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

In [2]:
```
df = pd.read_csv('student.csv')

df.head()
```

Out[2]:

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

5 rows × 39 columns

In [3]:
```
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
```

In [4]:
```
df.shape
```

Out[4]:
```
(3998, 39)
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Unnamed: 0              0
ID                      0
Salary                  0
DOJ                     0
DOL                     0
Designation             0
JobCity                 0
Gender                  0
DOB                     0
10percentage            0
10board                 0
12graduation            0
12percentage            0
12board                 0
CollegeID               0
CollegeTier             0
Degree                  0
Specialization          0
collegeGPA              0
CollegeCityID           0
CollegeCityTier         0
CollegeState            0
GraduationYear          0
English                 0
Logical                 0
Quant                   0
Domain                  0
ComputerProgramming     0
ElectronicsAndSemicon   0
ComputerScience         0
MechanicalEngg          0
ElectricalEngg          0
TelecomEngg             0
CivilEngg               0
conscientiousness       0
agreeableness           0
extraversion            0
nueroticism             0
openess_to_experience   0
dtype: int64
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

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')
```

In [8]:
```
df.nunique()
```

Out[8]:
```
Unnamed: 0                1
ID                     3998
Salary                  177
DOJ                      81
DOL                      67
Designation             419
JobCity                 339
Gender                    2
DOB                    1872
10percentage            851
10board                 275
12graduation             16
12percentage            801
12board                 340
CollegeID              1350
CollegeTier               2
Degree                    4
Specialization           46
collegeGPA             1282
CollegeCityID          1350
CollegeCityTier           2
CollegeState             26
GraduationYear           11
English                 111
Logical                 107
Quant                   138
Domain                  243
ComputerProgramming      79
ElectronicsAndSemicon    29
ComputerScience          20
MechanicalEngg           42
ElectricalEngg           31
TelecomEngg              26
CivilEngg                23
conscientiousness       141
agreeableness           149
extraversion            154
nueroticism             217
openess_to_experience   142
dtype: int64
```

In [9]:
```
df = df.drop(columns = ['Unnamed: 0', 'ID','CollegeID','CollegeCityID'])

df.head()
```

Out[9]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board
0	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	boarc ofsecondary education,ap
1	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	cbse
2	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	cbse
3	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	cbse
4	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	cbse

5 rows × 35 columns

Datatypes Conversion

In [10]:
```
df['DOL'] = pd.to_datetime(df['DOL'], format='%m/%d/%y %H:%M', errors='coerce')
```

In [11]:
```
df['DOL'] = pd.to_datetime(df['DOL'])
df['DOJ'] = pd.to_datetime(df['DOJ'])
df['DOB'] = pd.to_datetime(df['DOB'])
```

```
C:\Users\sidha\AppData\Local\Temp\ipykernel_28048\2091081124.py:2: UserWarning: Co
uld not infer format, so each element will be parsed individually, falling back to
`dateutil`. To ensure parsing is consistent and as-expected, please specify a form
at.
  df['DOJ'] = pd.to_datetime(df['DOJ'])
C:\Users\sidha\AppData\Local\Temp\ipykernel_28048\2091081124.py:3: UserWarning: Co
uld not infer format, so each element will be parsed individually, falling back to
`dateutil`. To ensure parsing is consistent and as-expected, please specify a form
at.
  df['DOB'] = pd.to_datetime(df['DOB'])
```

In [12]:
```
df.fillna('2015-12-31',inplace=True)
```

In [13]:
```
df.head()
```

Out[13]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.0	cbse	

5 rows × 35 columns

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 35 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Salary                 3998 non-null   float64
 1   DOJ                    3998 non-null   datetime64[ns]
 2   DOL                    3998 non-null   datetime64[ns]
 3   Designation            3998 non-null   object
 4   JobCity                3998 non-null   object
 5   Gender                 3998 non-null   object
 6   DOB                    3998 non-null   datetime64[ns]
 7   10percentage           3998 non-null   float64
 8   10board                3998 non-null   object
 9   12graduation           3998 non-null   int64
 10  12percentage           3998 non-null   float64
 11  12board                3998 non-null   object
 12  CollegeTier            3998 non-null   int64
 13  Degree                 3998 non-null   object
 14  Specialization         3998 non-null   object
 15  collegeGPA             3998 non-null   float64
 16  CollegeCityTier        3998 non-null   int64
 17  CollegeState           3998 non-null   object
 18  GraduationYear         3998 non-null   int64
 19  English                3998 non-null   int64
 20  Logical                3998 non-null   int64
 21  Quant                  3998 non-null   int64
 22  Domain                 3998 non-null   float64
 23  ComputerProgramming    3998 non-null   int64
 24  ElectronicsAndSemicon  3998 non-null   int64
 25  ComputerScience        3998 non-null   int64
 26  MechanicalEngg         3998 non-null   int64
 27  ElectricalEngg         3998 non-null   int64
 28  TelecomEngg            3998 non-null   int64
 29  CivilEngg              3998 non-null   int64
 30  conscientiousness      3998 non-null   float64
 31  agreeableness          3998 non-null   float64
 32  extraversion           3998 non-null   float64
 33  nueroticism            3998 non-null   float64
 34  openess_to_experience  3998 non-null   float64
dtypes: datetime64[ns](3), float64(10), int64(14), object(8)
memory usage: 1.1+ MB
```

In [15]:
```
categorical = ['Designation','JobCity',
               'Gender','10board','12board','CollegeTier','Degree',
               'Specialization','CollegeCityTier','CollegeState']
for cat in categorical:
    df[cat] = df[cat].astype('category')
```

In [16]:
```
df.dtypes
```

Out[16]:
```
Salary                          float64
DOJ                      datetime64[ns]
DOL                      datetime64[ns]
Designation                    category
JobCity                        category
Gender                         category
DOB                      datetime64[ns]
10percentage                    float64
10board                        category
12graduation                      int64
12percentage                    float64
12board                        category
CollegeTier                    category
Degree                         category
Specialization                 category
collegeGPA                      float64
CollegeCityTier                category
CollegeState                   category
GraduationYear                    int64
English                           int64
Logical                           int64
Quant                             int64
Domain                          float64
ComputerProgramming               int64
ElectronicsAndSemicon             int64
ComputerScience                   int64
MechanicalEngg                    int64
ElectricalEngg                    int64
TelecomEngg                       int64
CivilEngg                         int64
conscientiousness               float64
agreeableness                   float64
extraversion                    float64
nueroticism                     float64
openess_to_experience           float64
dtype: object
```

In [17]:
```
df.shape
```

Out[17]:
```
(3998, 35)
```

In [18]:
```
df = df.drop(df[~(df['DOL'] > df['DOJ'])].index)
print(df.shape)
```

```
(3943, 35)
```

In [19]:
```
print((df['10percentage'] <=10).sum())
print((df['12percentage'] <=10).sum())
print((df['collegeGPA'] <=10).sum())
```

```
0
0
12
```

In [20]:
```
df.loc[df['collegeGPA']<=10,'collegeGPA'] = (df.loc[df['collegeGPA']<=10,'collegeGP
df.head()
```

Out[20]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.0	cbse	

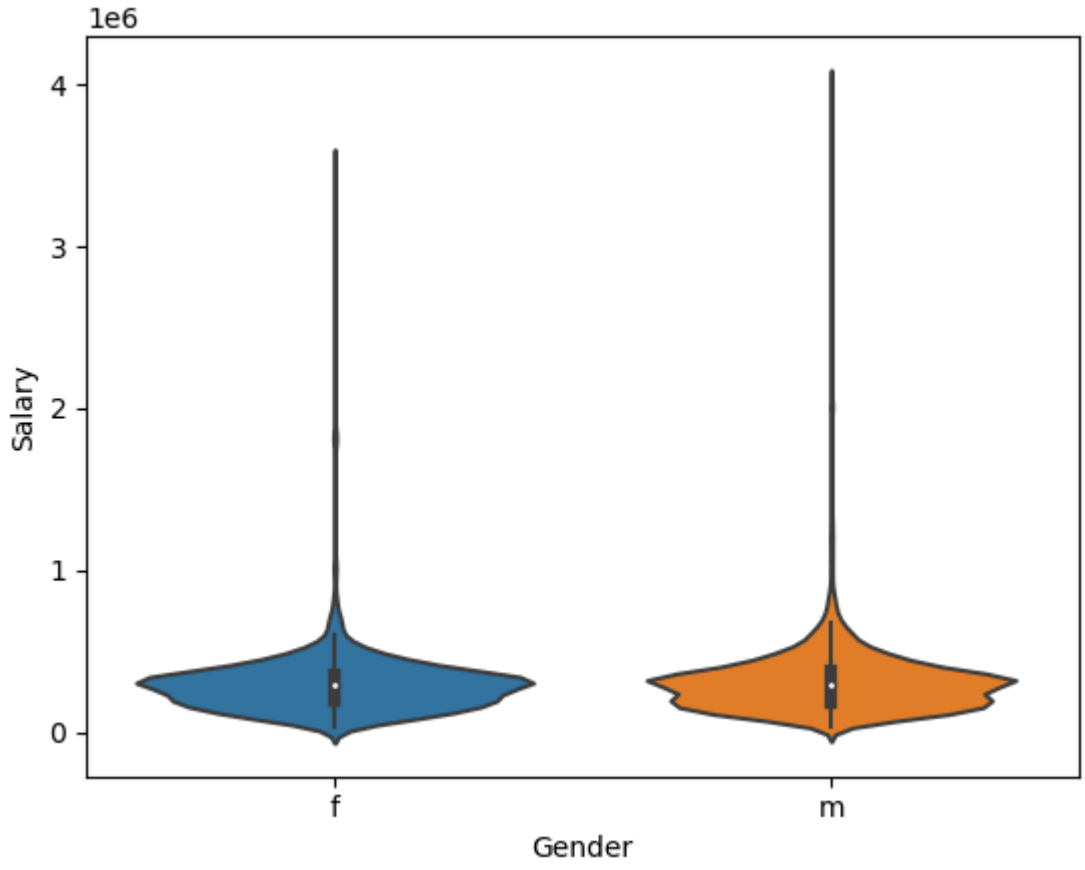
5 rows × 35 columns

In [21]:
```
print((df==0).sum()[(df==0).sum() > 0])
```

```
CollegeCityTier    2761
GraduationYear        1
dtype: int64
```

In [22]:
```
sns.violinplot(x='Gender', y='Salary', data=df)
```

Out[22]:
```
<Axes: xlabel='Gender', ylabel='Salary'>
```



In [23]:
```
df[['10percentage', '12percentage', 'collegeGPA',
          'Gender']].groupby('Gender').mean()
```

Out[23]:

Gender	10percentage	12percentage	collegeGPA
f	81.003270	77.068741	74.182847
m	76.983042	73.637638	70.916618

In [24]:
```
df[['10percentage', '12percentage', 'collegeGPA',
          'Gender']].groupby('Gender').median()
```

Out[24]:

Gender	10percentage	12percentage	collegeGPA
f	82.46	77.0	74.00
m	78.00	73.4	70.66

In [25]:
```
df[['conscientiousness', 'agreeableness', 'extraversion',
          'nueroticism', 'openess_to_experience', 'Gender']].groupby('Gender').mear
```

Out[25]:

Gender	conscientiousness	agreeableness	extraversion	nueroticism	openess_to_experience
f	0.120766	0.294788	0.008161	-0.187087	0.044733
m	-0.089991	0.099242	-0.004356	-0.164979	-0.197349

In [26]:
```
df[['Salary', 'Gender']].groupby('Gender').mean()
```

Out[26]:

Gender	Salary
f	296190.476190
m	312059.372915

In [27]:
```
df[['Salary', 'Gender']].groupby('Gender').median()
```

Out[27]:

Gender	Salary
f	300000.0
m	300000.0

In [28]:
```
print(df[['Salary', 'Gender']].groupby("Gender").max())
print(df[['Salary', 'Gender']].groupby('Gender').min())
```

```
           Salary
Gender
f       3500000.0
m       4000000.0
         Salary
Gender
f       35000.0
m       35000.0
```

In [29]:
```
df = df.drop(columns = ['MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
                        'CivilEngg'])
df.head()
```

Out[29]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.0	cbse	

5 rows × 31 columns

In [30]:
```
df['10board'] = df['10board'].astype(str)
df['12board'] = df['12board'].astype(str)
df['JobCity'] = df['JobCity'].astype(str)
```

In [31]:
```
df['10board'] = df['10board'].replace({'0':np.nan})
df['12board'] = df['12board'].replace({'0':np.nan})
df['GraduationYear'] = df['GraduationYear'].replace({0:np.nan})
df['JobCity'] = df['JobCity'].replace({'-1':np.nan})
df['Domain'] = df['Domain'].replace({-1:np.nan})
df['ElectronicsAndSemicon'] = df['ElectronicsAndSemicon'].replace({-1:0})
df['ComputerScience'] = df['ComputerScience'].replace({-1:0})
df['ComputerProgramming'] = df['ComputerProgramming'].replace({-1:np.nan})
```

In [32]:
```
df['10board'] = df['10board'].astype('category')
df['12board'] = df['12board'].astype('category')
df['JobCity'] = df['JobCity'].astype('category')
```

In [33]:
```
df['10board'].fillna(df['10board'].mode()[0], inplace = True)
df['12board'].fillna(df['12board'].mode()[0], inplace = True)
df['GraduationYear'].fillna(df['GraduationYear'].mode()[0], inplace = True)
df['JobCity'].fillna(df['JobCity'].mode()[0], inplace = True)
df
```

Out[33]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	1(
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	Bangalore	f	1990-02-19	84.30	ofsec educa
1	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.40	
2	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.00	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.60	
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.00	
...	...	...	...	...	...	...	...	...	
3992	800000.0	2014-04-01	2015-04-01	manager	Rajkot	m	1990-06-22	73.00	
3993	280000.0	2011-10-01	2012-10-01	software engineer	New Delhi	m	1987-04-15	52.09	
3995	320000.0	2013-07-01	2015-12-31	associate software engineer	Bangalore	m	1991-07-03	81.86	bse
3996	200000.0	2014-07-01	2015-01-01	software developer	Asifabadbanglore	f	1992-03-20	78.72	state
3997	400000.0	2013-02-01	2015-12-31	senior systems engineer	Chennai	f	1991-02-26	70.60	

3943 rows × 31 columns

In [34]:
```
df['Domain'].fillna(df['Domain'].median(), inplace = True)
df['ComputerProgramming'].fillna(df['ComputerProgramming'].median(),
                                 inplace = True)
df.head()
```

Out[34]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	Bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	Indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	Chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	Gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	Manesar	m	1991-02-27	78.0	cbse	

5 rows × 31 columns

In [35]:
```
def correct_string_data(data):
    df[data] = df[data].str.lower().str.strip()
```

In [36]:
```
textual_columns = ['Designation','JobCity','10board','12board','Specialization','Co
```

In [37]:
```
for col in textual_columns:
    print(f'Number of unique values in {col} with inconsistency : {df[col].nunique(
```

```
Number of unique values in Designation with inconsistency : 416
Number of unique values in JobCity with inconsistency : 337
Number of unique values in 10board with inconsistency : 274
Number of unique values in 12board with inconsistency : 339
Number of unique values in Specialization with inconsistency : 46
Number of unique values in CollegeState with inconsistency : 26
```

In [38]:
```
for col in textual_columns:
    correct_string_data(col)
```

In [39]:
```
for col in textual_columns:
    print(f'Number of unique values in {col} without inconsistency : {df[col].nunic
```

```
Number of unique values in Designation without inconsistency : 416
Number of unique values in JobCity without inconsistency : 230
Number of unique values in 10board without inconsistency : 272
Number of unique values in 12board without inconsistency : 336
Number of unique values in Specialization without inconsistency : 46
Number of unique values in CollegeState without inconsistency : 26
```

In [40]:
```
df['DOB'] = pd.to_datetime(df['DOB'])
df['Age'] = 2015 - df['DOB'].dt.year
df.head()
```

Out[40]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	manesar	m	1991-02-27	78.0	cbse	

5 rows × 32 columns

In [41]:
```
delta = (df['DOL'] - df['DOJ'])
tenure = np.zeros(len(df))
for i, date in enumerate(delta):
    tenure[i] = round(date.days/365,2)
df['Tenure'] = tenure

df.head()
```

Out[41]:

	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	10board	12
0	420000.0	2012-06-01	2015-12-31	senior quality engineer	bangalore	f	1990-02-19	84.3	board ofsecondary education,ap	
1	500000.0	2013-09-01	2015-12-31	assistant manager	indore	m	1989-10-04	85.4	cbse	
2	325000.0	2014-06-01	2015-12-31	systems engineer	chennai	f	1992-08-03	85.0	cbse	
3	1100000.0	2011-07-01	2015-12-31	senior software engineer	gurgaon	m	1989-12-05	85.6	cbse	
4	200000.0	2014-03-01	2015-03-01	get	manesar	m	1991-02-27	78.0	cbse	

5 rows × 33 columns

In [42]:
```
df = df.drop(df[(df['GraduationYear'] > df['DOJ'].dt.year)].index)
```

In [43]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3864 entries, 0 to 3997
Data columns (total 33 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Salary                 3864 non-null   float64
 1   DOJ                    3864 non-null   datetime64[ns]
 2   DOL                    3864 non-null   datetime64[ns]
 3   Designation            3864 non-null   object
 4   JobCity                3864 non-null   object
 5   Gender                 3864 non-null   category
 6   DOB                    3864 non-null   datetime64[ns]
 7   10percentage           3864 non-null   float64
 8   10board                3864 non-null   object
 9   12graduation           3864 non-null   int64
 10  12percentage           3864 non-null   float64
 11  12board                3864 non-null   object
 12  CollegeTier            3864 non-null   category
 13  Degree                 3864 non-null   category
 14  Specialization         3864 non-null   object
 15  collegeGPA             3864 non-null   float64
 16  CollegeCityTier        3864 non-null   category
 17  CollegeState           3864 non-null   object
 18  GraduationYear         3864 non-null   float64
 19  English                3864 non-null   int64
 20  Logical                3864 non-null   int64
 21  Quant                  3864 non-null   int64
 22  Domain                 3864 non-null   float64
 23  ComputerProgramming    3864 non-null   float64
 24  ElectronicsAndSemicon  3864 non-null   int64
 25  ComputerScience        3864 non-null   int64
 26  conscientiousness      3864 non-null   float64
 27  agreeableness          3864 non-null   float64
 28  extraversion           3864 non-null   float64
 29  nueroticism            3864 non-null   float64
 30  openess_to_experience  3864 non-null   float64
 31  Age                    3864 non-null   int32
 32  Tenure                 3864 non-null   float64
dtypes: category(4), datetime64[ns](3), float64(13), int32(1), int64(6), object(6)
memory usage: 906.2+ KB
```

Univariate Analysis

In [44]:
```
discrete_df = df.select_dtypes(include=['object'])

numerical_df = df.select_dtypes(include=['int64', 'float64','int32'])
```

In [45]:
```
def discrete_univariate_analysis(discrete_data):
    for col_name in discrete_data:
        print("*"*10, col_name, "*"*10)
        print(discrete_data[col_name].agg(['count', 'nunique', 'unique']))
        print('Top 10 Value Counts: \n', discrete_data[col_name].value_counts().hea
        print()
```

In [46]:
```
discrete_univariate_analysis(discrete_df)
```

```
********** Designation **********
count                                                    3864
nunique                                                   413
unique     [senior quality engineer, assistant manager, s...
Name: Designation, dtype: object
Top 10 Value Counts:
 Designation
software engineer             525
software developer            258
system engineer               201
programmer analyst            137
systems engineer              116
java software engineer        108
software test engineer         98
project engineer               73
technical support engineer     72
senior software engineer       71
Name: count, dtype: int64

********** JobCity **********
count                                                    3864
nunique                                                   222
unique     [bangalore, indore, chennai, gurgaon, manesar,...
Name: JobCity, dtype: object
Top 10 Value Counts:
 JobCity
bangalore    1085
noida         375
hyderabad     356
pune          318
chennai       310
gurgaon       209
new delhi     198
mumbai        119
kolkata       116
jaipur         49
Name: count, dtype: int64

********** 10board **********
count                                                    3864
nunique                                                   269
unique     [board ofsecondary education,ap, cbse, state b...
Name: 10board, dtype: object
Top 10 Value Counts:
 10board
cbse                           1688
state board                    1116
icse                            271
ssc                             121
up board                         83
matriculation                    38
rbse                             21
board of secondary education     20
up                               18
mp board                         17
Name: count, dtype: int64

********** 12board **********
count                                                    3864
nunique                                                   332
unique     [board of intermediate education,ap, cbse, sta...
Name: 12board, dtype: object
Top 10 Value Counts:
 12board
```

```
cbse                                1697
state board                         1206
icse                                 127
up board                              85
isc                                   44
board of intermediate                 37
board of intermediate education       31
up                                    19
rbse                                  17
mp board                              17
Name: count, dtype: int64

********** Specialization **********
count                                                3864
nunique                                                42
unique     [computer engineering, electronics and communi...
Name: Specialization, dtype: object
Top 10 Value Counts:
 Specialization
electronics and communication engineering    856
computer science & engineering               714
information technology                       649
computer engineering                         582
computer application                         232
mechanical engineering                       194
electronics and electrical engineering       185
electronics & telecommunications             119
electrical engineering                        79
electronics & instrumentation eng             32
Name: count, dtype: int64

********** CollegeState **********
count                                                3864
nunique                                                26
unique     [andhra pradesh, madhya pradesh, uttar pradesh...
Name: CollegeState, dtype: object
Top 10 Value Counts:
 CollegeState
uttar pradesh     888
tamil nadu        359
karnataka         359
telangana         307
maharashtra       252
andhra pradesh    219
west bengal       188
madhya pradesh    187
punjab            177
haryana           174
Name: count, dtype: int64
```

In [47]:
```
def numerical_univariate_analysis(numerical_data):
    for col_name in numerical_data:
        print("*"*10, col_name, "*"*10)
        print(numerical_data[col_name].agg(['min', 'max', 'mean', 'median', 'std']))
        print()
```

In [48]:
```
numerical_univariate_analysis(numerical_df)
```

```
********** Salary **********
min       3.500000e+04
max       4.000000e+06
mean      3.093838e+05
median    3.000000e+05
std       2.125428e+05
Name: Salary, dtype: float64

********** 10percentage **********
min       43.000000
max       97.760000
mean      77.974503
median    79.200000
std        9.832284
Name: 10percentage, dtype: float64

********** 12graduation **********
min       1998.000000
max       2013.000000
mean      2008.072723
median    2008.000000
std          1.634833
Name: 12graduation, dtype: float64

********** 12percentage **********
min       40.000000
max       98.700000
mean      74.514772
median    74.400000
std       11.008297
Name: 12percentage, dtype: float64

********** collegeGPA **********
min       49.070000
max       99.930000
mean      71.697945
median    71.775000
std        7.412470
Name: collegeGPA, dtype: float64

********** GraduationYear **********
min       2007.000000
max       2015.000000
mean      2012.562629
median    2013.000000
std          1.285620
Name: GraduationYear, dtype: float64

********** English **********
min       180.000000
max       875.000000
mean      501.591097
median    500.000000
std       104.509765
Name: English, dtype: float64

********** Logical **********
min       195.000000
max       795.000000
mean      501.652950
median    505.000000
std        86.555756
Name: Logical, dtype: float64
```

```
********** Quant **********
min       120.000000
max       900.000000
mean      513.717133
median    515.000000
std       122.171597
Name: Quant, dtype: float64

********** Domain **********
min       0.002750
max       0.999910
mean      0.612619
median    0.649390
std       0.264916
Name: Domain, dtype: float64

********** ComputerProgramming **********
min       115.000000
max       840.000000
mean      452.441511
median    455.000000
std        85.997659
Name: ComputerProgramming, dtype: float64

********** ElectronicsAndSemicon **********
min         0.000000
max       612.000000
mean       96.441253
median      0.000000
std       158.045705
Name: ElectronicsAndSemicon, dtype: float64

********** ComputerScience **********
min         0.000000
max       715.000000
mean       90.826863
median      0.000000
std       174.661705
Name: ComputerScience, dtype: float64

********** conscientiousness **********
min      -4.12670
max       1.99530
mean     -0.03976
median    0.04640
std       1.02725
Name: conscientiousness, dtype: float64

********** agreeableness **********
min      -5.781600
max       1.904800
mean      0.146948
median    0.212400
std       0.940645
Name: agreeableness, dtype: float64

********** extraversion **********
min      -4.600900
max       2.535400
mean     -0.002940
median    0.091400
std       0.952482
Name: extraversion, dtype: float64
```

```
********** nueroticism **********
min      -2.643000
max       3.352500
mean     -0.167970
median   -0.234400
std       1.006697
Name: nueroticism, dtype: float64

********** openess_to_experience **********
min      -7.375700
max       1.822400
mean     -0.139965
median   -0.094300
std       1.005369
Name: openess_to_experience, dtype: float64

********** Age **********
min       18.000000
max       34.000000
mean      24.584627
median    24.000000
std        1.750436
Name: Age, dtype: float64

********** Tenure **********
min       0.080000
max       5.840000
mean      1.747741
median    1.500000
std       1.132959
Name: Tenure, dtype: float64
```

In [49]:
```
numerical_df.columns
```

Out[49]:
```
Index(['Salary', '10percentage', '12graduation', '12percentage', 'collegeGPA',
       'GraduationYear', 'English', 'Logical', 'Quant', 'Domain',
       'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
       'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience', 'Age', 'Tenure'],
      dtype='object')
```

In [50]:
```
discrete_num_cols = ['Salary', '10percentage', '12graduation', '12percentage', 'col
       'GraduationYear', 'English', 'Logical', 'Quant', 'Domain',
       'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
       'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience', 'Age', 'Tenure']
numerical_df.drop(columns=discrete_num_cols, axis=1, inplace=True)

print('Shape:', numerical_df.shape)
print('Columns:', numerical_df.columns)
```

```
Shape: (3864, 0)
Columns: Index([], dtype='object')
```

In [51]:
```
discrete_num_df = df[discrete_num_cols]

print('Shape:', discrete_num_df.shape)
print('Columns:', discrete_num_df.columns)
```

```
Shape: (3864, 20)
Columns: Index(['Salary', '10percentage', '12graduation', '12percentage', 'college
GPA',
       'GraduationYear', 'English', 'Logical', 'Quant', 'Domain',
       'ComputerProgramming', 'ElectronicsAndSemicon', 'ComputerScience',
       'conscientiousness', 'agreeableness', 'extraversion', 'nueroticism',
       'openess_to_experience', 'Age', 'Tenure'],
      dtype='object')
```

In [52]:
```
numerical_univariate_analysis(numerical_df)
```

In [53]:
```
discrete_univariate_analysis(discrete_df)
```

```
********** Designation **********
count                                                    3864
nunique                                                   413
unique     [senior quality engineer, assistant manager, s...
Name: Designation, dtype: object
Top 10 Value Counts:
 Designation
software engineer             525
software developer            258
system engineer               201
programmer analyst            137
systems engineer              116
java software engineer        108
software test engineer         98
project engineer               73
technical support engineer     72
senior software engineer       71
Name: count, dtype: int64

********** JobCity **********
count                                                    3864
nunique                                                   222
unique     [bangalore, indore, chennai, gurgaon, manesar,...
Name: JobCity, dtype: object
Top 10 Value Counts:
 JobCity
bangalore    1085
noida         375
hyderabad     356
pune          318
chennai       310
gurgaon       209
new delhi     198
mumbai        119
kolkata       116
jaipur         49
Name: count, dtype: int64

********** 10board **********
count                                                    3864
nunique                                                   269
unique     [board ofsecondary education,ap, cbse, state b...
Name: 10board, dtype: object
Top 10 Value Counts:
 10board
cbse                           1688
state board                    1116
icse                            271
ssc                             121
up board                         83
matriculation                    38
rbse                             21
board of secondary education     20
up                               18
mp board                         17
Name: count, dtype: int64

********** 12board **********
count                                                    3864
nunique                                                   332
unique     [board of intermediate education,ap, cbse, sta...
Name: 12board, dtype: object
Top 10 Value Counts:
 12board
```

```
cbse                                1697
state board                         1206
icse                                 127
up board                              85
isc                                   44
board of intermediate                 37
board of intermediate education       31
up                                    19
rbse                                  17
mp board                              17
Name: count, dtype: int64

********** Specialization **********
count                                                3864
nunique                                                42
unique     [computer engineering, electronics and communi...
Name: Specialization, dtype: object
Top 10 Value Counts:
 Specialization
electronics and communication engineering    856
computer science & engineering               714
information technology                       649
computer engineering                         582
computer application                         232
mechanical engineering                       194
electronics and electrical engineering       185
electronics & telecommunications             119
electrical engineering                        79
electronics & instrumentation eng             32
Name: count, dtype: int64

********** CollegeState **********
count                                                3864
nunique                                                26
unique     [andhra pradesh, madhya pradesh, uttar pradesh...
Name: CollegeState, dtype: object
Top 10 Value Counts:
 CollegeState
uttar pradesh     888
tamil nadu        359
karnataka         359
telangana         307
maharashtra       252
andhra pradesh    219
west bengal       188
madhya pradesh    187
punjab            177
haryana           174
Name: count, dtype: int64
```

In [54]:
```
discrete_univariate_analysis(discrete_num_df)
```

```
********** Salary **********
count                                                  3864
nunique                                                 174
unique     [420000.0, 500000.0, 325000.0, 1100000.0, 2000...
Name: Salary, dtype: object
Top 10 Value Counts:
 Salary
300000.0    286
180000.0    227
200000.0    198
325000.0    185
120000.0    154
240000.0    152
400000.0    126
350000.0    122
100000.0    103
150000.0     84
Name: count, dtype: int64

********** 10percentage **********
count                                                  3864
nunique                                                 830
unique     [84.3, 85.4, 85.0, 85.6, 78.0, 89.92, 86.08, 9...
Name: 10percentage, dtype: object
Top 10 Value Counts:
 10percentage
78.0    76
82.0    69
80.0    65
86.0    64
85.0    63
73.0    63
76.0    62
75.0    62
72.0    61
87.0    59
Name: count, dtype: int64

********** 12graduation **********
count                                                  3864
nunique                                                  15
unique     [2007, 2010, 2008, 2009, 2006, 2011, 2005, 200...
Name: 12graduation, dtype: object
Top 10 Value Counts:
 12graduation
2009    1013
2008     907
2010     710
2007     518
2006     403
2005     156
2004      71
2011      35
2003      23
2002      13
Name: count, dtype: int64

********** 12percentage **********
count                                                  3864
nunique                                                 789
unique     [95.8, 85.0, 68.2, 83.6, 76.8, 87.0, 67.5, 91....
Name: 12percentage, dtype: object
Top 10 Value Counts:
 12percentage
```

```
70.0    70
72.0    66
74.0    58
62.0    57
65.0    55
68.0    55
76.0    55
64.0    54
78.0    53
61.0    52
Name: count, dtype: int64

********** collegeGPA **********
count                                                  3864
nunique                                                1251
unique     [78.0, 70.06, 70.0, 74.64, 73.9, 76.32, 72.98,...
Name: collegeGPA, dtype: object
Top 10 Value Counts:
 collegeGPA
70.0    106
72.0     93
75.0     82
65.0     79
68.0     71
71.0     69
73.0     68
76.0     67
78.0     66
74.0     66
Name: count, dtype: int64

********** GraduationYear **********
count                                                  3864
nunique                                                   8
unique     [2011.0, 2012.0, 2014.0, 2013.0, 2010.0, 2015....
Name: GraduationYear, dtype: object
Top 10 Value Counts:
 GraduationYear
2013.0    1152
2014.0     998
2012.0     835
2011.0     501
2010.0     291
2015.0      62
2009.0      24
2007.0       1
Name: count, dtype: int64

********** English **********
count                                                  3864
nunique                                                 110
unique     [515, 695, 615, 635, 545, 560, 590, 605, 565, ...
Name: English, dtype: object
Top 10 Value Counts:
 English
475    155
465    148
545    144
535    137
405    110
485    106
395     96
455     95
525     95
```

```
500     92
Name: count, dtype: int64

********** Logical **********
count                                                  3864
nunique                                                 107
unique     [585, 610, 545, 625, 555, 435, 670, 565, 455, ...
Name: Logical, dtype: object
Top 10 Value Counts:
 Logical
495    153
485    151
555    145
545    144
505    113
475    110
425    110
435    107
525    105
605     99
Name: count, dtype: int64

********** Quant **********
count                                                  3864
nunique                                                 136
unique     [525, 780, 370, 625, 465, 620, 380, 530, 545, ...
Name: Quant, dtype: object
Top 10 Value Counts:
 Quant
605    139
485    124
545    122
575    114
515     98
415     96
475     93
500     92
445     86
535     83
Name: count, dtype: int64

********** Domain **********
count                                                  3864
nunique                                                 238
unique     [0.6359787565, 0.960603252, 0.4508765845, 0.97...
Name: Domain, dtype: object
Top 10 Value Counts:
 Domain
0.649390    294
0.622643    110
0.538387    108
0.486747    102
0.376060     99
0.744758     97
0.356536     95
0.694479     95
0.824666     81
0.229482     81
Name: count, dtype: int64

********** ComputerProgramming **********
count                                                  3864
nunique                                                  76
unique     [445.0, 455.0, 395.0, 615.0, 645.0, 405.0, 735...
```

```
Name: ComputerProgramming, dtype: object
Top 10 Value Counts:
 ComputerProgramming
455.0    969
445.0    143
435.0    140
475.0    138
465.0    128
395.0    119
495.0    113
485.0    112
525.0    110
405.0    108
Name: count, dtype: int64

********** ElectronicsAndSemicon **********
count                                                 3864
nunique                                                 29
unique     [0, 466, 233, 366, 324, 266, 333, 356, 420, 26...
Name: ElectronicsAndSemicon, dtype: object
Top 10 Value Counts:
 ElectronicsAndSemicon
0      2754
333     122
300     108
366     102
266      88
400      82
292      71
356      64
324      62
233      52
Name: count, dtype: int64

********** ComputerScience **********
count                                                 3864
nunique                                                 20
unique     [0, 407, 346, 376, 500, 438, 315, 253, 469, 19...
Name: ComputerScience, dtype: object
Top 10 Value Counts:
 ComputerScience
0      3001
407     125
376     119
346     111
438     105
469      75
315      74
500      62
284      46
530      43
Name: count, dtype: int64

********** conscientiousness **********
count                                                 3864
nunique                                                141
unique     [0.9737, -0.7335, 0.2718, 0.0464, -0.881, -0.3...
Name: conscientiousness, dtype: object
Top 10 Value Counts:
 conscientiousness
 0.2718    141
-0.1590    128
 0.1282    126
 0.4155    123
```

```
 0.5591    122
-0.0154    121
 0.8463    117
-0.3027    116
-0.4463    110
 0.9900    103
Name: count, dtype: int64

********** agreeableness **********
count                                                  3864
nunique                                                 148
unique     [0.8128, 0.3789, 1.7109, 0.3448, -0.2793, -0.6...
Name: agreeableness, dtype: object
Top 10 Value Counts:
 agreeableness
 0.3789    187
 0.2124    174
 0.5454    171
 0.0459    159
 0.7119    151
 0.8784    150
 1.0449    138
-0.1206    137
-0.2871    129
 1.2114    120
Name: count, dtype: int64

********** extraversion **********
count                                                  3864
nunique                                                 153
unique     [0.5269, 1.2396, 0.1637, -0.344, -1.0697, -2.2...
Name: extraversion, dtype: object
Top 10 Value Counts:
 extraversion
 0.3174    171
 0.4711    170
 0.1637    150
 0.7785    142
 0.6248    129
-0.1437    127
 0.0100    122
-0.2974    121
 0.9322    112
 1.0859    104
Name: count, dtype: int64

********** nueroticism **********
count                                                  3864
nunique                                                 217
unique     [1.3549, -0.1076, -0.8682, -0.4078, 0.09163, -...
Name: nueroticism, dtype: object
Top 10 Value Counts:
 nueroticism
-0.4879    120
-0.7415    112
 0.0192    109
-0.6147    103
-0.3612    102
-0.2344     99
-0.1076     97
-0.8682     96
 0.2727     95
-0.9950     93
Name: count, dtype: int64
```

```
********** openess_to_experience **********
count                                                 3864
nunique                                                141
unique     [-0.4455, 0.8637, 0.6721, -0.9194, -0.1295, -0...
Name: openess_to_experience, dtype: object
Top 10 Value Counts:
 openess_to_experience
-0.0943    178
 0.6721    177
 0.0973    177
 0.2889    170
 0.4805    169
-0.2859    154
 0.8637    152
-0.6692    143
 1.0554    128
-0.2875    123
Name: count, dtype: int64

********** Age **********
count                                                 3864
nunique                                                 16
unique     [25, 26, 23, 24, 22, 28, 27, 29, 21, 30, 18, 3...
Name: Age, dtype: object
Top 10 Value Counts:
 Age
24    940
23    815
25    760
26    513
27    302
22    287
28    113
29     57
21     28
30     27
Name: count, dtype: int64

********** Tenure **********
count                                                 3864
nunique                                                110
unique     [3.58, 2.33, 1.58, 4.5, 1.0, 0.75, 2.5, 1.5, 4...
Name: Tenure, dtype: object
Top 10 Value Counts:
 Tenure
1.00    188
1.50    162
1.58    129
1.42    126
1.08    119
0.75    118
1.33    114
1.25    112
2.00    112
0.50    112
Name: count, dtype: int64
```

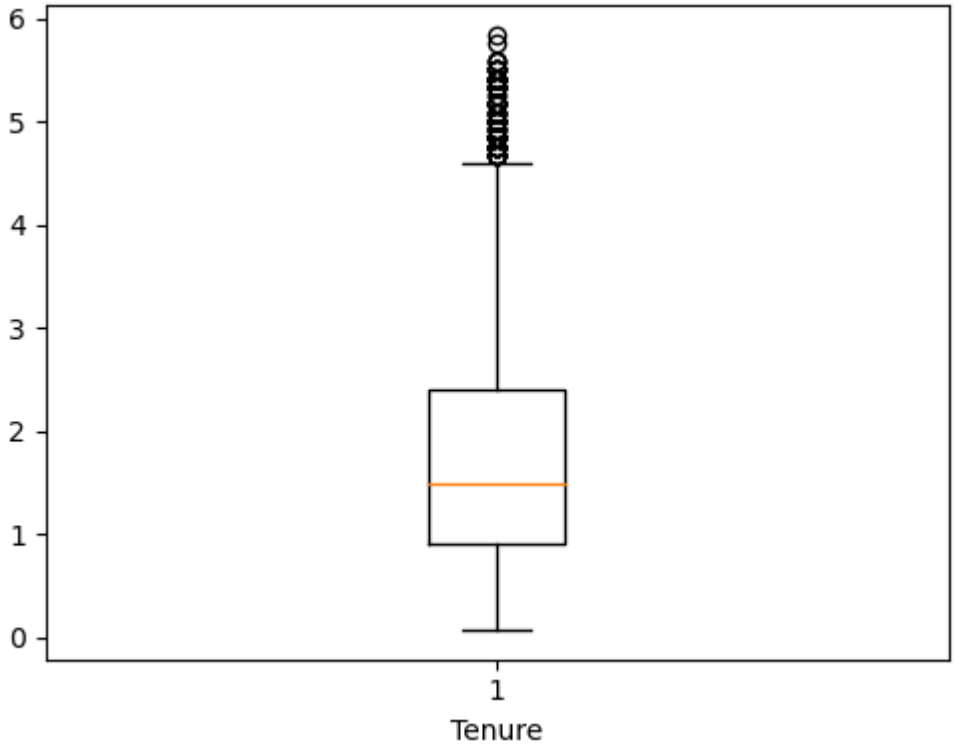
Univariate - Visual Analysis(Continuous Features)

In [55]:
```
df.shape
```

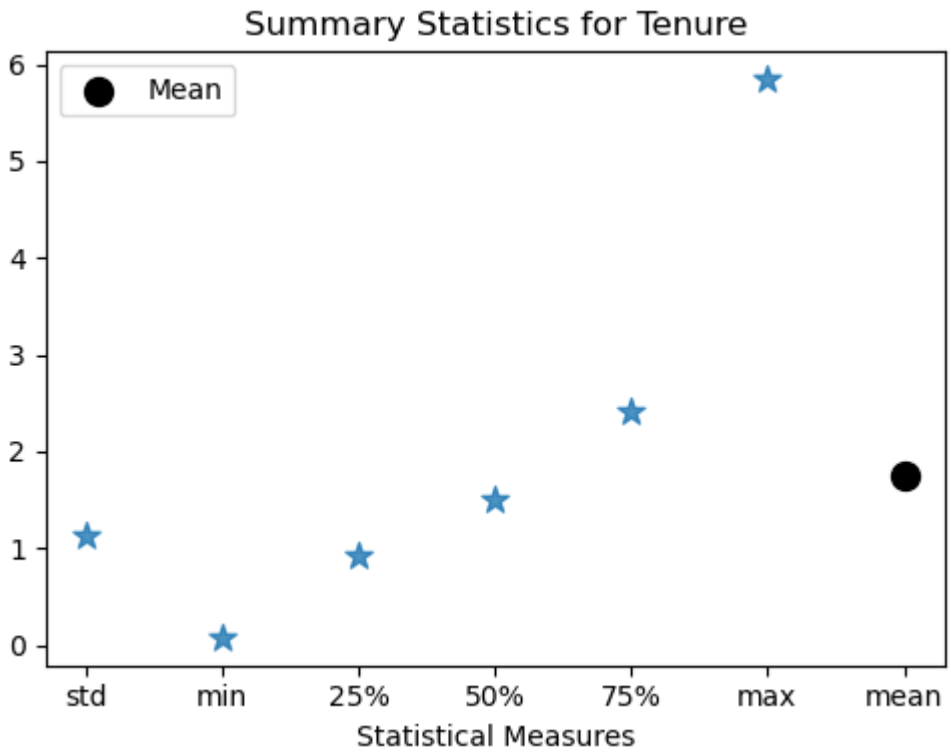
Out[55]:
```
(3864, 33)
```

Tenure

In [56]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['Tenure'])
plt.xlabel('Tenure')
plt.tight_layout()
plt.show()
```

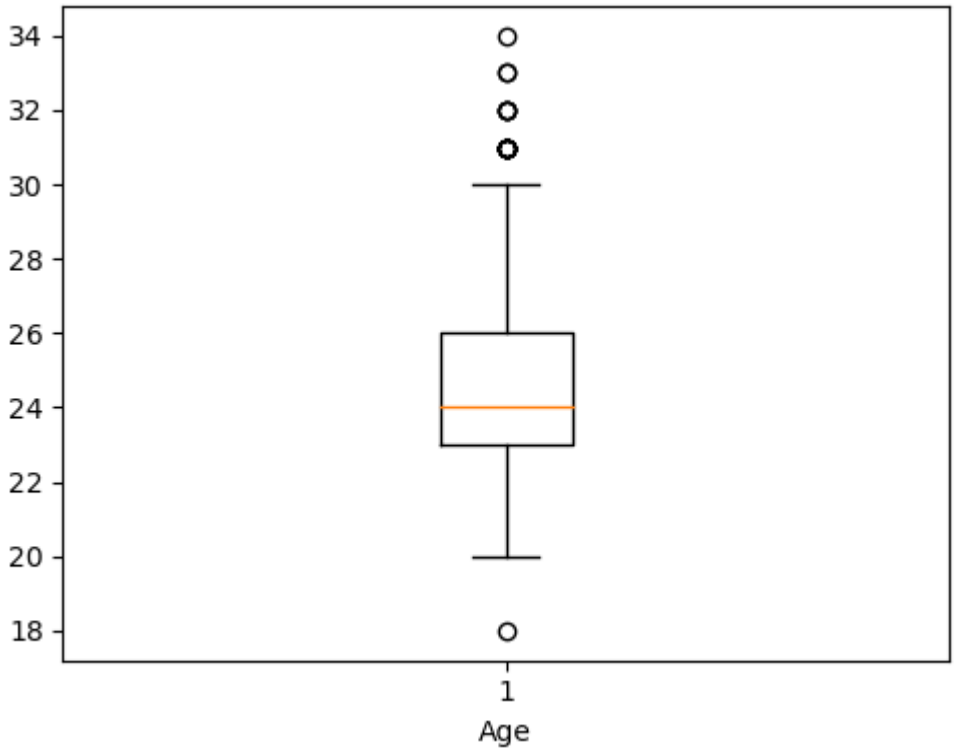


In [57]:
```
plt.figure(figsize=(5,4))
stats = df['Tenure'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='black', label='Mean', s=100)
plt.title('Summary Statistics for Tenure')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```



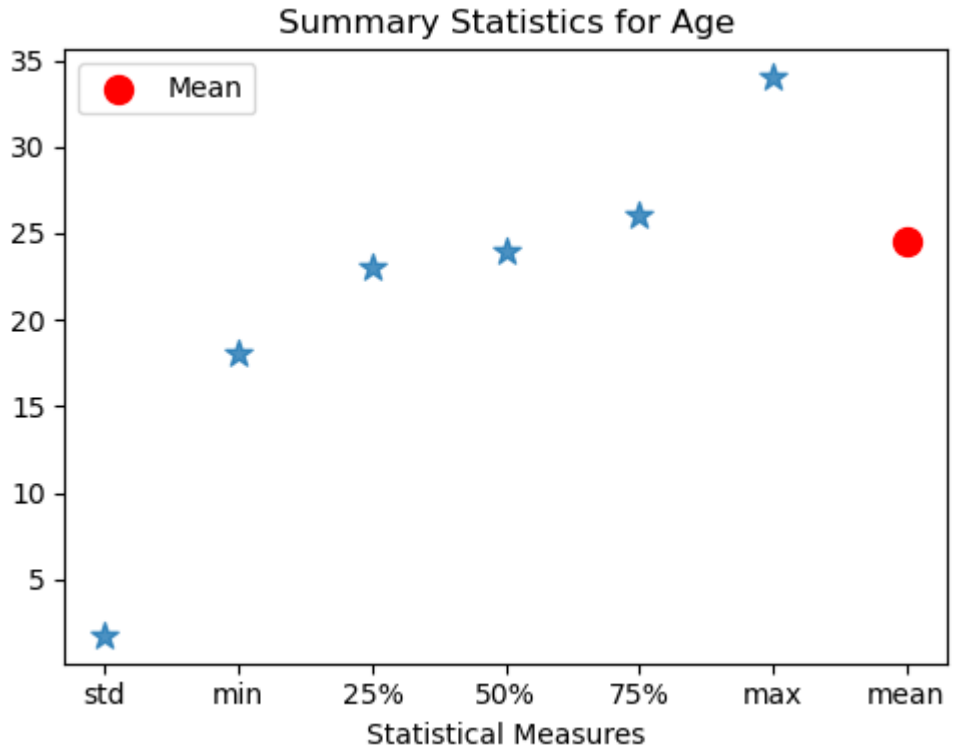
Age

In [58]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['Age'])
plt.xlabel('Age')
plt.tight_layout()
plt.show()
```



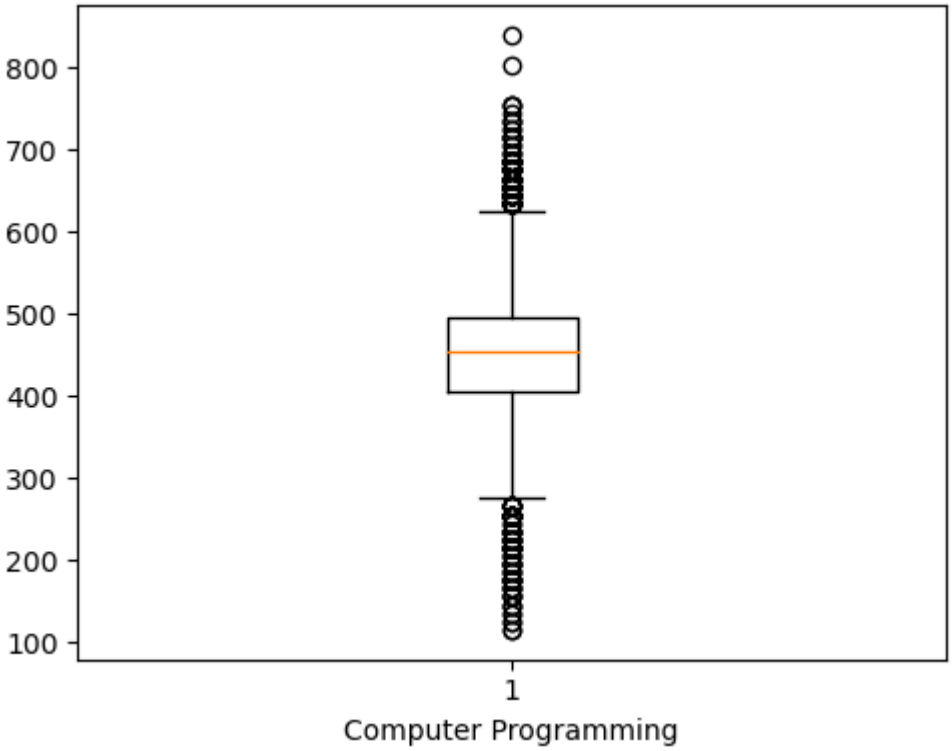
In [59]:
```
plt.figure(figsize=(5,4))
stats = df['Age'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='red', label='Mean', s=100)
```

```
plt.title('Summary Statistics for Age')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```

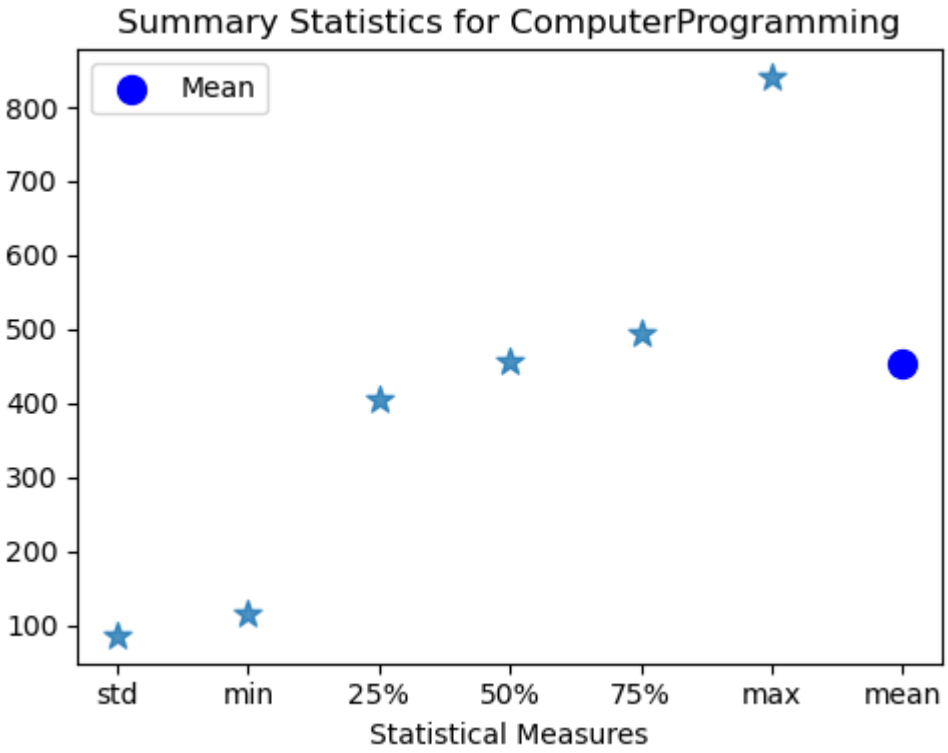


Computer Programming

In [60]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['ComputerProgramming'])
plt.xlabel('Computer Programming')
plt.tight_layout()
plt.show()
```

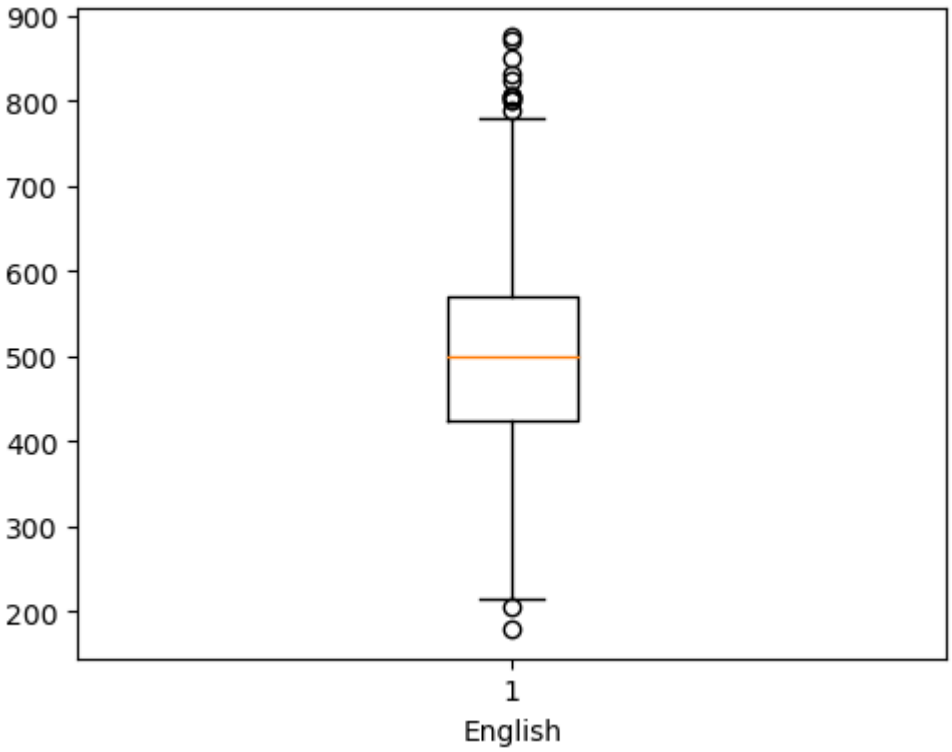


In [61]:
```
plt.figure(figsize=(5,4))
stats = df['ComputerProgramming'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='blue', label='Mean', s=100)
plt.title('Summary Statistics for ComputerProgramming')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```

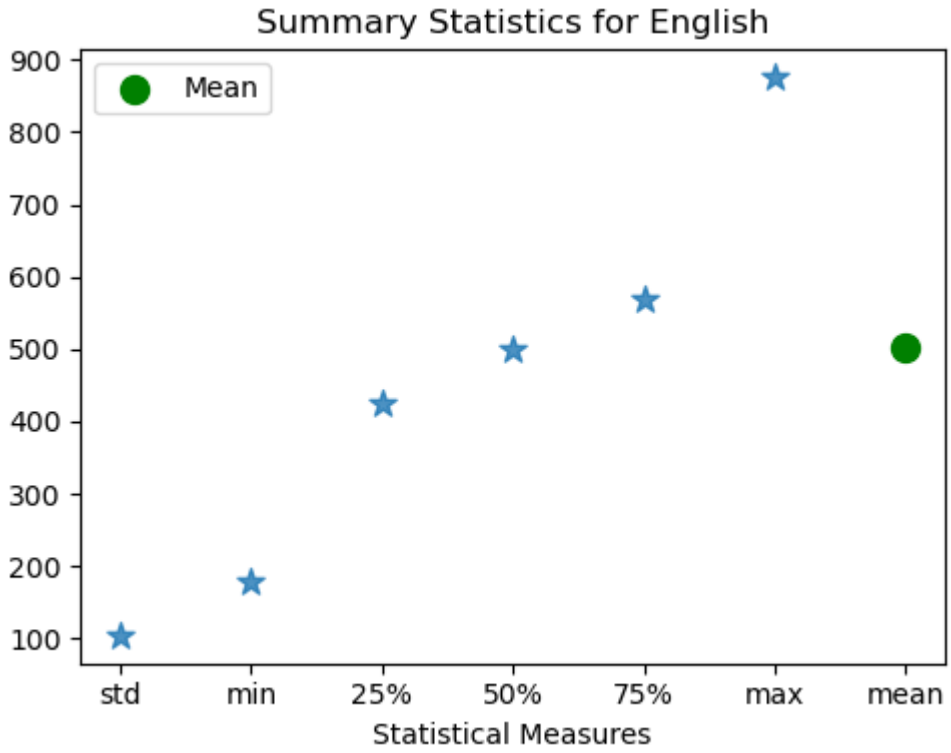


English

In [62]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['English'])
plt.xlabel('English')
plt.tight_layout()
plt.show()
```

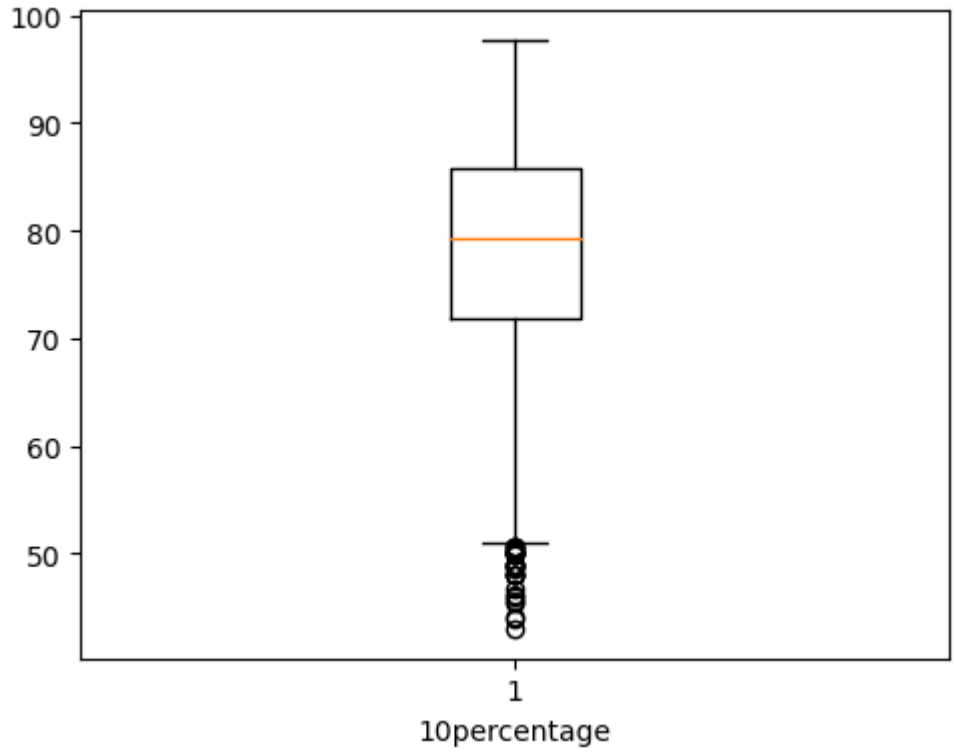


In [63]:
```
plt.figure(figsize=(5,4))
stats = df['English'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='green', label='Mean', s=100)
plt.title('Summary Statistics for English')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```



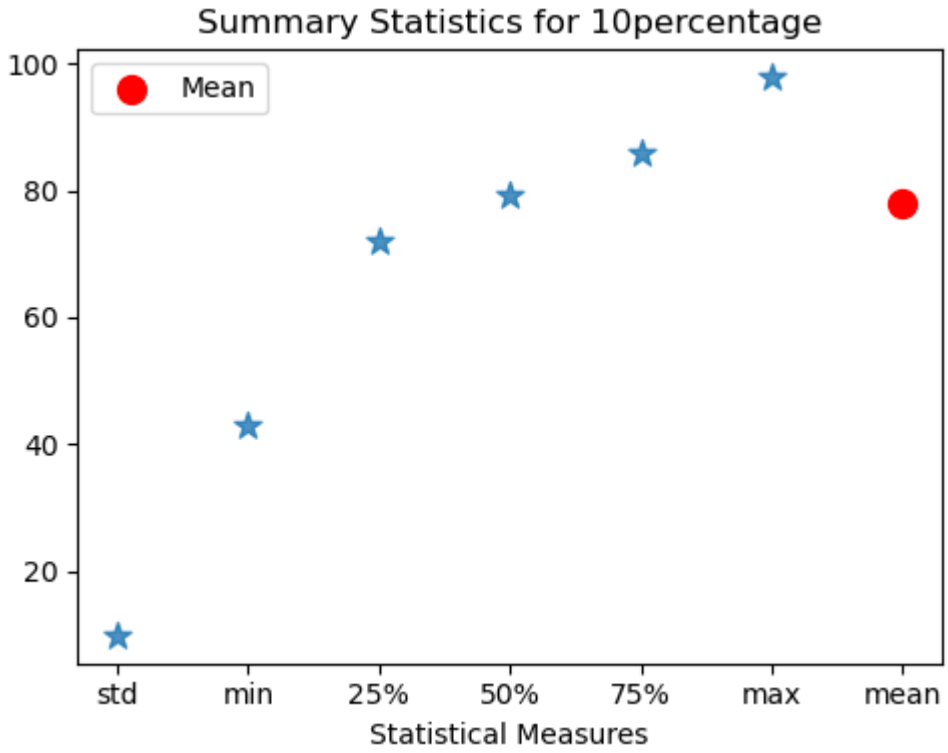
10th Percentage

In [64]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['10percentage'])
plt.xlabel('10percentage')
plt.tight_layout()
plt.show()
```



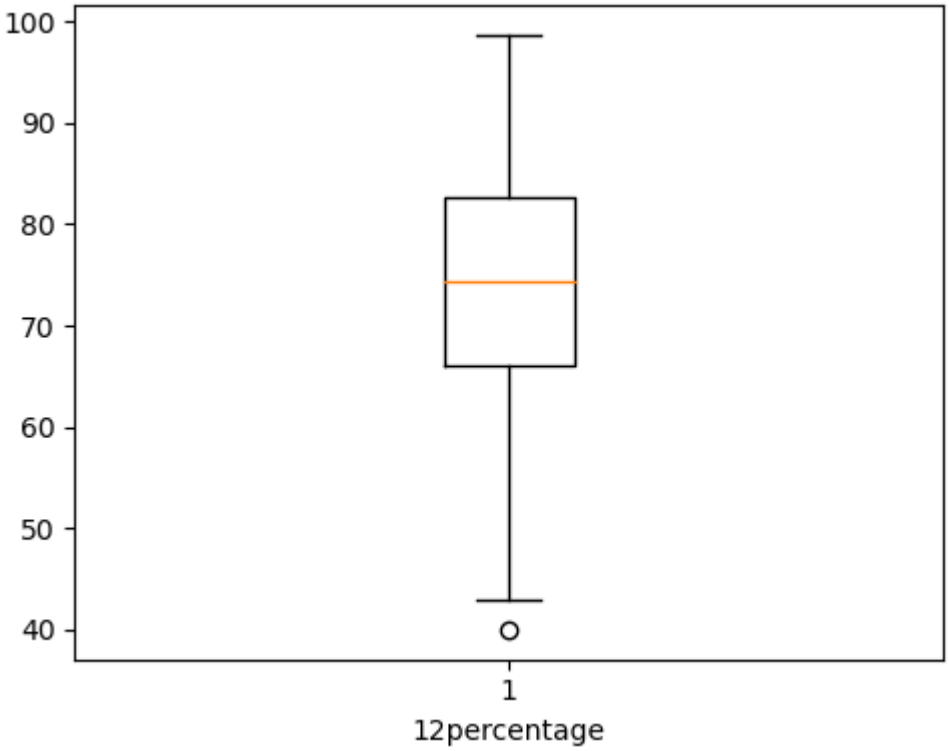
In [65]:
```
plt.figure(figsize=(5,4))
stats = df['10percentage'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='red', label='Mean', s=100)
```

```
plt.title('Summary Statistics for 10percentage')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```

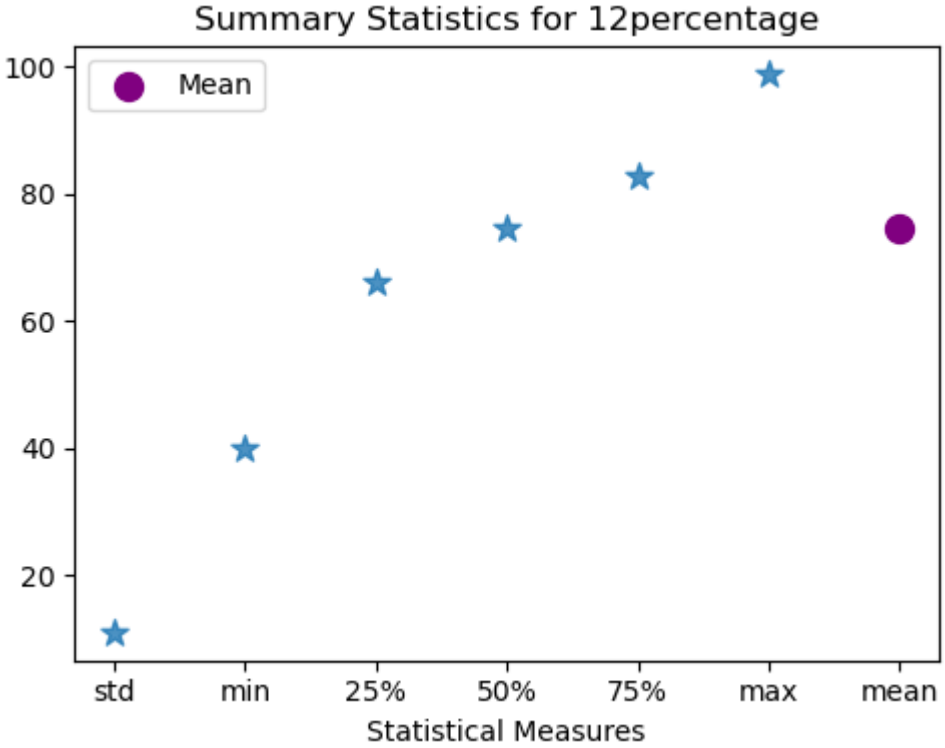


12th Percentage

In [66]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['12percentage'])
plt.xlabel('12percentage')
plt.tight_layout()
plt.show()
```

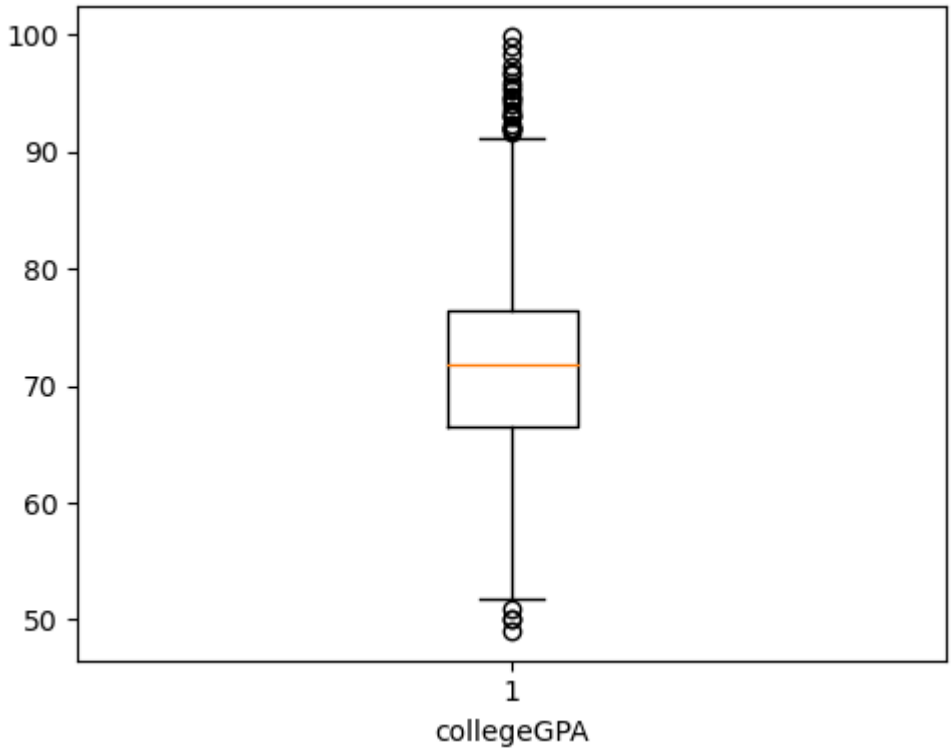


In [67]:
```
plt.figure(figsize=(5,4))
stats = df['12percentage'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='purple', label='Mean', s=100)
plt.title('Summary Statistics for 12percentage')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```

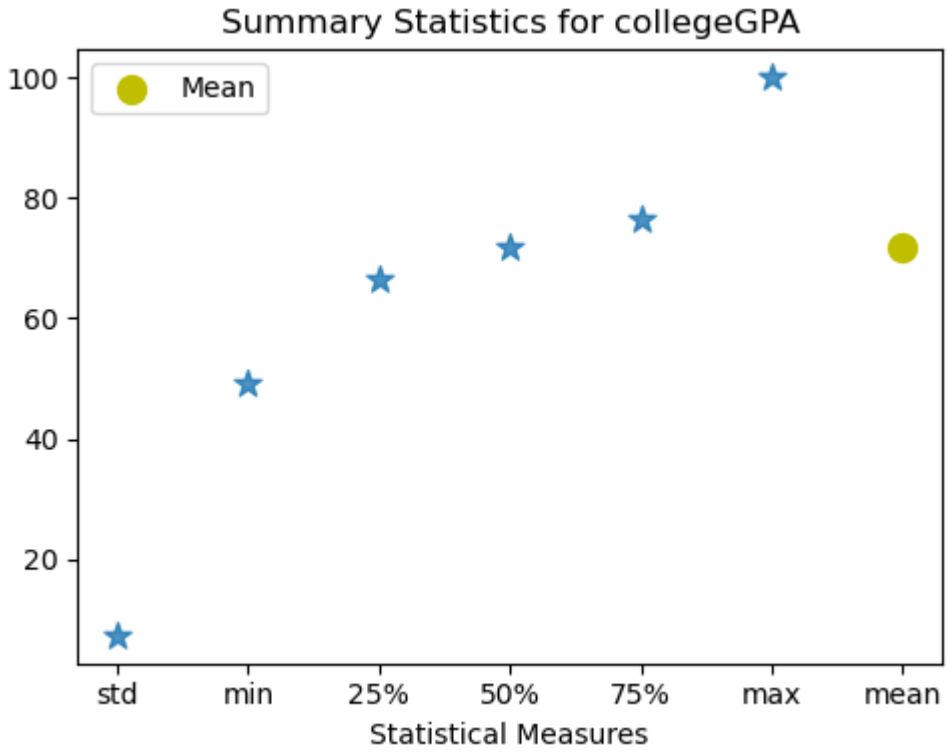


CollegeGPA

In [68]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['collegeGPA'])
plt.xlabel('collegeGPA')
plt.tight_layout()
plt.show()
```

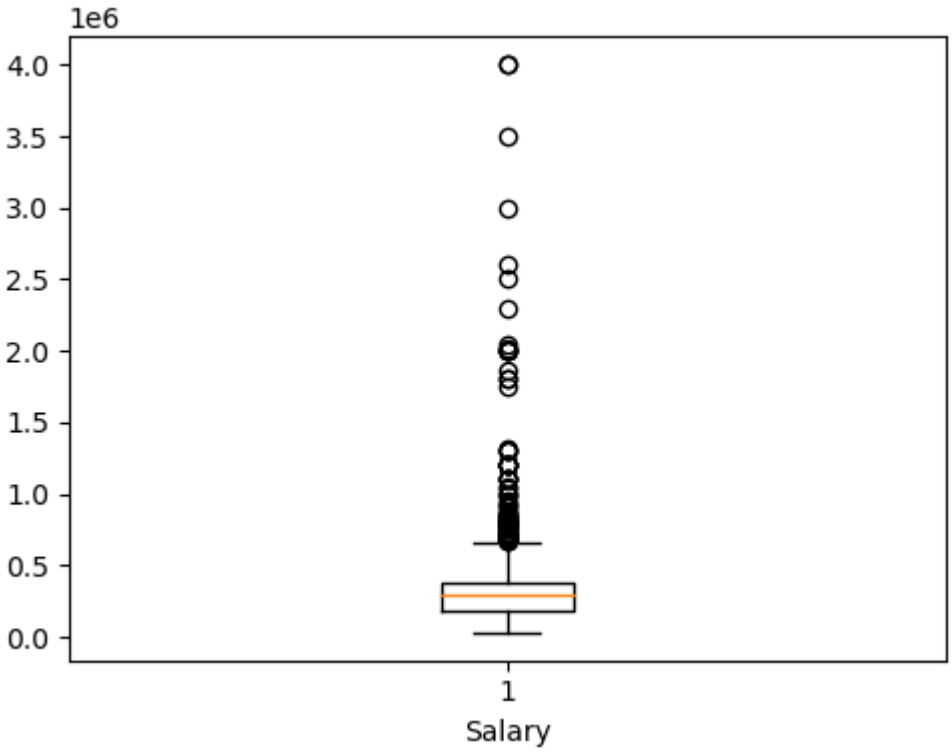


In [69]:
```
plt.figure(figsize=(5,4))
stats = df['collegeGPA'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='y', label='Mean', s=100)
plt.title('Summary Statistics for collegeGPA')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```



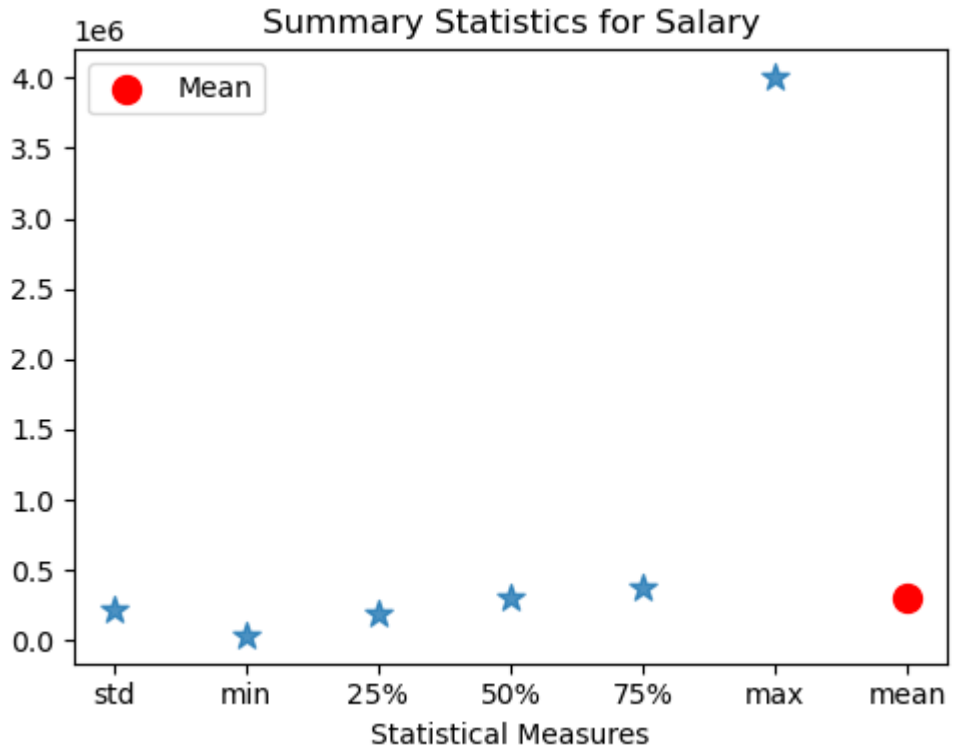
Salary

In [70]:
```
plt.figure(figsize=(5,4))
plt.boxplot(df['Salary'])
plt.xlabel('Salary')
plt.tight_layout()
plt.show()
```



In [71]:
```
plt.figure(figsize=(5,4))
stats = df['Salary'].describe()[1:]
plt.scatter(stats.index[1:], stats.values[1:], marker='*', s=100, alpha=0.8)
plt.scatter('mean', stats['mean'], marker='o', color='r', label='Mean', s=100)
```

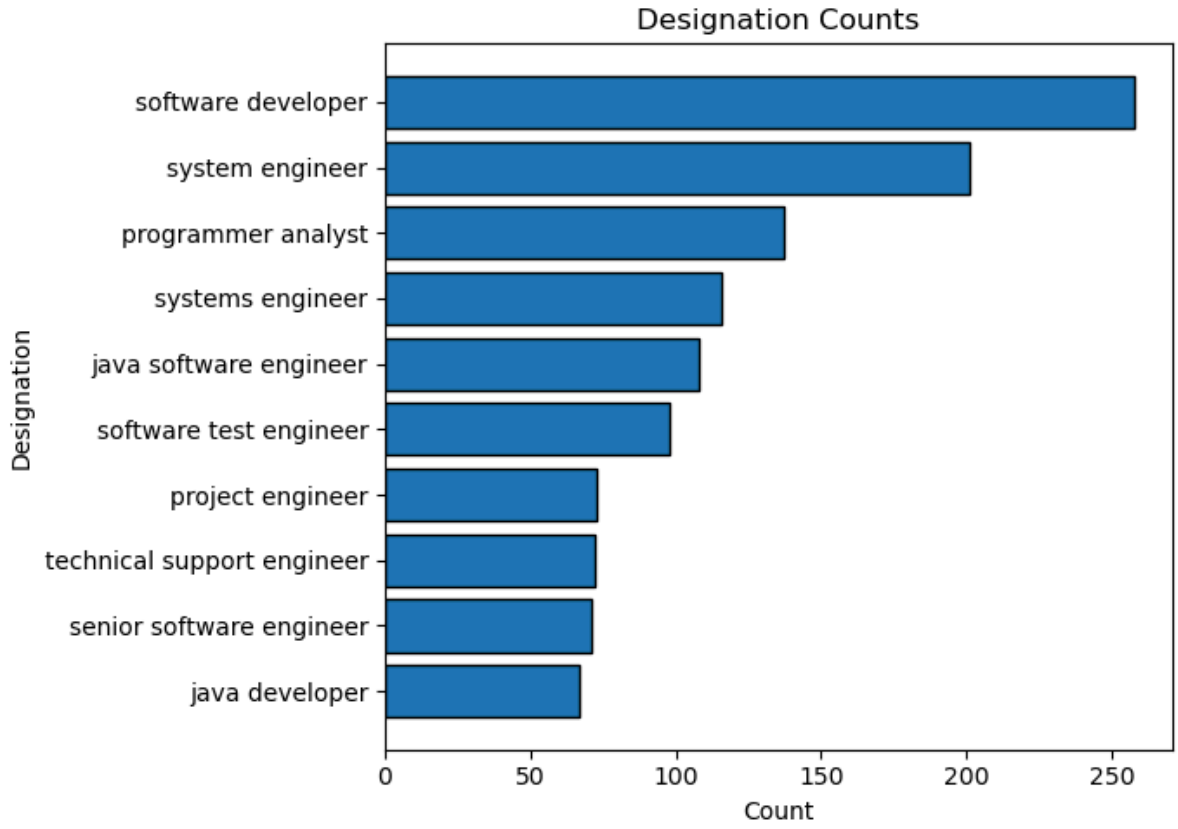
```
plt.title('Summary Statistics for Salary')
plt.xlabel('Statistical Measures')
plt.legend()
plt.tight_layout()
plt.show()
```



Univariate - Visual Analysis(Categorical Features)

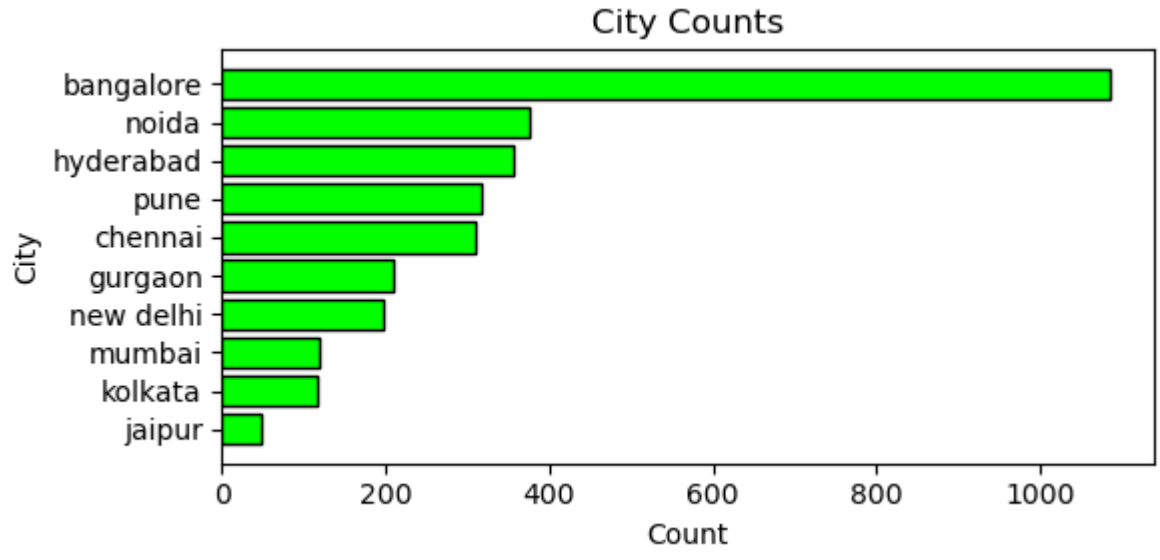
Designation

In [72]:
```
designation_counts = df['Designation'].value_counts()[1:].sort_values(ascending=Tru
top_10_designations = designation_counts.tail(10)
plt.figure(figsize=(7, 5))
plt.barh(top_10_designations.index, top_10_designations.values, edgecolor='k')
plt.title('Designation Counts')
plt.xlabel('Count')
plt.ylabel('Designation')
plt.tight_layout()
plt.show()
```



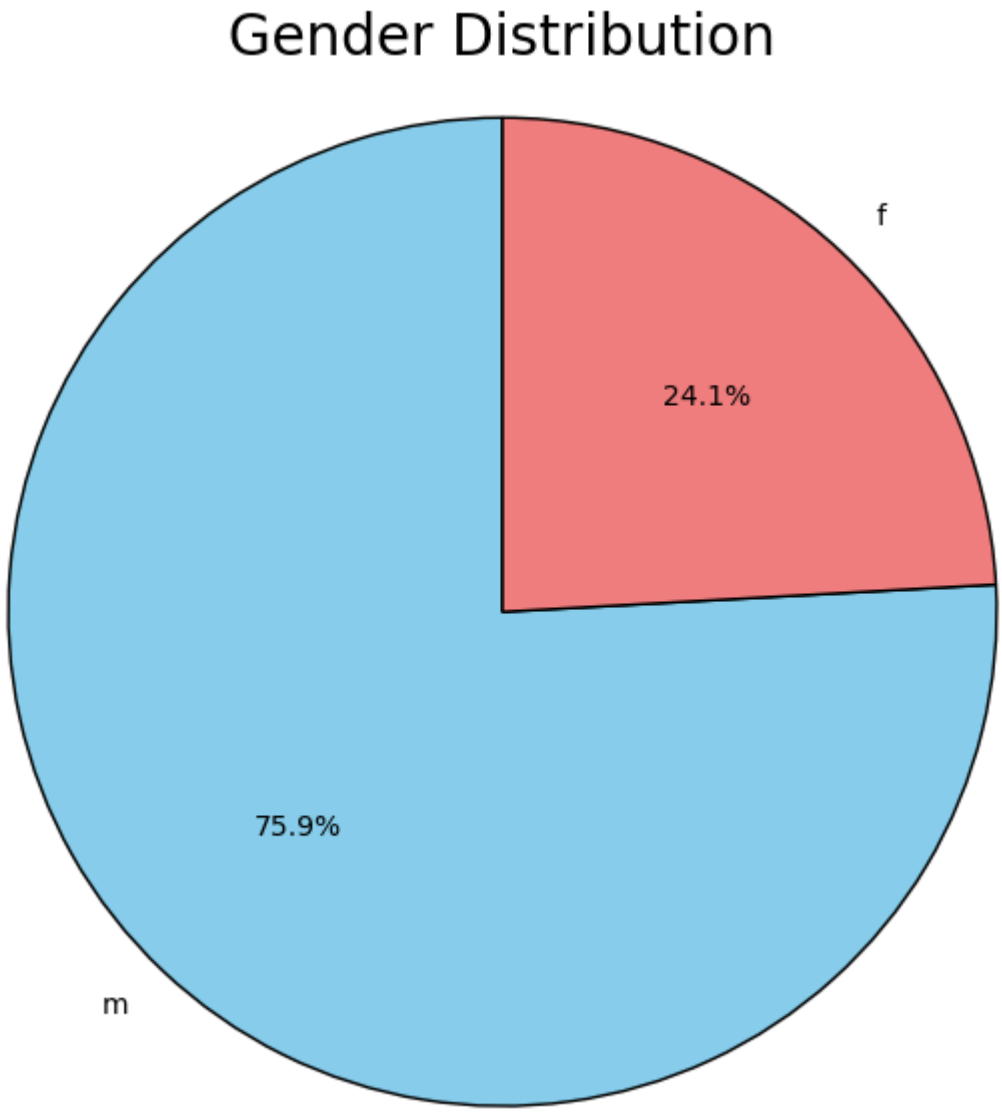
Job City

In [73]:
```
city_counts = df['JobCity'].value_counts().sort_values(ascending=True)
top_10_cities = city_counts.tail(10)

plt.figure(figsize=(6, 3))
plt.barh(top_10_cities.index, top_10_cities.values, color='lime', edgecolor='black'
plt.title('City Counts')
plt.xlabel('Count')
plt.ylabel('City')
plt.tight_layout()
plt.show()
```



Gender

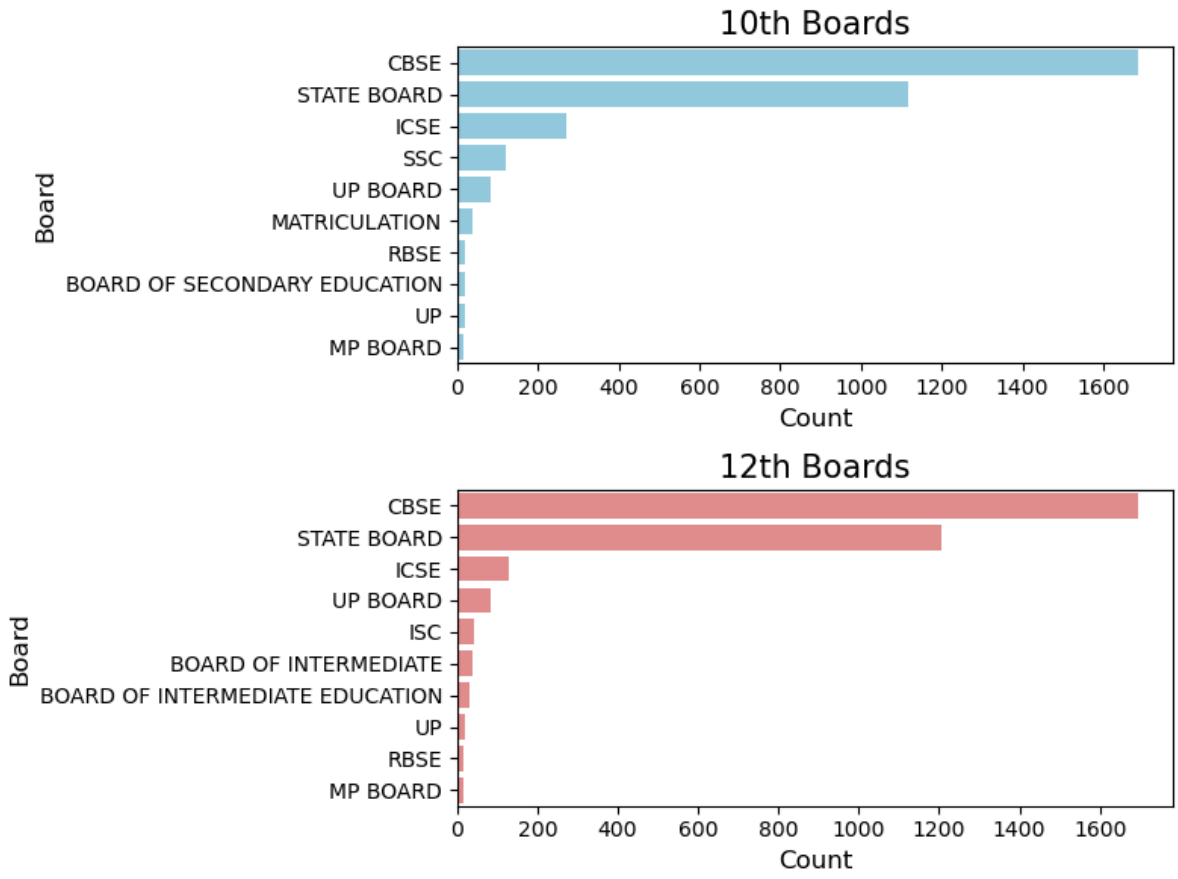
In [74]:
```
gender_counts = df['Gender'].value_counts()
labels = gