation'].value_counts()
```

Out[42]:

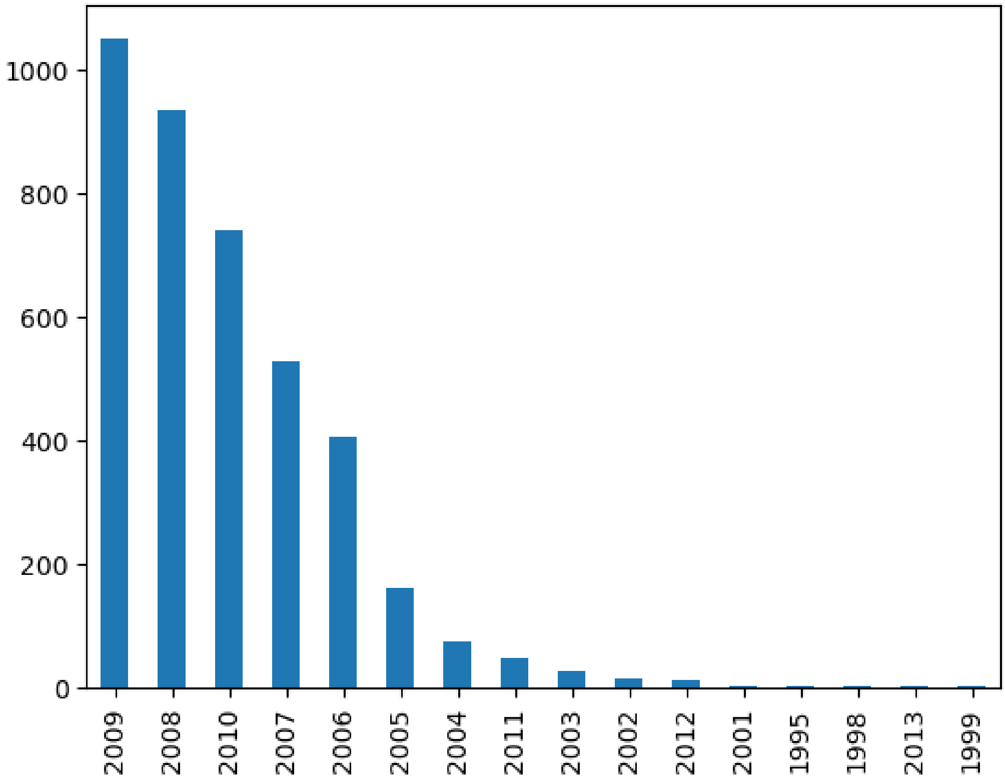
```
2009    1052
2008     935
2010     742
2007     528
2006     407
2005     160
2004      73
2011      46
2003      25
2002      14
2012      10
2001       2
1995       1
1998       1
2013       1
1999       1
Name: 12graduation, dtype: int64
```

In [43]:

```
df['12graduation'].value_counts().plot(kind="bar")
```

Out[43]:

```
<AxesSubplot:>
```



CollegeTier

In [44]:

```
# It give in which collegID so many students were went for the exam
# 1350 college were selected to conduct the exam
df['CollegeID'].value_counts()
```

Out[44]:

```
272      94
64       38
11759    35
44       35
47       33
         ..
128       1
5068      1
8637      1
9361      1
4883      1
Name: CollegeID, Length: 1350, dtype: int64
```

CollegeTier

In [45]:

```
df['CollegeTier'].value_counts()
```

Out[45]:

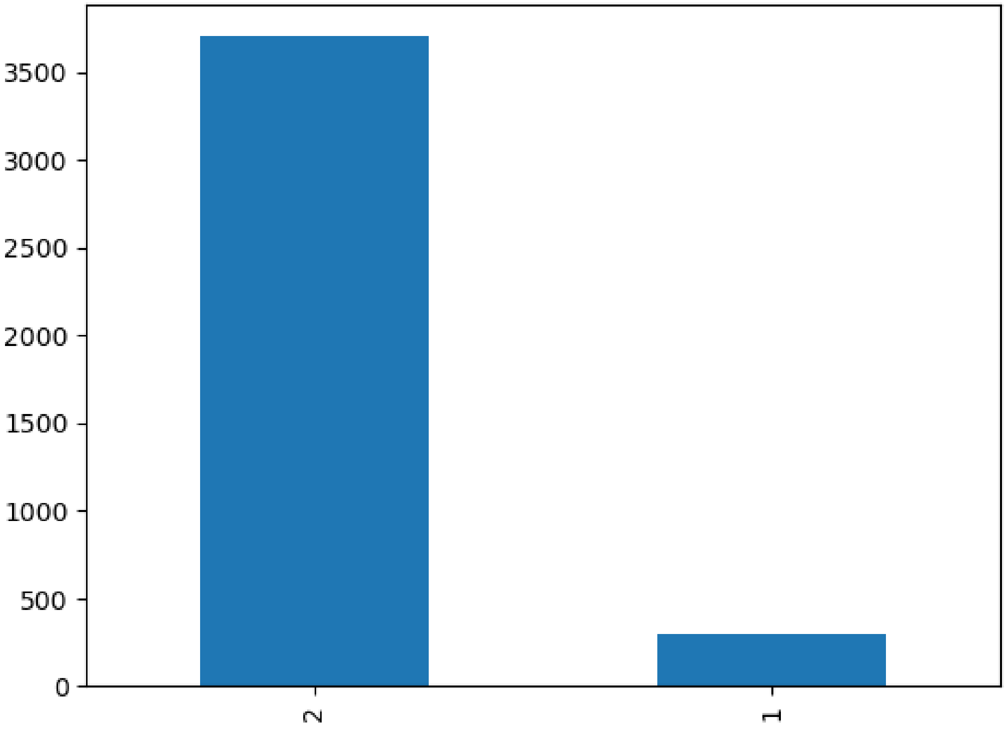
```
2    3701
1     297
Name: CollegeTier, dtype: int64
```

In [46]:

```
df['CollegeTier'].value_counts().plot(kind="bar")
```

Out[46]:

```
<AxesSubplot:>
```



collegeGPA

In [47]:

```
df['collegeGPA'].min()
```

Out[47]:

```
51.527499999999996
```

In [48]:

```
df['collegeGPA'].max()
```

Out[48]:

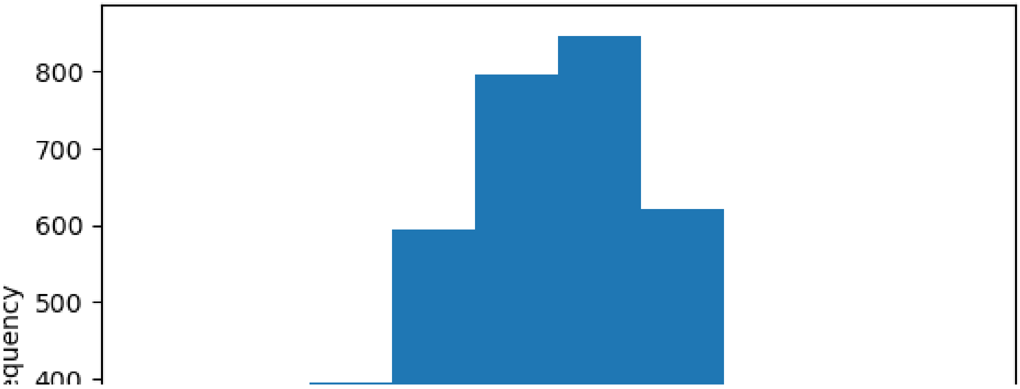
```
91.20750000000001
```

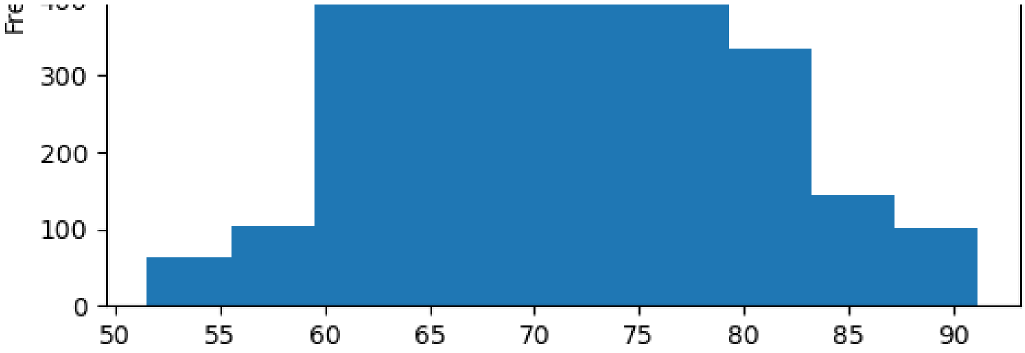
In [49]:

```
# 'collegeGPA'column Normally distributed
df['collegeGPA'].plot(kind="hist")
```

Out[49]:

```
<AxesSubplot:ylabel='Frequency'>
```





CollegeCityID

In [50]:

```
# 'CollegeCityID' is exactly same as 'CollegeID' so this column is not required for our analysis
df['CollegeCityID'].value_counts()
```

Out[50]:

```
272      94
64       38
11759    35
44       35
47       33
         ..
128       1
5068      1
8637      1
9361      1
4883      1
Name: CollegeCityID, Length: 1350, dtype: int64
```

CollegeCityTier

In [51]:

```
df['CollegeCityTier'].value_counts()
```

Out[51]:

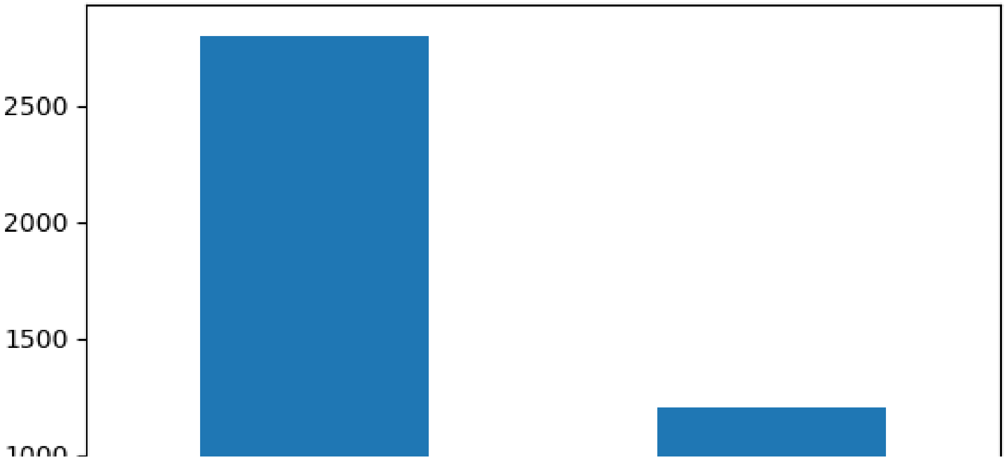
```
0    2797
1    1201
Name: CollegeCityTier, dtype: int64
```

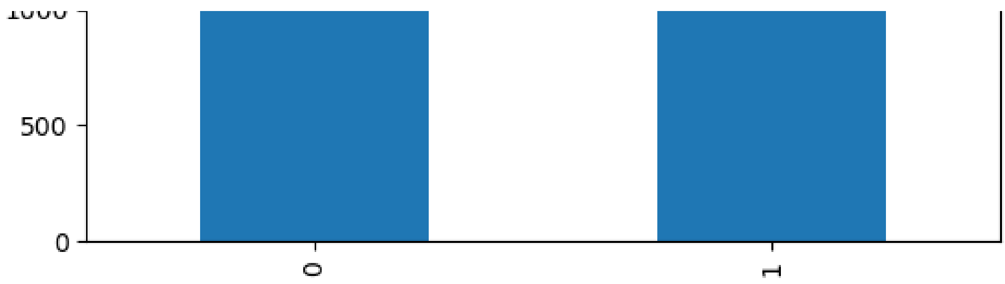
In [52]:

```
df['CollegeCityTier'].value_counts().plot(kind="bar")
```

Out[52]:

```
<AxesSubplot:>
```





GraduationYear

In [53]:

```
# So many student were graduated in the year 2013
df['GraduationYear'].value_counts()
```

Out[53]:

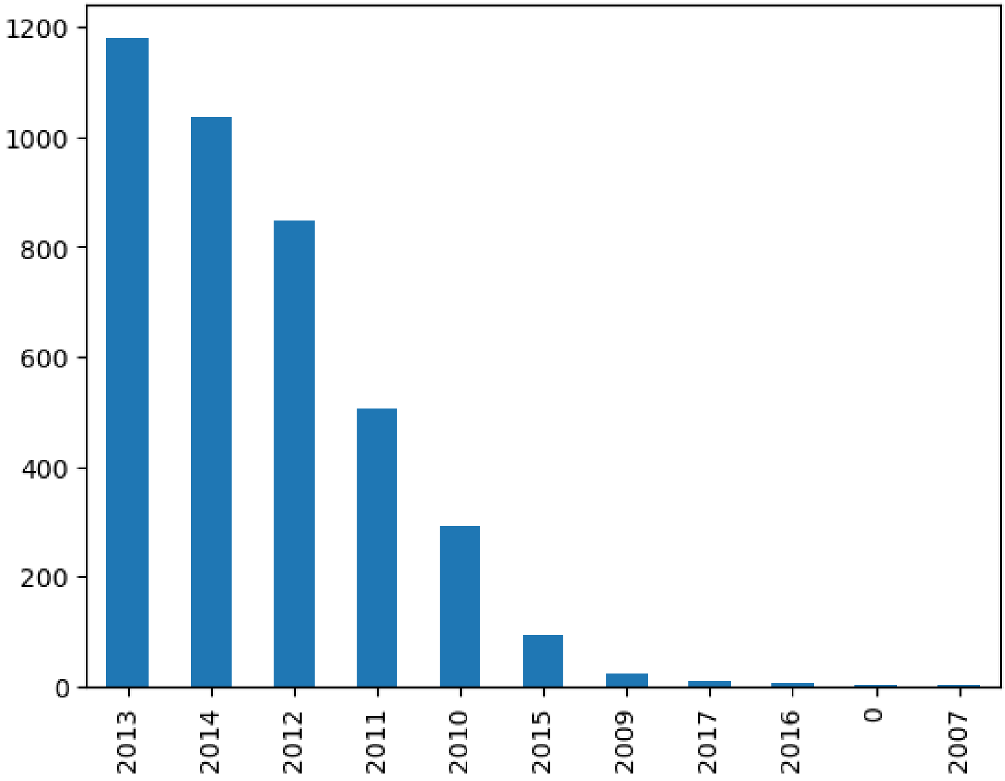
```
2013    1181
2014    1036
2012     847
2011     507
2010     292
2015      94
2009      24
2017       8
2016       7
0          1
2007       1
Name: GraduationYear, dtype: int64
```

In [54]:

```
df['GraduationYear'].value_counts().plot(kind="bar")
```

Out[54]:

```
<AxesSubplot:>
```



English

In [55]:

```
df['English'].min()
```

Out[55]:

```
207.5
```

In [56]:

```
df['English'].max()
```

Out[56]:

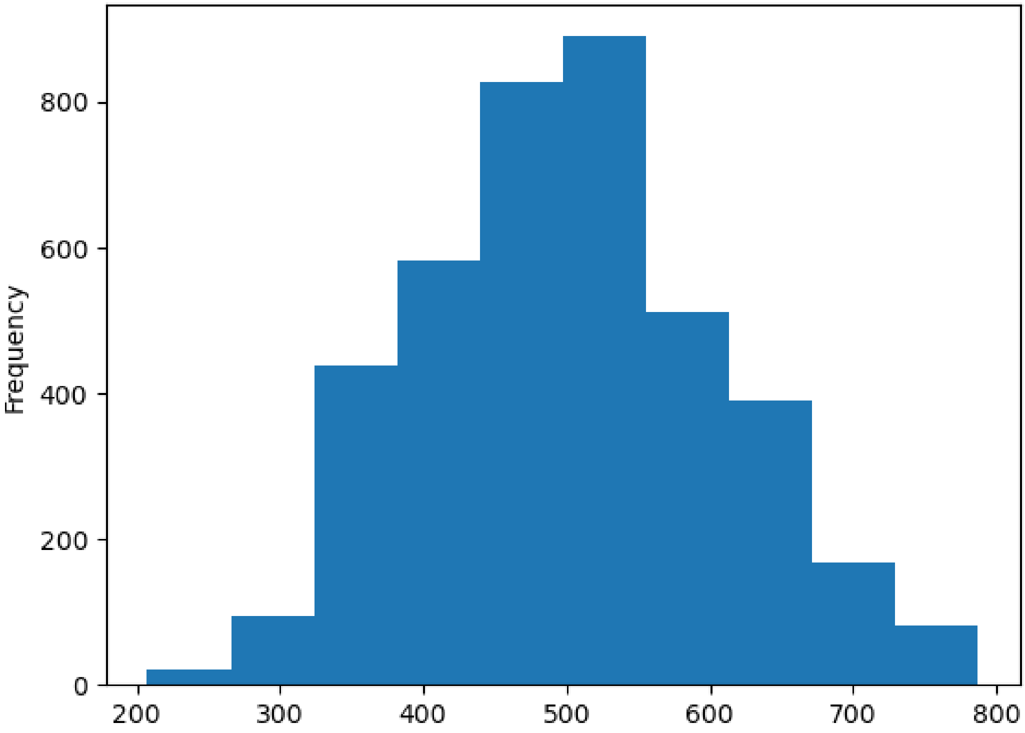
```
787.5
```

In [57]:

```
# This also Normally Distributed
df['English'].plot(kind="hist")
```

Out[57]:

```
<AxesSubplot:ylabel='Frequency'>
```



Logical

In [58]:

```
df['Logical'].min()
```

Out[58]:

```
265.0
```

In [59]:

```
df['Logical'].max()
```

Out[59]:

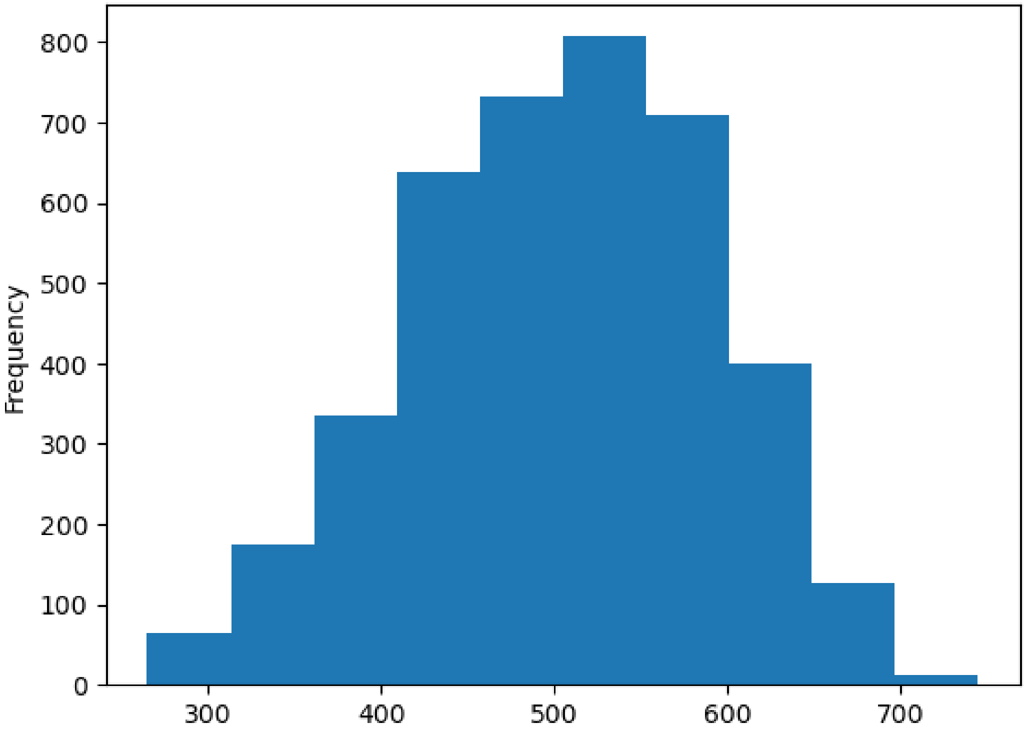
```
745.0
```

In [60]:

```
# It slidely left skewed
df['Logical'].plot(kind="hist")
```

Out[60]:

```
<AxesSubplot:ylabel='Frequency'>
```



Quant

In [61]:

```
df['Quant'].min()
```

Out[61]:

```
182.5
```

In [62]:

```
df['Quant'].max()
```

Out[62]:

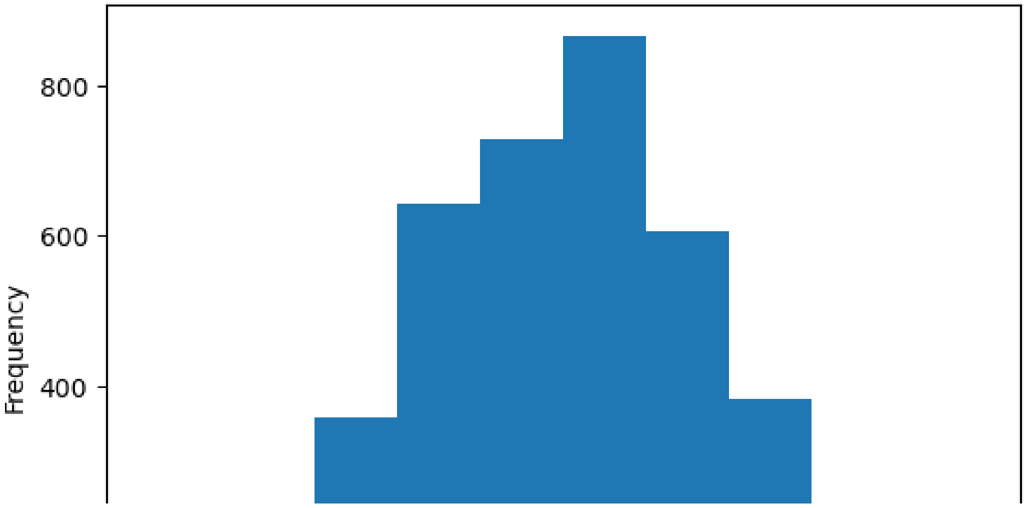
```
842.5
```

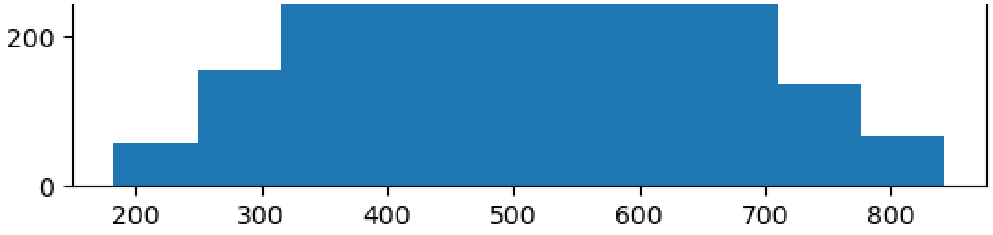
In [63]:

```
# It Normally Distributed
df['Quant'].plot(kind="hist")
```

Out[63]:

```
<AxesSubplot:ylabel='Frequency'>
```





ComputerProgramming

In [64]:

```
df['ComputerProgramming']
```

Out[64]:

```
0       445
1        -1
2       395
3       615
4        -1
       ...
3993    345
3994    325
3995    405
3996    445
3997    435
Name: ComputerProgramming, Length: 3998, dtype: int64
```

In [65]:

```
# It give how many student's were not give ComputerProgramming Exam
df[df['ComputerProgramming']==-1].shape
```

Out[65]:

```
(868, 38)
```

In [66]:

```
# Minimum Marks of ComputerProgramming Exam who were gave the exam
df[df['ComputerProgramming']!=-1]['ComputerProgramming'].min()
```

Out[66]:

```
105
```

In [67]:

```
# Maximum Marks of ComputerProgramming Exam who were gave the exam
df[df['ComputerProgramming']!=-1]['ComputerProgramming'].max()
```

Out[67]:

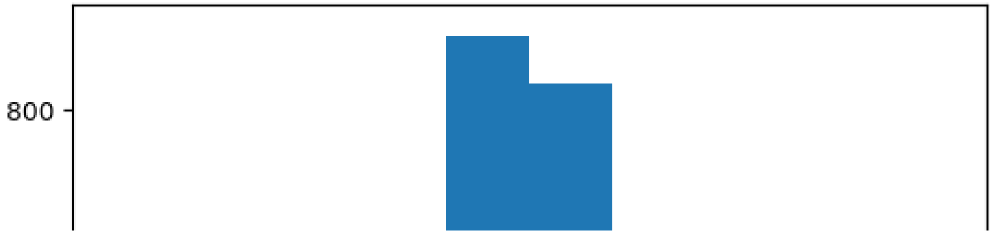
```
840
```

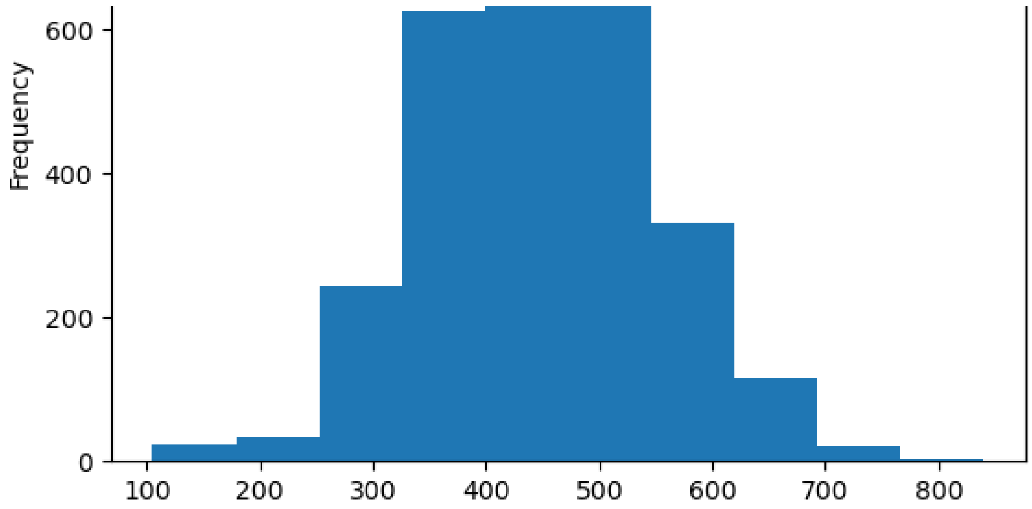
In [68]:

```
df[df['ComputerProgramming']!=-1]['ComputerProgramming'].plot(kind="hist")
```

Out[68]:

```
<AxesSubplot:ylabel='Frequency'>
```





Electronics And Semicon

In [69]:

```
df['ElectronicsAndSemicon']
```

Out[69]:

```
0        -1
1       466
2        -1
3        -1
4       233
       ... 
3993     -1
3994    420
3995     -1
3996     -1
3997     -1
Name: ElectronicsAndSemicon, Length: 3998, dtype: int64
```

In [70]:

```
# It give how many student's were not give Electronics And Semicon Exam
df[df['ElectronicsAndSemicon']==-1].shape
```

Out[70]:

```
(2854, 38)
```

In [71]:

```
# Minimum Marks of ComputerProgramming Exam who were gave the exam
df[df['ElectronicsAndSemicon']!=-1]['ElectronicsAndSemicon'].min()
```

Out[71]:

```
133
```

In [72]:

```
# Maximum Marks of ComputerProgramming Exam who were gave the exam
df[df['ElectronicsAndSemicon']!=-1]['ElectronicsAndSemicon'].max()
```

Out[72]:

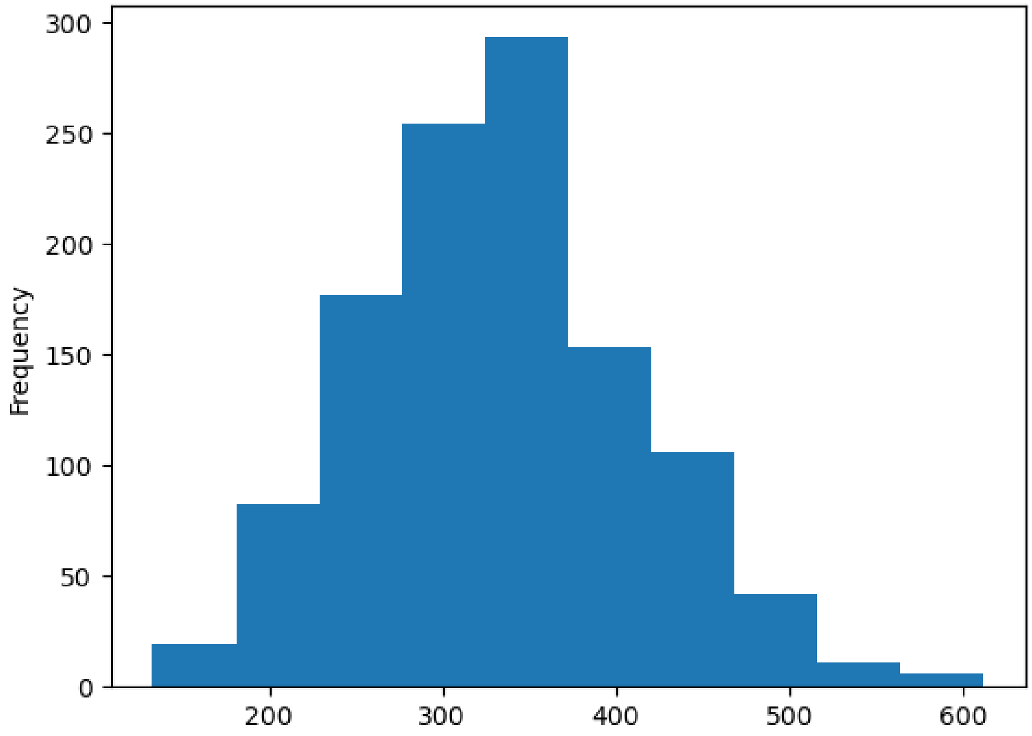
```
612
```

In [73]:

```
# It slidely right skewwed that means so many students were lies between 350-600
df[df['ElectronicsAndSemicon']!=-1]['ElectronicsAndSemicon'].plot(kind="hist")
```

Out[73]:

```
<AxesSubplot:ylabel='Frequency'>
```



ComputerScience

In [74]:

```
df['ComputerScience']
```

Out[74]:

```
0        -1
1        -1
2        -1
3        -1
4        -1
         ...
3993     -1
3994     -1
3995     -1
3996    438
3997     -1
Name: ComputerScience, Length: 3998, dtype: int64
```

In [75]:

```
# It give how many student's were not give Computer Science Exam
df[df['ComputerScience']==-1].shape
```

Out[75]:

```
(3096, 38)
```

In [76]:

```
# Minimum Marks of Computer Science Exam who were gave the exam
df[df['ComputerScience']!=-1]['ComputerScience'].min()
```

Out[76]:

```
130
```

In [77]:

```
# Maximum Marks of Computer Science Exam who were gave the exam
df[df['ComputerScience']!=-1]['ComputerScience'].max()
```

Out[77]:

Out[77]:

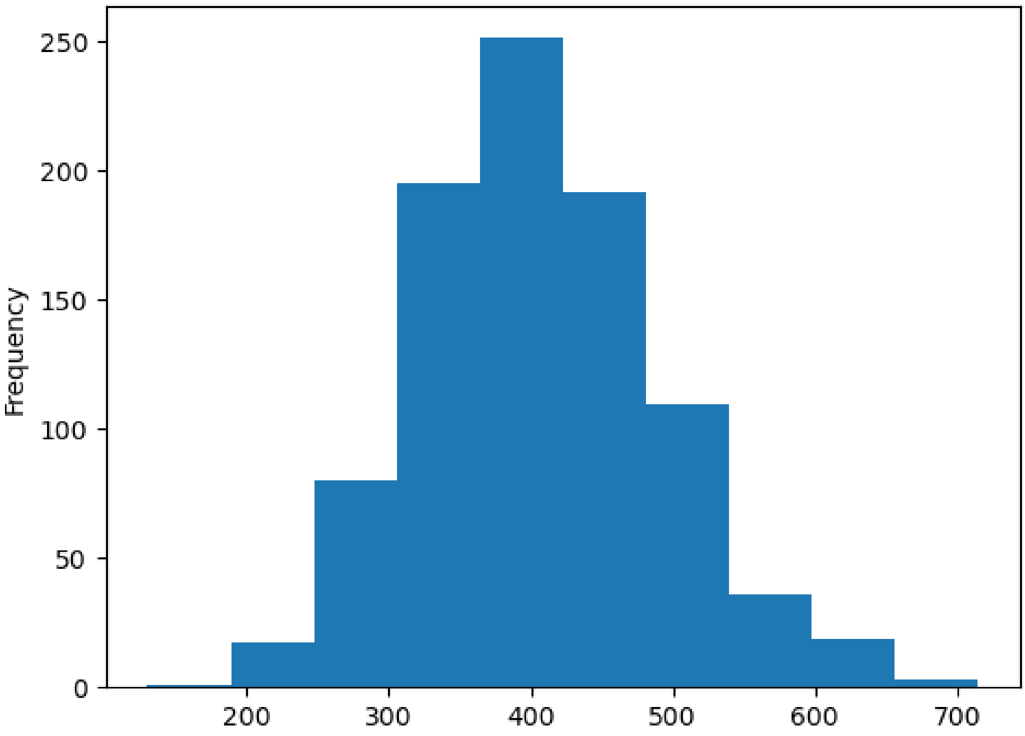
```
715
```

In [78]:

```
# It Normally Distributed
df[df['ComputerScience']!=-1]['ComputerScience'].plot(kind="hist")
```

Out[78]:

```
<AxesSubplot:ylabel='Frequency'>
```



MechanicalEngg

In [79]:

```
df['MechanicalEngg']
```

Out[79]:

```
0       -1
1       -1
2       -1
3       -1
4       -1
        ..
3993    -1
3994    -1
3995    -1
3996    -1
3997    -1
Name: MechanicalEngg, Length: 3998, dtype: int64
```

In [80]:

```
# It give how many student's were not give Mechanical Engg Exam
df[df['MechanicalEngg']==-1].shape
```

Out[80]:

```
(3763, 38)
```

In [81]:

```
# Minimum Marks of Mechanical Engg Exam who were gave the exam
```

```
df[df['MechanicalEngg']!=-1]['MechanicalEngg'].min()
```

Out[81]:

```
180
```

In [82]:

```
# Maximum Marks of Mechanical Engg Exam who were gave the exam
df[df['MechanicalEngg']!=-1]['MechanicalEngg'].max()
```

Out[82]:

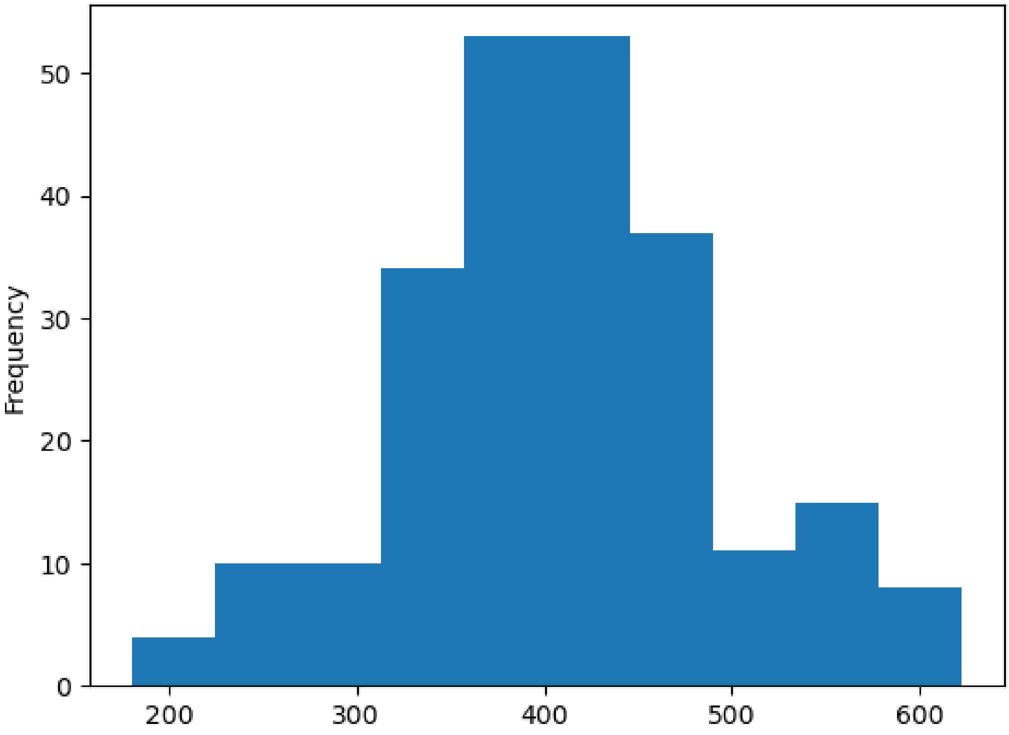
```
623
```

In [83]:

```
# It Normally Distributed
df[df['MechanicalEngg']!=-1]['MechanicalEngg'].plot(kind="hist")
```

Out[83]:

```
<AxesSubplot:ylabel='Frequency'>
```



ElectricalEngg

In [84]:

```
df['ElectricalEngg']
```

Out[84]:

```
0       -1
1       -1
2       -1
3       -1
4       -1
        ..
3993    -1
3994    -1
3995    -1
3996    -1
3997    -1
Name: ElectricalEngg, Length: 3998, dtype: int64
```

In [85]:

```
# It give how many student's were not give Electrical Engg Exam
df[df['ElectricalEngg']==-1].shape
```

Out[85]:

```
(3837, 38)
```

In [86]:

```
# Minimum Marks of Electrical Engg Exam who were gave the exam
df[df['ElectricalEngg']!=-1]['ElectricalEngg'].min()
```

Out[86]:

```
206
```

In [87]:

```
# Maximum Marks of Electrical Engg Exam who were gave the exam
df[df['ElectricalEngg']!=-1]['ElectricalEngg'].max()
```

Out[87]:

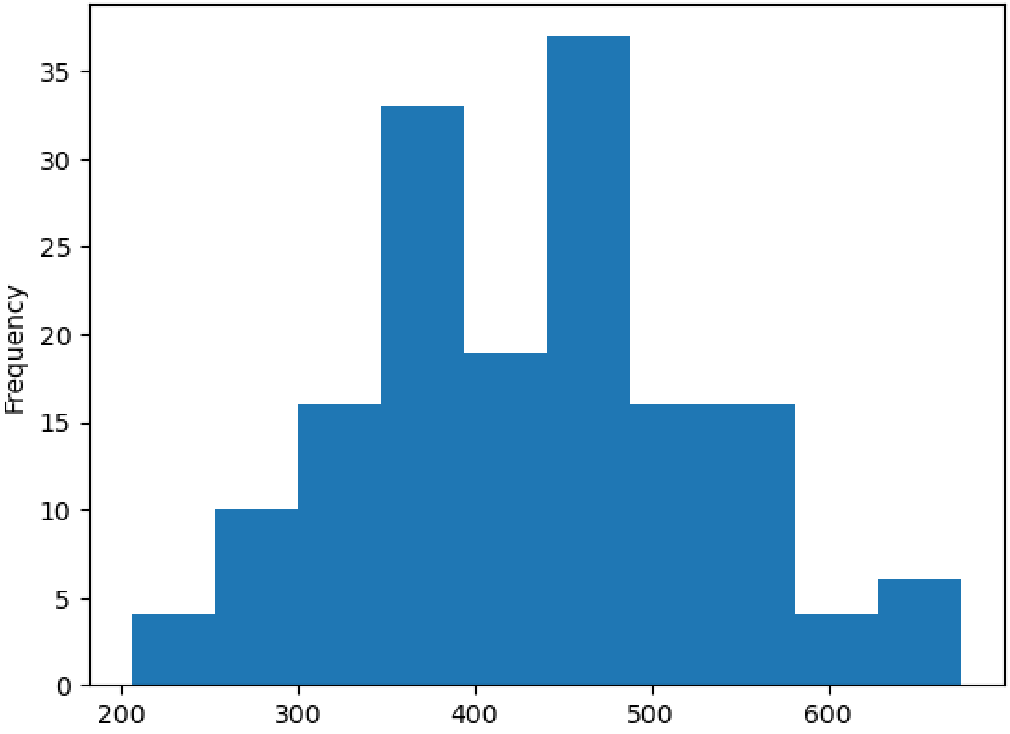
```
676
```

In [88]:

```
df[df['ElectricalEngg']!=-1]['ElectricalEngg'].plot(kind="hist")
```

Out[88]:

```
<AxesSubplot:ylabel='Frequency'>
```



TelecomEngg

In [89]:

```
df['TelecomEngg']
```

Out[89]:

```
0       -1
1       -1
2       -1
3       -1
4       -1
        ..
```

```
3993    -1
3994    -1
3995    -1
3996    -1
3997    -1
Name: TelecomEngg, Length: 3998, dtype: int64
```

In [90]:

```
# It give how many student's were not give Telecom Engg Exam
df[df['TelecomEngg']==-1].shape
```

Out[90]:

```
(3624, 38)
```

In [91]:

```
# Minimum Marks of Telecom Engg Exam who were gave the exam
df[df['TelecomEngg']!=-1]['TelecomEngg'].min()
```

Out[91]:

```
153
```

In [92]:

```
# Maximum Marks of Telecom Engg Exam who were gave the exam
df[df['TelecomEngg']!=-1]['TelecomEngg'].max()
```

Out[92]:

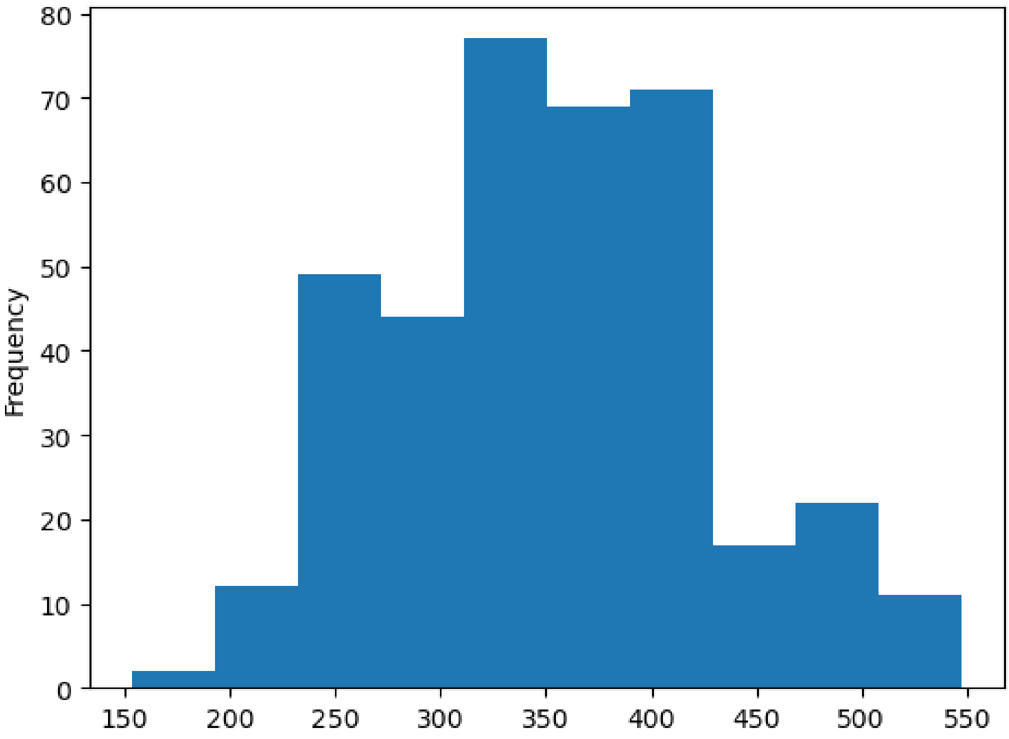
```
548
```

In [93]:

```
df[df['TelecomEngg']!=-1]['TelecomEngg'].plot(kind="hist")
```

Out[93]:

```
<AxesSubplot:ylabel='Frequency'>
```



CivilEngg

In [94]:

```
df['CivilEngg']
```

Out[94]:

```
0      -1
1      -1
2      -1
3      -1
4      -1
       ..
3993   -1
3994   -1
3995   -1
3996   -1
3997   -1
Name: CivilEngg, Length: 3998, dtype: int64
```

In [95]:

```
# It give how many student's were not give Civil Engg Exam
df[df['CivilEngg']==-1].shape
```

Out[95]:

```
(3956, 38)
```

In [96]:

```
# Minimum Marks of Civil Engg Exam who were gave the exam
df[df['CivilEngg']!=-1]['CivilEngg'].min()
```

Out[96]:

```
166
```

In [97]:

```
# Maximum Marks of Civil Engg Exam who were gave the exam
df[df['CivilEngg']!=-1]['CivilEngg'].max()
```

Out[97]:

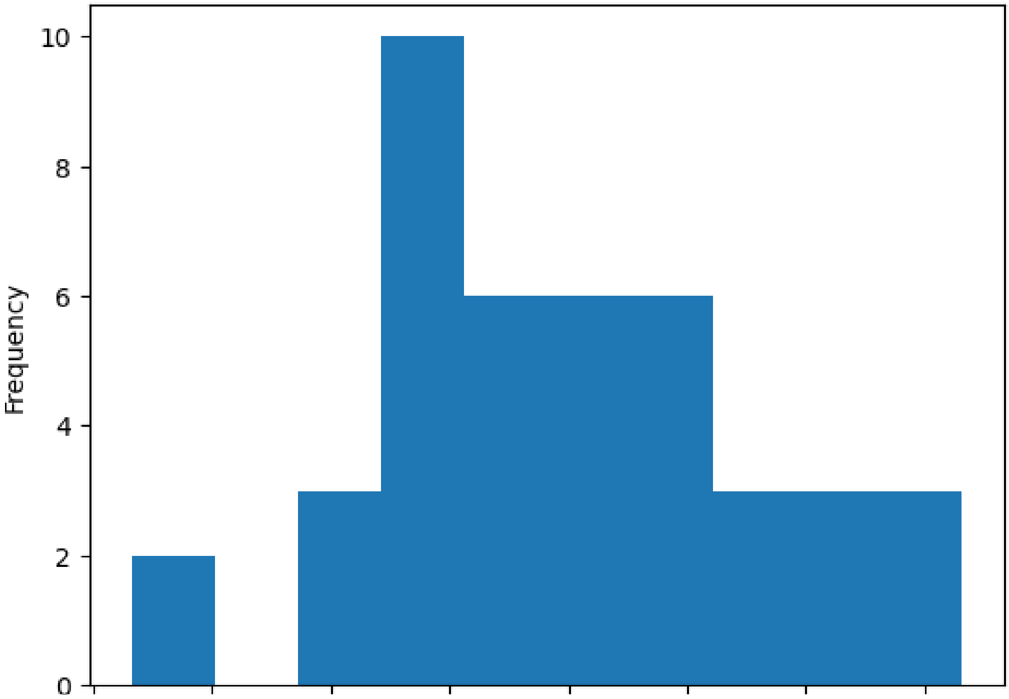
```
516
```

In [98]:

```
df[df['CivilEngg']!=-1]['CivilEngg'].plot(kind="hist")
```

Out[98]:

```
<AxesSubplot:ylabel='Frequency'>
```



CATEGORICAL DATA TYPE

In [99]:

```
df.select_dtypes("object").columns
```

Out[99]:

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

In [100]:

```
# Software Engineer is highest frequency
df['Designation'].value_counts()
```

Out[100]:

```
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: Designation, Length: 419, dtype: int64
```

In [101]:

```
df['Designation'].value_counts().head(30)
```

Out[101]:

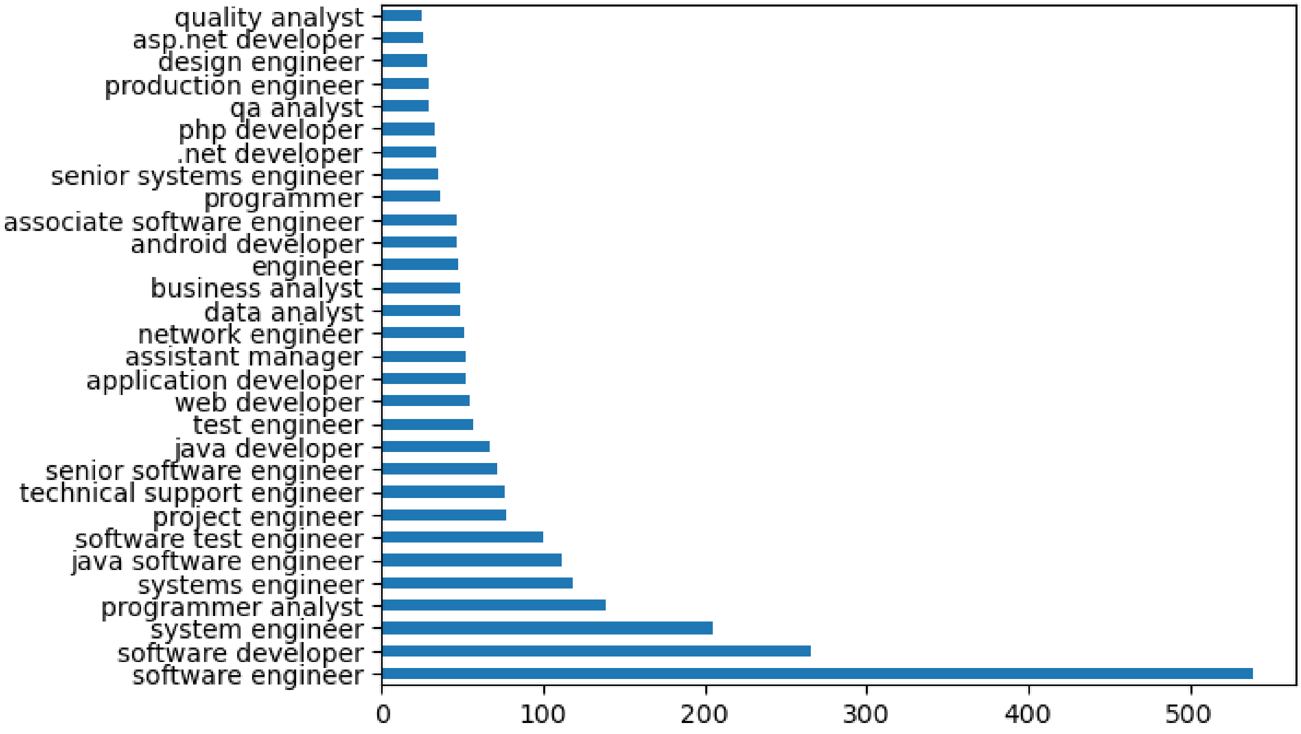
```
software engineer               539
software developer              265
system engineer                 205
programmer analyst              139
systems engineer                118
java software engineer          111
software test engineer          100
project engineer                 77
technical support engineer       76
senior software engineer         72
java developer                   67
test engineer                    57
web developer                    54
application developer            52
assistant manager                52
network engineer                 51
data analyst                     49
business analyst                 49
engineer                         47
android developer                46
associate software engineer      46
programmer                       36
senior systems engineer          35
.net developer                   34
php developer                    33
qa analyst                       29
production engineer              29
design engineer                  28
design engineer                28
asp.net developer              26
quality analyst                25
Name: Designation, dtype: int64
```

In [102]:

```
df['Designation'].value_counts().head(30).plot(kind="barh")
```

Out[102]:

```
<AxesSubplot:>
```



In [103]:

```
# Student from Bangalore City is high
df['JobCity'].value_counts()
```

Out[103]:

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

In [104]:

```
df['JobCity'].value_counts().head(25)
```

Out[104]:

```
Bangalore        627
-1               461
Noida            368
Hyderabad        335
Pune             290
Chennai          272
Gurgaon          198
New Delhi        196
Mumbai           108
```

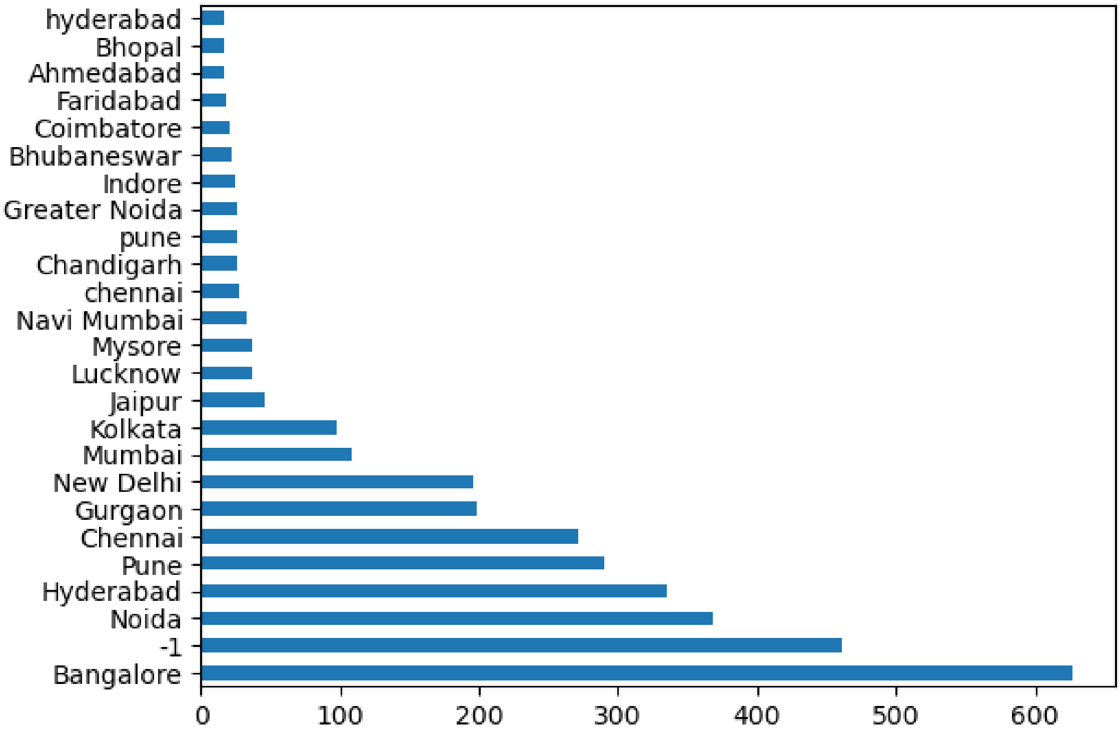
```
Kolkata            98
Jaipur             46
Lucknow            36
Mysore             36
Navi Mumbai        32
chennai            27
Chandigarh         26
pune               26
Greater Noida      26
Indore             24
Bhubaneswar        22
Coimbatore         20
Faridabad          18
Ahmedabad          17
Bhopal             17
hyderabad          16
Name: JobCity, dtype: int64
```

In [105]:

```
df['JobCity'].value_counts().head(25).plot(kind="barh")
```

Out[105]:

```
<AxesSubplot:>
```



In [106]:

```
# Male Gender have more Frequency
df['Gender'].value_counts()
```

Out[106]:

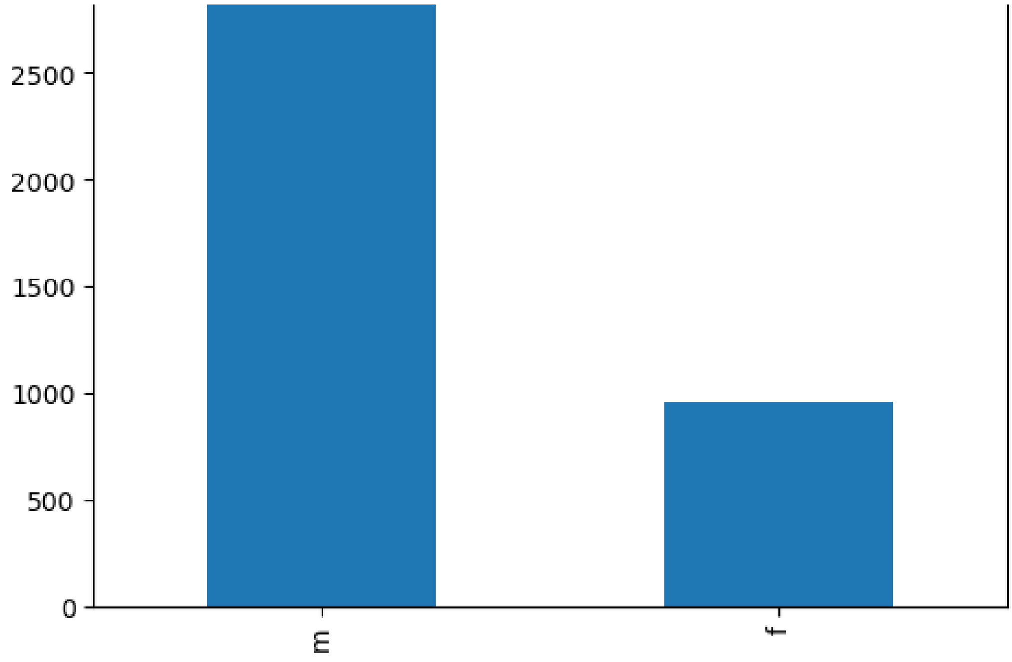
```
m    3041
f     957
Name: Gender, dtype: int64
```

In [107]:

```
df['Gender'].value_counts().plot(kind="bar")
```

Out[107]:

```
<AxesSubplot:>
```



In [108]:

```
# Student from CBSE Board are high
df['10board'].value_counts()
```

Out[108]:

```
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: 10board, Length: 275, dtype: int64
```

In [109]:

```
df['10board'].value_counts().head(20)
```

Out[109]:

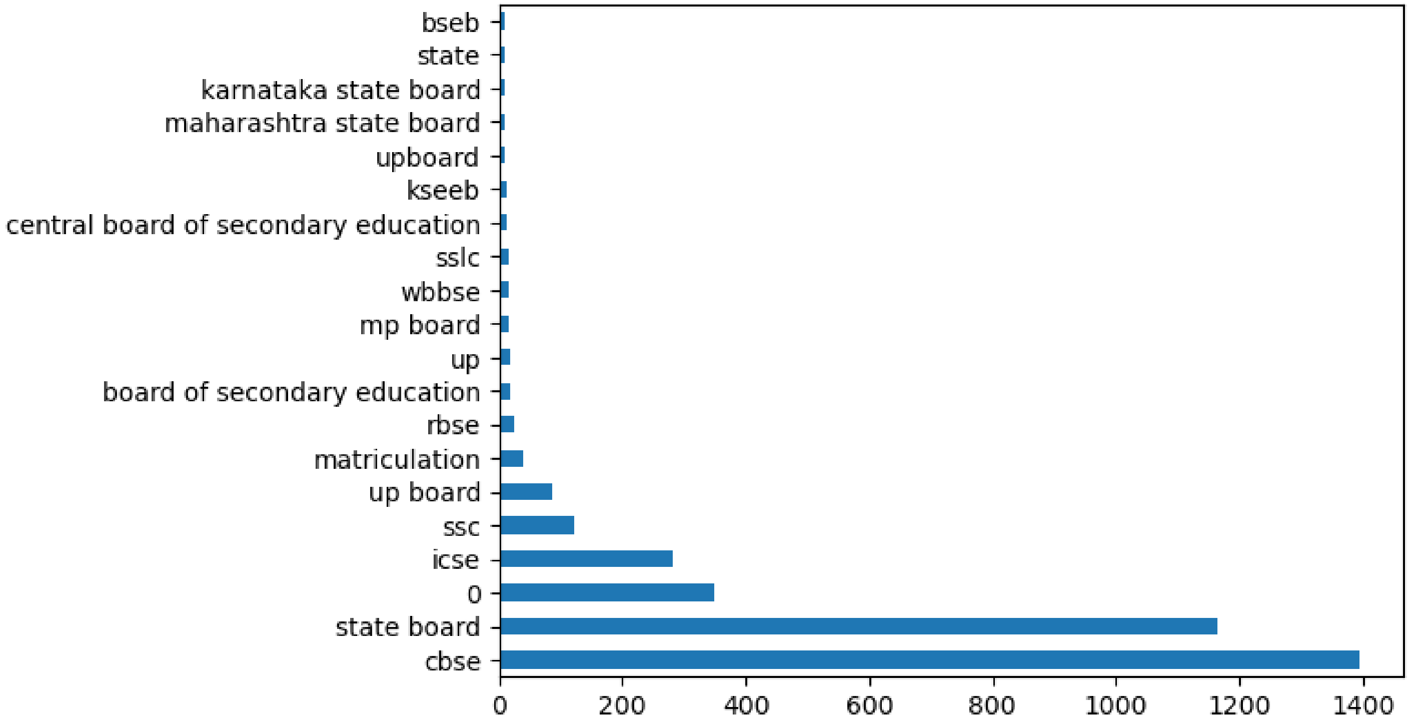
```
cbse                                        1395
state board                                 1164
0                                            350
icse                                         281
ssc                                          122
up board                                      85
matriculation                                 38
rbse                                          23
board of secondary education                  20
up                                            19
mp board                                      17
wbbse                                         16
sslc                                          16
central board of secondary education          13
kseeb                                         12
upboard                                       11
maharashtra state board                       11
karnataka state board                         10
state                                          9
bseb                                           9
Name: 10board, dtype: int64
```

In [110]:

```
df['10board'].value_counts().head(20).plot(kind="barh")
```

Out[110]:

```
<AxesSubplot:>
```



In [111]:

```
# Student from CBSE Board are high
df['12board'].value_counts()
```

Out[111]:

```
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: 12board, Length: 340, dtype: int64
```

In [112]:

```
df['12board'].value_counts().head(20)
```

Out[112]:

```
cbse                                      1400
state board                               1254
0                                          359
icse                                       129
up board                                    87
isc                                         45
board of intermediate                       36
board of intermediate education             31
up                                          20
rbse                                        19
mp board                                    17
bie                                         15
chse                                        14
ipe                                         14
hsc                                         13
maharashtra state board                     12
```

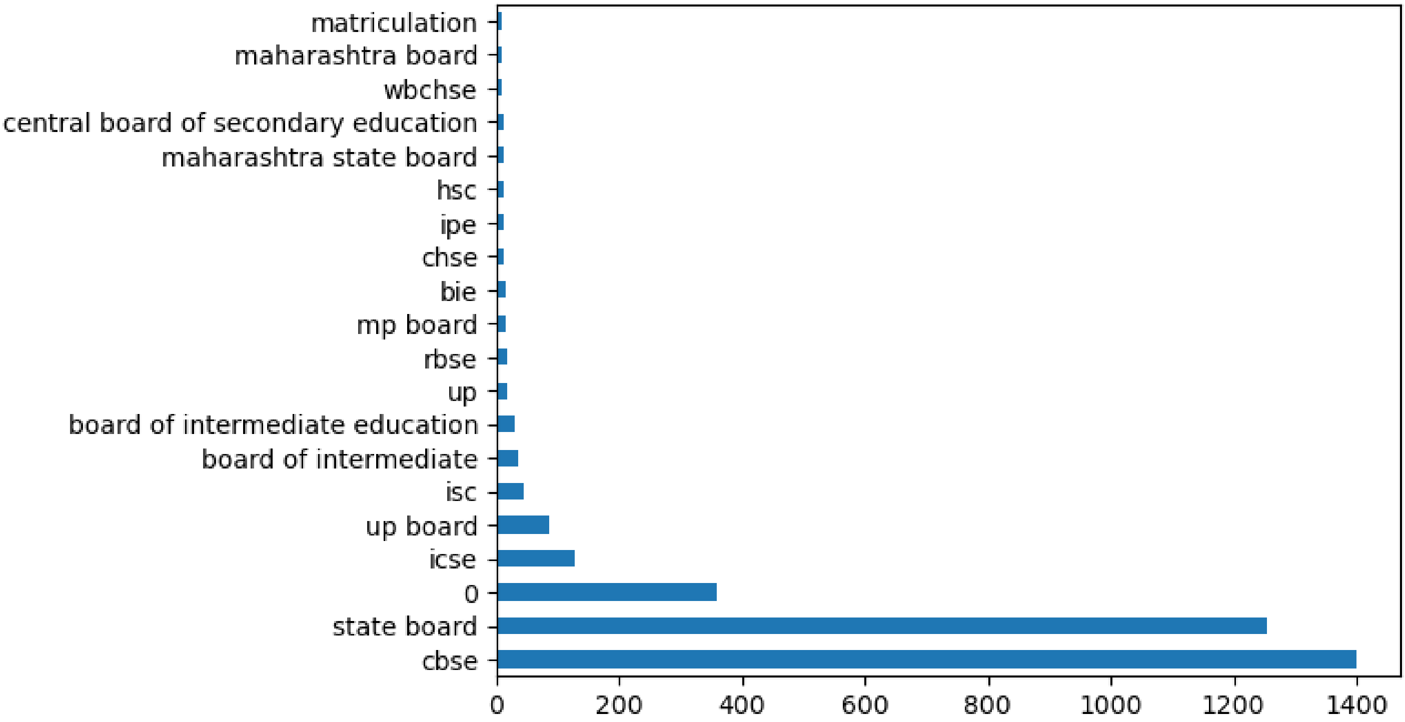
```
central board of secondary education      12
wbchse                                    11
maharashtra board                         10
matriculation                              9
Name: 12board, dtype: int64
```

In [113]:

```
df['12board'].value_counts().head(20).plot(kind="barh")
```

Out[113]:

```
<AxesSubplot:>
```



In [114]:

```
# Student from B.Tech/B.E. Degree are high
df['Degree'].value_counts()
```

Out[114]:

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

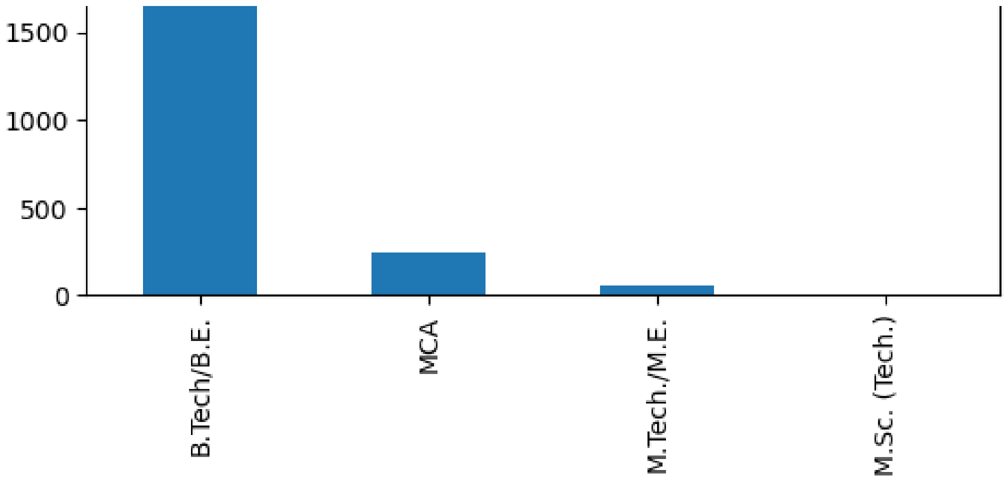
In [115]:

```
df['Degree'].value_counts().plot(kind="bar")
```

Out[115]:

```
<AxesSubplot:>
```

3500
3000
2500
2000



In [116]:

```
# Student from electronics and communication engineering Specialization are high
df['Specialization'].value_counts()
```

Out[116]:

```
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: Specialization, dtype: int64
```

In [117]:

```
df['Specialization'].value_counts().head(25)
```

Out[117]:

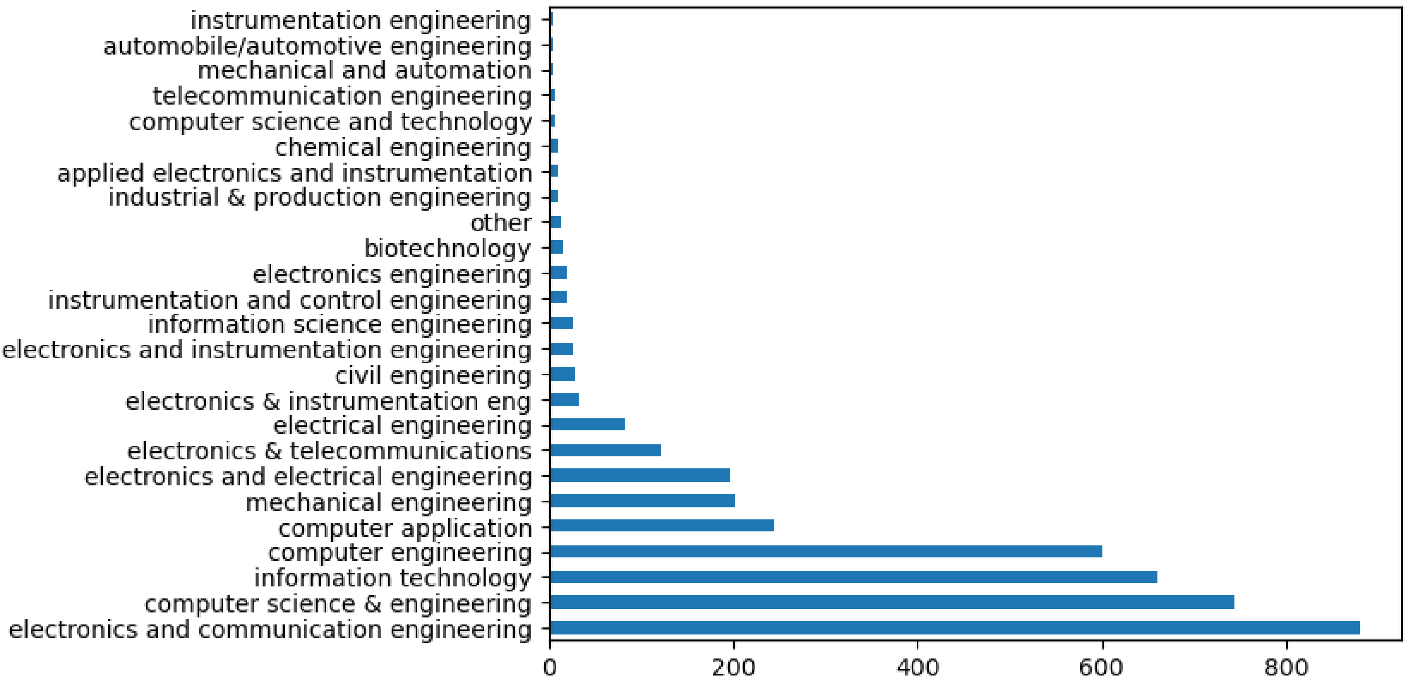
```
electronics and communication engineering      880
computer science & engineering                 744
information technology                         660
computer engineering                           600
computer application                           244
mechanical engineering                         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
Name: Specialization, dtype: int64
```

In [118]:

```
df['Specialization'].value_counts().head(25).plot(kind="barh")
```

Out[118]:

```
<AxesSubplot:>
```



In [119]:

```
# Student from Uttar Pradesh State are high
df['CollegeState'].value_counts().head(20)
```

Out[119]:

```
Uttar Pradesh        915
Karnataka            370
```

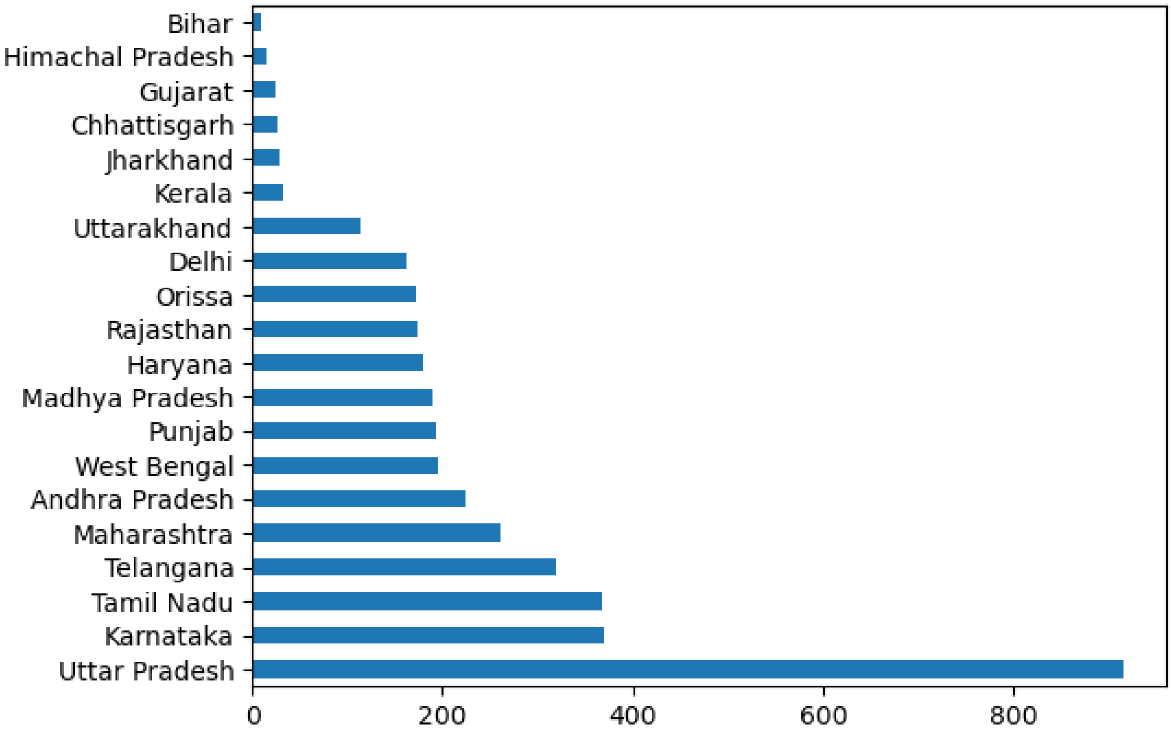
```
Tamil Nadu           367
Telangana            319
Maharashtra          262
Andhra Pradesh       225
West Bengal          196
Punjab               193
Madhya Pradesh       189
Haryana              180
Rajasthan            174
Orissa               172
Delhi                162
Uttarakhand          113
Kerala                33
Jharkhand             28
Chhattisgarh          27
Gujarat               24
Himachal Pradesh      16
Bihar                 10
Name: CollegeState, dtype: int64
```

In [120]:

```
df['CollegeState'].value_counts().head(20).plot(kind="barh")
```

Out[120]:

```
<AxesSubplot:>
```



BIVARIATE ANALYSIS

In [121]:

```
df.select_dtypes("object").columns
```

Out[121]:

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

In [122]:

```
df.select_dtypes(["int64","float64"]).columns
```

Out[122]:

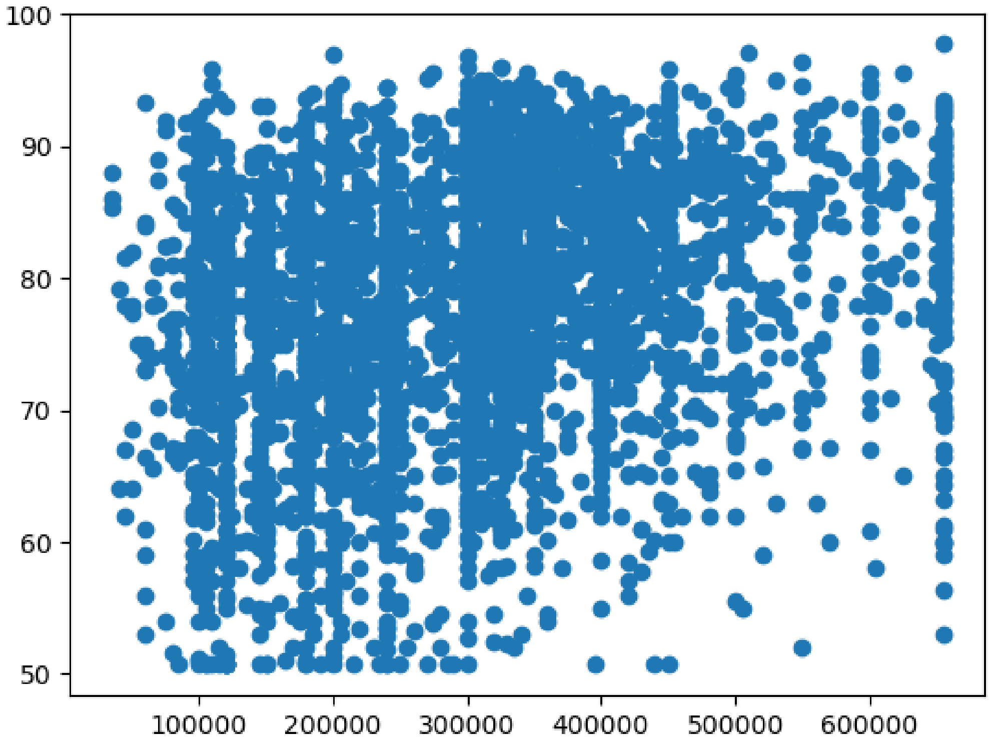
```
Index(['ID', 'Salary', '10percentage', '12graduation', '12percentage',
       'CollegeID', 'CollegeTier', 'collegeGPA', 'CollegeCityID',
       'CollegeCityTier', 'GraduationYear', 'English', 'Logical', 'Quant',
       'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience',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

In [123]:

```
plt.scatter(df['Salary'],df['10percentage'])
```

Out[123]:

```
<matplotlib.collections.PathCollection at 0x22fae63b070>
```



In [124]:

```
df.pivot_table(index='Designation',values='Salary',aggfunc="sum",sort=False).head(30)
```

Out[124]:

	Salary
Designation	
senior quality engineer	2220000.0
assistant manager	22285000.0
systems engineer	43455000.0
senior software engineer	34095000.0
get	3785000.0
system engineer	72580000.0
java software engineer	32355000.0
mechanical engineer	1575000.0
electrical engineer	6520000.0
project engineer	24095000.0

Designation	Salary
senior php developer	1455000.0
senior systems engineer	16155000.0
quality assurance engineer	4150000.0
qa analyst	7650000.0
network engineer	11420000.0
product development engineer	3865000.0
associate software developer	890000.0
data entry operator	360000.0
software engineer	181025000.0
developer	540000.0
electrical project engineer	1825000.0
programmer analyst	47230000.0
systems analyst	4215000.0
ase	1020000.0
telecommunication engineer	145000.0
application developer	18355000.0
ios developer	3145000.0
executive assistant	715000.0
online marketing manager	795000.0
documentation specialist	80000.0

In [125]:

```
# Total Salary of each designation
df.pivot_table(index='Designation',values='Salary',aggfunc="sum",sort=False).head(30).sort_values('Salary',ascending=False)
```

Out[125]:

	Salary
Designation	
software engineer	181025000.0
system engineer	72580000.0
programmer analyst	47230000.0
systems engineer	43455000.0
senior software engineer	34095000.0
java software engineer	32355000.0
project engineer	24095000.0
assistant manager	22285000.0
application developer	18355000.0
senior systems engineer	16155000.0
network engineer	11420000.0
qa analyst	7650000.0
electrical engineer	6520000.0
systems analyst	4215000.0
quality assurance engineer	4150000.0
product development engineer	3865000.0
get	3785000.0
ios developer	3145000.0

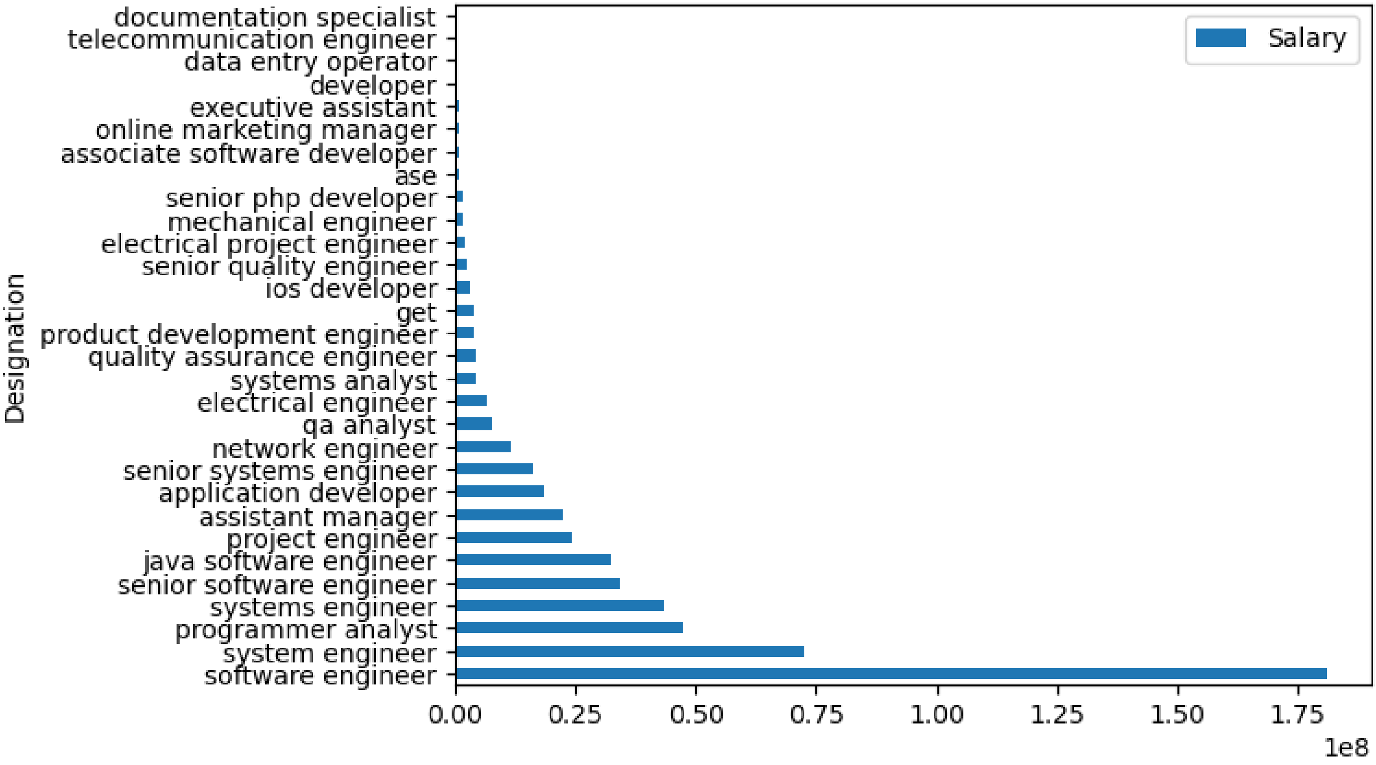
	Salary
Designation	
ios developer	3140000.0
senior quality engineer	2220000.0
electrical project engineer	1825000.0
mechanical engineer	1575000.0
senior php developer	1455000.0
ase	1020000.0
associate software developer	890000.0
online marketing manager	795000.0
executive assistant	715000.0
developer	540000.0
data entry operator	360000.0
telecommunication engineer	145000.0
documentation specialist	80000.0

In [126]:

```
df.pivot_table(index='Designation',values='Salary',aggfunc="sum",sort=False).head(30).so
rt_values('Salary',ascending=False).plot(kind="barh")
```

Out[126]:

```
<AxesSubplot:ylabel='Designation'>
```



Software Engineer Candidates get high salary

In [127]:

```
# Average Salary of each JobCity
df.pivot_table(index='JobCity',values='Salary',aggfunc="mean",sort=False).head(30).sort_
values('Salary',ascending=False)
```

Out[127]:

	Salary
JobCity	
Rajkot	452500.000000

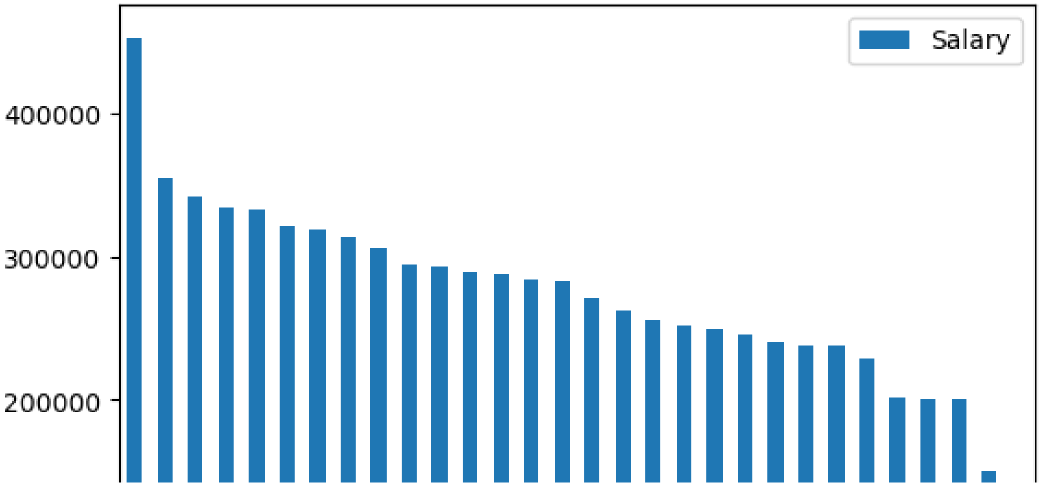
JobCity	Salary
Mumbai	355138.888889
Bangalore	341435.406699
Banglore	333888.888889
Mangalore	333181.818182
Pune	320775.862069
Navi Mumbai	318593.750000
Gurgaon	313181.818182
Hyderabad	305791.044776
Hyderabad	294500.000000
Chennai	293437.500000
-1	288850.325380
Noida	288546.195652
Mysore	284444.444444
Bangalore	282500.000000
noida	271250.000000
Delhi	262500.000000
New Delhi	255765.306122
Jaipur	252500.000000
Kolkata	249438.775510
Greater Noida	244961.538462
Rewari	240000.000000
mohali	238333.333333
Indore	237708.333333
Bhubaneswar	229318.181818
delhi	201666.666667
Gaziabaad	200000.000000
Manesar	200000.000000
Bhiwadi	150000.000000
Jhansi	120000.000000

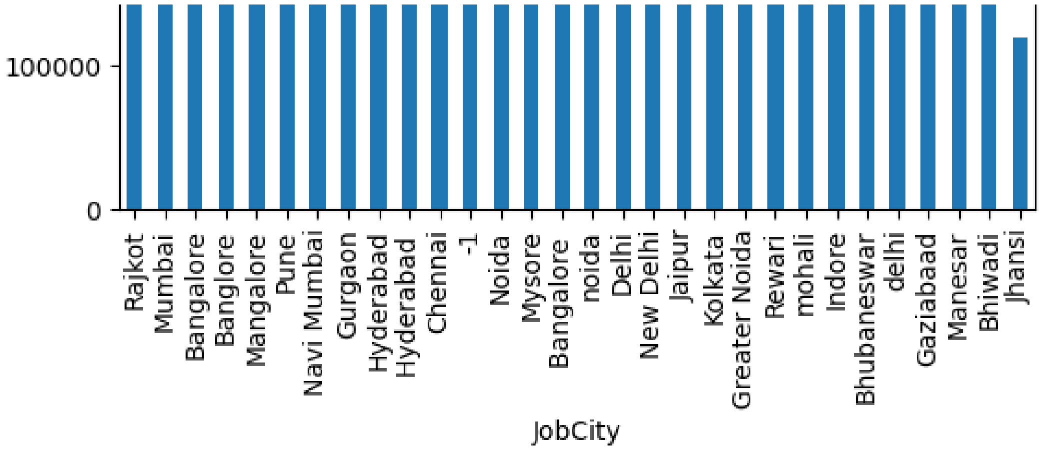
In [128]:

```
df.pivot_table(index='JobCity',values='Salary',aggfunc="mean",sort=False).head(30).sort_
values('Salary',ascending=False).plot(kind="bar")
```

Out[128]:

```
<AxesSubplot:xlabel='JobCity'>
```





Rajkot city having Highest Mean Salary

In [129]:

```
# Gender wise specialization wise total sales
df_sum=df.groupby(['Gender','Specialization'],as_index=False).agg(Total_Salary=("Salary"
,"sum")).sort_values(by=['Gender',"Total_Salary"],ascending=False)
```

In [130]:

```
df_sum
```

Out[130]:

	Gender	Specialization	Total_Salary
47	m	electronics and communication engineering	197555000.0
39	m	computer science & engineering	152700000.0
36	m	computer engineering	150875000.0
58	m	information technology	148710000.0
64	m	mechanical engineering	56999000.0
...	...	...	...
10	f	computer science and technology	320000.0
26	f	telecommunication engineering	300000.0
0	f	aeronautical engineering	180000.0
8	f	computer science	180000.0
4	f	chemical engineering	100000.0

71 rows × 3 columns

In [131]:

```
# In Male electronics and communication engineering Specialization having more Salary
df_sum[df_sum["Gender"]=="m"]
```

Out[131]:

	Gender	Specialization	Total_Salary
47	m	electronics and communication engineering	197555000.0
39	m	computer science & engineering	152700000.0
36	m	computer engineering	150875000.0
58	m	information technology	148710000.0
64	m	mechanical engineering	56999000.0
35	m	computer application	47720000.0

	Gender	Specialization	Total_Salary
49	m	electronics and electrical engineering	44815000.0
46	m	electronics & telecommunications	27600000.0
43	m	electrical engineering	17460000.0
33	m	civil engineering	8845000.0
45	m	electronics & instrumentation eng	7515000.0
50	m	electronics and instrumentation engineering	6705000.0
57	m	information science engineering	5390000.0
51	m	electronics engineering	4250000.0
59	m	instrumentation and control engineering	3775000.0
67	m	other	3465000.0
32	m	chemical engineering	3110000.0
54	m	industrial & production engineering	2960000.0
28	m	applied electronics and instrumentation	2265000.0
70	m	telecommunication engineering	1755000.0
30	m	biotechnology	1590000.0
63	m	mechanical and automation	1545000.0
40	m	computer science and technology	1155000.0
29	m	automobile/automotive engineering	1110000.0
60	m	instrumentation engineering	960000.0
66	m	metallurgical engineering	675000.0
65	m	mechatronics	665000.0
48	m	electronics and computer engineering	660000.0
68	m	polymer technology	655000.0
37	m	computer networking	565000.0
56	m	information science	460000.0
42	m	electrical and power engineering	420000.0
38	m	computer science	400000.0
55	m	industrial engineering	390000.0
61	m	internal combustion engine	360000.0
31	m	ceramic engineering	335000.0
53	m	industrial & management engineering	320000.0
41	m	control and instrumentation engineering	305000.0
27	m	aeronautical engineering	265000.0
52	m	embedded systems technology	200000.0
34	m	computer and communication engineering	120000.0
62	m	mechanical & production engineering	100000.0
69	m	power systems and automation	100000.0
44	m	electronics	40000.0

In [132]:

```
# In Female Computer Engineering Specialization having more Salary
df_sum[df_sum["Gender"]=="f"]
```

Out[132]:

	Gender	Specialization	Total_Salary
7	f	computer engineering	59545000.0

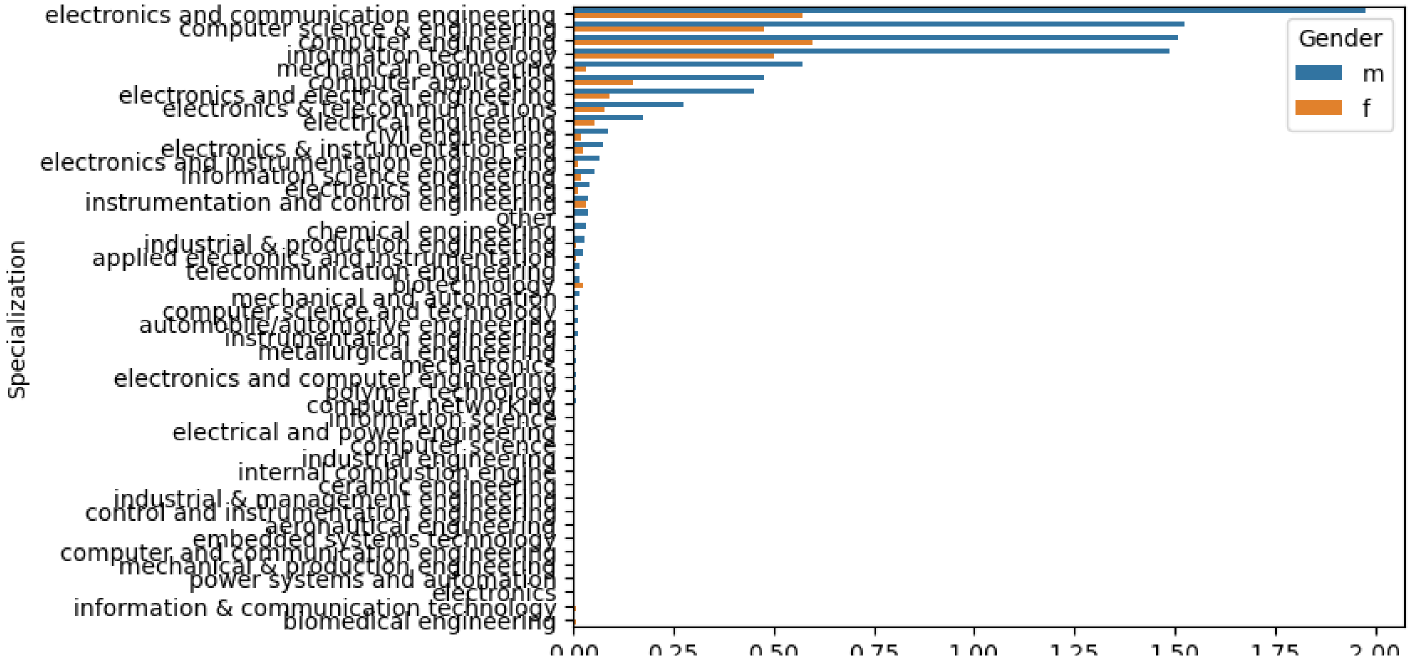
	Gender	Specialization	Total_Salary
14	f	electronics and communication engineering	57215000.0
22	f	information technology	49900000.0
9	f	computer science & engineering	47590000.0
6	f	computer application	15105000.0
15	f	electronics and electrical engineering	8930000.0
13	f	electronics & telecommunications	7920000.0
11	f	electrical engineering	5380000.0
23	f	instrumentation and control engineering	3415000.0
24	f	mechanical engineering	3195000.0
12	f	electronics & instrumentation eng	2505000.0
3	f	biotechnology	2225000.0
21	f	information science engineering	2070000.0
5	f	civil engineering	1845000.0
17	f	electronics engineering	1060000.0
16	f	electronics and instrumentation engineering	1045000.0
18	f	industrial & production engineering	880000.0
20	f	information & communication technology	775000.0
2	f	biomedical engineering	580000.0
1	f	applied electronics and instrumentation	575000.0
19	f	industrial engineering	350000.0
25	f	mechatronics	350000.0
10	f	computer science and technology	320000.0
26	f	telecommunication engineering	300000.0
0	f	aeronautical engineering	180000.0
8	f	computer science	180000.0
4	f	chemical engineering	100000.0

In [133]:

```
sns.barplot(y='Specialization',x='Total_Salary',hue="Gender",data=df_sum)
```

Out[133]:

```
<AxesSubplot:xlabel='Total_Salary', ylabel='Specialization'>
```



In [134]:

```
# CollegeState wise Total Salary
df.pivot_table(index='CollegeState',values='Salary',aggfunc="sum",sort=False).head(30).s
ort_values('Salary',ascending=False)
```

Out[134]:

	Salary
CollegeState	
Uttar Pradesh	258549000.0
Karnataka	118815000.0
Tamil Nadu	99760000.0
Telangana	93325000.0
Maharashtra	74860000.0
Andhra Pradesh	69520000.0
Madhya Pradesh	58400000.0
Delhi	56905000.0
Punjab	56525000.0
West Bengal	53690000.0
Haryana	53200000.0
Orissa	52095000.0
Rajasthan	50250000.0
Uttarakhand	33960000.0
Jharkhand	12460000.0
Kerala	9175000.0
Chhattisgarh	7065000.0
Gujarat	6770000.0
Himachal Pradesh	5125000.0
Bihar	2870000.0
Jammu and Kashmir	2775000.0
Assam	2115000.0
Sikkim	1080000.0
Union Territory	930000.0
Meghalaya	830000.0
Goa	450000.0

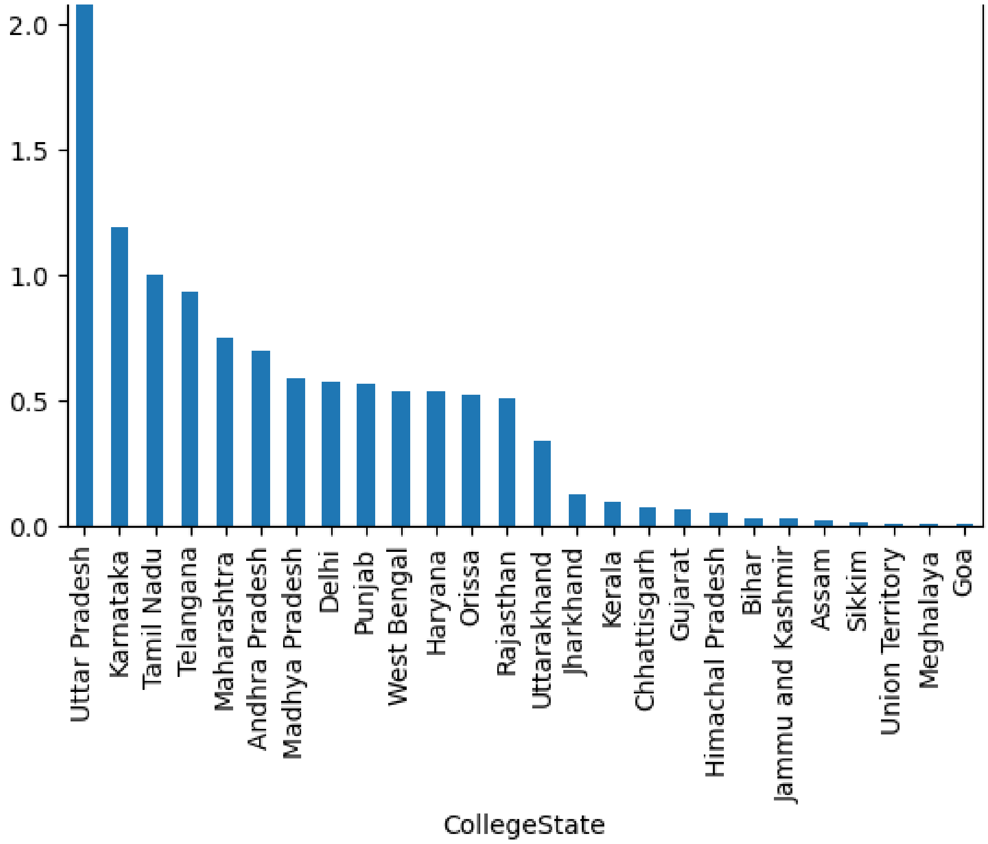
In [135]:

```
df.pivot_table(index='CollegeState',values='Salary',aggfunc="sum",sort=False).head(30).s
ort_values('Salary',ascending=False).plot(kind="bar")
```

Out[135]:

```
<AxesSubplot:xlabel='CollegeState'>
```

1e8
2.5
Salary



from Uttar Pradesh State get more salary

Step - 5 - Research Questions

- Times of India article dated Jan 18, 2019 states that “After doing your Computer Science Engineering if you take up jobs as a Programming Analyst, Software Engineer, Hardware Engineer and Associate Engineer you can earn up to 2.5-3 lakhs as a fresh graduate.” Test this claim with the data given to you.
- Is there a relationship between gender and specialization? (i.e. Does the preference of Specialisation depend on the Gender?)

In [136]:

```
df
```

Out[136]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Comput
0	203097	420000.0	2012-06-01	2024-02-17	senior quality engineer	Bangalore	f	1990-02-19	84.30	board ofsecondary education,ap	...	
1	579905	500000.0	2013-09-01	2024-02-17	assistant manager	Indore	m	1989-10-04	85.40	cbse	...	
2	810601	325000.0	2014-06-01	2024-02-17	systems engineer	Chennai	f	1992-08-03	85.00	cbse	...	
3	267447	655000.0	2011-07-01	2024-02-17	senior software engineer	Gurgaon	m	1989-12-05	85.60	cbse	...	
4	343523	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.00	cbse	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3993	47916	280000.0	2011-10-01	2012-10-01	software engineer	New Delhi	m	1987-04-15	52.09	cbse	...	
3994	752781	100000.0	2013-07-01	2013-07-01	technical writer	Hyderabad	f	1992-08-27	90.00	state board	...	

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	Comput
3995	355888	320000.0	2013-07-01	2014-02-17	associate software engineer	Bangalore	m	1990-07-03	81.86	bse,odisha	...	
3996	947111	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20	78.72	state board	...	
3997	324966	400000.0	2013-02-01	2024-02-17	senior systems engineer	Chennai	f	1991-02-26	70.60	cbse	...	

3998 rows × 38 columns

In [137]:

```
df.columns
```

Out[137]:

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

1.

In [138]:

```
df[df['Specialization']=="computer science & engineering"]
```

Out[138]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ComputerS
6	947847	300000.0	2014-08-01	2015-05-01	java software engineer	Banglore	m	1993-02-01	86.08	state board	...	
18	711342	120000.0	2014-01-01	2014-06-01	data entry operator	Gurgaon	m	1992-12-07	65.00	state board	...	
24	963123	335000.0	2014-06-01	2015-06-01	programmer analyst	Hyderabad	m	1993-06-28	88.00	state board	...	
25	350211	435000.0	2012-09-01	2024-02-17	systems analyst	Gurgaon	f	1991-03-02	86.80	cbse	...	
31	1094324	340000.0	2014-08-01	2015-04-01	software engineer	Bangalore	m	1992-10-23	77.20	state board	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3969	1233826	330000.0	2015-06-01	2024-02-17	technical engineer	pune	m	1993-01-24	76.00	state board	...	
3975	1240207	300000.0	2014-07-01	2015-04-01	game developer	Noida	m	1991-06-03	86.00	cbse	...	
3981	1077872	220000.0	2014-09-01	2024-02-17	software engineer	Gurgaon	m	1991-12-17	53.40	cbse	...	
3989	1204604	300000.0	2014-09-01	2024-02-17	software engineer	Bangalore	m	1991-11-23	74.88	state board	...	
3996	947111	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20	78.72	state board	...	

744 rows × 38 columns

In [139]:

```
df_re=df[df['Specialization']=="computer science & engineering"]
```

In [140]:

```
df_re["Yearr"]=df_re["DOJ"].dt.year
```

```
C:\Users\mitra\AppData\Local\Temp\ipykernel_10960\536175998.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  df_re["Yearr"]=df_re["DOJ"].dt.year
```

In [141]:

```
dff=df_re[df_re["Yearr"]==df_re["GraduationYear"]]
```

In [142]:

```
dff["Experience"]="Fresher"
```

```
C:\Users\mitra\AppData\Local\Temp\ipykernel_10960\345134175.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  dff["Experience"]="Fresher"
```

In [143]:

```
dff
```

Out[143]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEr
6	947847	300000.0	2014-08-01	2015-05-01	java software engineer	Banglore	m	1993-02-01	86.08	state board	...	
24	963123	335000.0	2014-06-01	2015-06-01	programmer analyst	Hyderabad	m	1993-06-28	88.00	state board	...	
25	350211	435000.0	2012-09-01	2024-02-17	systems analyst	Gurgaon	f	1991-03-02	86.80	cbse	...	
31	1094324	340000.0	2014-08-01	2015-04-01	software engineer	Bangalore	m	1992-10-23	77.20	state board	...	
41	955678	145000.0	2014-07-01	2014-09-01	software developer	Delhi	m	1992-04-21	75.00	cbse	...	
...	...	...	...	...	...	...	...	...	...	...	...	
3962	644440	110000.0	2013-09-01	2014-07-01	ui developer	Pondicherry	m	1991-08-07	84.20	state board	...	
3969	1233826	330000.0	2015-06-01	2024-02-17	technical engineer	pune	m	1993-01-24	76.00	state board	...	
3975	1240207	300000.0	2014-07-01	2015-04-01	game developer	Noida	m	1991-06-03	86.00	cbse	...	
3989	1204604	300000.0	2014-09-01	2024-02-17	software engineer	Bangalore	m	1991-11-23	74.88	state board	...	
3996	947111	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20	78.72	state board	...	

452 rows × 40 columns

PROGRAM ANALYST

In [144]:

```
dff[dff["Designation"]=="programmer analyst"]
```

Out[144]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEngg
24	963123	335000.0	2014-06-01	2015-06-01	programmer analyst	Hyderabad	m	1993-06-28	88.00	state board	...	-1
834	1111415	310000.0	2014-08-01	2024-02-17	programmer analyst	Bangalore	f	1992-08-23	85.00	cbse	...	-1
965	963058	335000.0	2014-09-01	2015-04-01	programmer analyst	Hyderabad	m	1993-01-06	85.33	state board	...	-1
1343	913572	305000.0	2014-07-01	2015-04-01	programmer analyst	Coimbatore	m	1992-08-16	79.40	state board	...	-1
1390	823528	305000.0	2014-08-01	2024-02-17	programmer analyst	Bangalore	m	1992-05-18	88.00	cbse	...	-1
1651	913451	330000.0	2014-08-01	2015-04-01	programmer analyst	Chennai	m	1992-10-04	86.00	state board	...	-1
1855	754959	340000.0	2013-08-01	2015-04-01	programmer analyst	-1	m	1991-07-27	87.60	icse	...	-1
1868	1113188	300000.0	2014-12-01	2024-02-17	programmer analyst	Pune	f	1991-09-17	93.30	state board	...	-1
2077	922684	305000.0	2014-09-01	2015-04-01	programmer analyst	Coimbatore	f	1991-05-03	92.00	icse	...	-1
2132	614028	300000.0	2014-08-01	2015-05-01	programmer analyst	Bangalore	m	1993-02-15	89.40	cbse	...	-1
2911	1204221	350000.0	2015-06-01	2024-02-17	programmer analyst	Chennai	m	1994-01-17	84.50	state board	...	-1
2929	829991	325000.0	2014-07-01	2024-02-17	programmer analyst	Bangalore	m	1991-08-17	70.00	icse	...	-1
3429	615310	290000.0	2014-10-01	2024-02-17	programmer analyst	Chennai	m	1993-01-15	70.00	cbse	...	-1
3880	1233727	300000.0	2015-06-01	2024-02-17	programmer analyst	Gurgaon	m	1994-06-30	81.00	cbse	...	-1

14 rows × 40 columns

It proved data from Program Analyst Designation who are graduated recently and also fresher are paid 2.5-3 Lakhs

SOFTWARE ENGINEER

In [145]:

```
dff[dff["Designation"]=="software engineer"]
```

Out[145]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEngg
31	1094324	340000.0	2014-08-01	2015-04-01	software engineer	Bangalore	m	1992-10-23	77.20	state board	...	-1
			2013-	2024-	software			1991-				

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEngg
48	338428	390000.0	2013-[illegible]	2024-02-17	software engineer	Bangalore	m	1991-02-28	86.60	cbse	...	-1
55	989860	250000.0	2014-08-01	2024-02-17	software engineer	Mangalore	m	1992-02-13	90.80	state board	...	-1
115	815219	330000.0	2013-12-01	2015-04-01	software engineer	Chennai	m	1992-01-13	76.17	state board	...	-1
130	902366	325000.0	2014-09-01	2024-02-17	software engineer	Greater Noida	m	1992-01-10	82.80	cbse	...	-1
...	...	...	...	...	...	...	...	...	...	...	...	...
3795	553645	350000.0	2013-11-01	2024-02-17	software engineer	Noida	m	1990-11-08	70.80	cbse	...	-1
3818	1089624	240000.0	2014-02-01	2024-02-17	software engineer	Mumbai	f	1991-09-08	73.80	cbse	...	-1
3881	982135	600000.0	2014-01-01	2024-02-17	software engineer	Bangalore	m	1992-01-31	80.40	jharkhand acedemic council	...	-1
3939	716325	100000.0	2013-07-01	2014-12-01	software engineer	Hyderabad	m	1992-07-05	65.00	state board	...	-1
3989	1204604	300000.0	2014-09-01	2024-02-17	software engineer	Bangalore	m	1991-11-23	74.88	state board	...	-1

82 rows × 40 columns

It proved data from Software Engineer Designation who are graduated recently and also fresher are paid 2.5-3 Lakhs

ASSOCIATE ENGINEER

In [146]:

```
dff[dff["Designation"]=="associate engineer"]
```

Out[146]:

	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEngg	Tele
819	1068402	350000.0	2014-04-01	2024-02-17	associate engineer	Bangalore	m	1993-06-16	74.83	state board	...	-1	

1 rows × 40 columns

It proved data from Associate Engineer Designation who are graduated recently and also fresher are paid 2.5-3 Lakhs

HARDWARE ENGINEER

In [147]:

```
dff[dff["Designation"]=="hardware engineer"]
```

Out[147]:

ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	...	ElectricalEngg	TelecomEngg	CivilEı

0 rows × 40 columns

No one is not from Hardware Engineer

2.

Yes There is a relationship between Gender & Specialization

In [148]:

```
dfff=df.groupby(['Gender','Specialization','DOJ','GraduationYear'],as_index=False).agg(Total_Salary=("Salary","sum")).sort_values(by=['Gender',"Total_Salary"],ascending=False)
```

In [149]:

```
df_summ=df.groupby(['Gender','Specialization'],as_index=False).agg(Total_Salary=("Salary","sum")).sort_values(by=['Gender',"Total_Salary"],ascending=False)
```

In [150]:

```
df_summ
```

Out[150]:

	Gender	Specialization	Total_Salary
47	m	electronics and communication engineering	197555000.0
39	m	computer science & engineering	152700000.0
36	m	computer engineering	150875000.0
58	m	information technology	148710000.0
64	m	mechanical engineering	56999000.0
...	...	...	...
10	f	computer science and technology	320000.0
26	f	telecommunication engineering	300000.0
0	f	aeronautical engineering	180000.0
8	f	computer science	180000.0
4	f	chemical engineering	100000.0

71 rows × 3 columns

In [151]:

```
df_summ[df_summ["Gender"]=="m"]
```

Out[151]:

	Gender	Specialization	Total_Salary
47	m	electronics and communication engineering	197555000.0
39	m	computer science & engineering	152700000.0
36	m	computer engineering	150875000.0
58	m	information technology	148710000.0
64	m	mechanical engineering	56999000.0
35	m	computer application	47720000.0
49	m	electronics and electrical engineering	44915000.0
46	m	electronics & telecommunications	27600000.0
43	m	electrical engineering	17460000.0
33	m	civil engineering	8845000.0

	Gender	Specialization	Total_Salary
33	m	civil engineering	8845000.0
45	m	electronics & instrumentation eng	7515000.0
50	m	electronics and instrumentation engineering	6705000.0
57	m	information science engineering	5390000.0
51	m	electronics engineering	4250000.0
59	m	instrumentation and control engineering	3775000.0
67	m	other	3465000.0
32	m	chemical engineering	3110000.0
54	m	industrial & production engineering	2960000.0
28	m	applied electronics and instrumentation	2265000.0
70	m	telecommunication engineering	1755000.0
30	m	biotechnology	1590000.0
63	m	mechanical and automation	1545000.0
40	m	computer science and technology	1155000.0
29	m	automobile/automotive engineering	1110000.0
60	m	instrumentation engineering	960000.0
66	m	metallurgical engineering	675000.0
65	m	mechatronics	665000.0
48	m	electronics and computer engineering	660000.0
68	m	polymer technology	655000.0
37	m	computer networking	565000.0
56	m	information science	460000.0
42	m	electrical and power engineering	420000.0
38	m	computer science	400000.0
55	m	industrial engineering	390000.0
61	m	internal combustion engine	360000.0
31	m	ceramic engineering	335000.0
53	m	industrial & management engineering	320000.0
41	m	control and instrumentation engineering	305000.0
27	m	aeronautical engineering	265000.0
52	m	embedded systems technology	200000.0
34	m	computer and communication engineering	120000.0
62	m	mechanical & production engineering	100000.0
69	m	power systems and automation	100000.0
44	m	electronics	40000.0

From Gender Male electronics and communication engineering Specialization have more Salary

In [152]:

```
df_summ[df_summ["Gender"]=="f"]
```

Out[152]:

	Gender	Specialization	Total_Salary
7	f	computer engineering	59545000.0
14	f	electronics and communication engineering	57215000.0
22	f	information technology	49900000.0

	Gender	Specialization	Total_Salary
9	f	computer science & engineering	41590000.0
6	f	computer application	15105000.0
15	f	electronics and electrical engineering	8930000.0
13	f	electronics & telecommunications	7920000.0
11	f	electrical engineering	5380000.0
23	f	instrumentation and control engineering	3415000.0
24	f	mechanical engineering	3195000.0
12	f	electronics & instrumentation eng	2505000.0
3	f	biotechnology	2225000.0
21	f	information science engineering	2070000.0
5	f	civil engineering	1845000.0
17	f	electronics engineering	1060000.0
16	f	electronics and instrumentation engineering	1045000.0
18	f	industrial & production engineering	880000.0
20	f	information & communication technology	775000.0
2	f	biomedical engineering	580000.0
1	f	applied electronics and instrumentation	575000.0
19	f	industrial engineering	350000.0
25	f	mechatronics	350000.0
10	f	computer science and technology	320000.0
26	f	telecommunication engineering	300000.0
0	f	aeronautical engineering	180000.0
8	f	computer science	180000.0
4	f	chemical engineering	100000.0

From Gender Female computer engineering engineering Specialization have more Salary

In [153]:

```
df_fema=dfff[dfff["Gender"]=="f"]
```

In [154]:

```
df_fema
```

Out[154]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary
183	f	computer science & engineering	2014-07-01	2014	4500000.0
180	f	computer science & engineering	2014-06-01	2014	3805000.0
185	f	computer science & engineering	2014-08-01	2014	3695000.0
188	f	computer science & engineering	2014-09-01	2014	3085000.0
454	f	information technology	2014-08-01	2014	2585000.0
...	...	...	...	...	...
67	f	computer engineering	2007-02-01	2012	65000.0
363	f	electronics and electrical engineering	2014-06-01	2014	60000.0
136	f	computer engineering	2014-06-01	2014	50000.0
348	f	electronics and electrical engineering	2012-09-01	2012	50000.0
372	f	electronics and instrumentation engineering	2011-11-01	2010	50000.0

	Gender	Specialization	DOJ	GraduationYear	Total_Salary

483 rows × 5 columns

In [155]:

```
df_fema["Yearr"]=df_fema["DOJ"].dt.year
```

```
C:\Users\mitra\AppData\Local\Temp\ipykernel_10960\2767113613.py:1: SettingWithCopyWarning
:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  df_fema["Yearr"]=df_fema["DOJ"].dt.year
```

In [156]:

```
df_fema["GraduationYear"]==df_fema["Yearr"]
```

Out[156]:

```
183     True
180     True
185     True
188     True
454     True
       ...
67     False
363     True
136     True
348     True
372    False
Length: 483, dtype: bool
```

In [157]:

```
df_fema[df_fema["GraduationYear"]==df_fema["Yearr"]]
```

Out[157]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
183	f	computer science & engineering	2014-07-01	2014	4500000.0	2014
180	f	computer science & engineering	2014-06-01	2014	3805000.0	2014
185	f	computer science & engineering	2014-08-01	2014	3695000.0	2014
188	f	computer science & engineering	2014-09-01	2014	3085000.0	2014
454	f	information technology	2014-08-01	2014	2585000.0	2014
...	...	...	...	...	...	...
134	f	computer engineering	2014-05-01	2014	85000.0	2014
261	f	electronics and communication engineering	2010-10-01	2010	75000.0	2010
363	f	electronics and electrical engineering	2014-06-01	2014	60000.0	2014
136	f	computer engineering	2014-06-01	2014	50000.0	2014
348	f	electronics and electrical engineering	2012-09-01	2012	50000.0	2012

240 rows × 6 columns

From Female Gender 240 are Freshers

In [158]:

```
df_fema[df_fema["GraduationYear"]!=df_fema["Yearr"]]
```

Out[158]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
113	f	computer engineering	2013-03-01	2012	2255000.0	2013
309	f	electronics and communication engineering	2014-02-01	2013	2110000.0	2014
89	f	computer engineering	2012-01-01	2011	2010000.0	2012
110	f	computer engineering	2013-01-01	2012	1980000.0	2013
292	f	electronics and communication engineering	2013-04-01	2012	1915000.0	2013
...	...	...	...	...	...	...
276	f	electronics and communication engineering	2012-06-01	2011	85000.0	2012
375	f	electronics and instrumentation engineering	2014-08-01	2013	85000.0	2014
362	f	electronics and electrical engineering	2014-05-01	2013	80000.0	2014
67	f	computer engineering	2007-02-01	2012	65000.0	2007
372	f	electronics and instrumentation engineering	2011-11-01	2010	50000.0	2011

243 rows × 6 columns

From Female Gender 243 have got job after some year of graduation.

In [159]:

```
df_mal=dfff[dfff["Gender"]=="m"]
```

In [160]:

```
df_mal
```

Out[160]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary
805	m	computer science & engineering	2014-07-01	2014	11145000.0
802	m	computer science & engineering	2014-06-01	2014	10980000.0
809	m	computer science & engineering	2014-08-01	2014	8850000.0
665	m	computer engineering	2012-07-01	2012	7085000.0
1070	m	electronics and communication engineering	2014-08-01	2014	6765000.0
...	...	...	...	...	...
610	m	computer application	2014-11-01	2014	60000.0
818	m	computer science & engineering	2014-11-01	2012	60000.0
1460	m	mechanical engineering	2015-05-01	2015	60000.0
1157	m	electronics and electrical engineering	2014-02-01	2012	45000.0
889	m	electronics	2013-10-01	2014	40000.0

1001 rows × 5 columns

In [161]:

```
df_mal["Yearr"]=df_mal["DOJ"].dt.year
```

```
C:\Users\mitra\AppData\Local\Temp\ipykernel_10960\3912209026.py:1: SettingWithCopyWarning
:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  df_mal["Yearr"]=df_mal["DOJ"].dt.year
```

In [162]:

```
df_mal
```

Out[162]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
805	m	computer science & engineering	2014-07-01	2014	11145000.0	2014
802	m	computer science & engineering	2014-06-01	2014	10980000.0	2014
809	m	computer science & engineering	2014-08-01	2014	8850000.0	2014
665	m	computer engineering	2012-07-01	2012	7085000.0	2012
1070	m	electronics and communication engineering	2014-08-01	2014	6765000.0	2014
...	...	...	...	...	...	...
610	m	computer application	2014-11-01	2014	60000.0	2014
818	m	computer science & engineering	2014-11-01	2012	60000.0	2014
1460	m	mechanical engineering	2015-05-01	2015	60000.0	2015
1157	m	electronics and electrical engineering	2014-02-01	2012	45000.0	2014
889	m	electronics	2013-10-01	2014	40000.0	2013

1001 rows × 6 columns

In [163]:

```
df_mal[df_mal["GraduationYear"]==df_mal["Yearr"]]
```

Out[163]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
805	m	computer science & engineering	2014-07-01	2014	11145000.0	2014
802	m	computer science & engineering	2014-06-01	2014	10980000.0	2014
809	m	computer science & engineering	2014-08-01	2014	8850000.0	2014
665	m	computer engineering	2012-07-01	2012	7085000.0	2012
1070	m	electronics and communication engineering	2014-08-01	2014	6765000.0	2014
...	...	...	...	...	...	...
1461	m	mechatronics	2012-06-01	2012	100000.0	2012
619	m	computer engineering	2009-06-01	2009	95000.0	2009
912	m	electronics & telecommunications	2010-09-01	2010	95000.0	2010
610	m	computer application	2014-11-01	2014	60000.0	2014
1460	m	mechanical engineering	2015-05-01	2015	60000.0	2015

459 rows × 6 columns

From Male Gender 459 are Freshers

In [164]:

```
df_mal[df_mal["GraduationYear"]!=df_mal["Yearr"]]
```

Out[164]:

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
788	m	computer science & engineering	2014-02-01	2013	6570000.0	2014
1048	m	electronics and communication engineering	2014-02-01	2013	6415000.0	2014
1051	m	electronics and communication engineering	2014-03-01	2013	6380000.0	2014

	Gender	Specialization	DOJ	GraduationYear	Total_Salary	Yearr
684	m	computer engineering	2013-03-01	2012	5980000.0	2013
1044	m	electronics and communication engineering	2014-01-01	2013	5715000.0	2014
...	...	...	...	...	...	...
565	m	computer application	2012-08-01	2011	85000.0	2012
968	m	electronics and communication engineering	2010-10-01	2014	80000.0	2010
818	m	computer science & engineering	2014-11-01	2012	60000.0	2014
1157	m	electronics and electrical engineering	2014-02-01	2012	45000.0	2014
889	m	electronics	2013-10-01	2014	40000.0	2013

542 rows × 6 columns

From Male Gender 542 have got job after some year of graduation

In [ ]:

In [ ]: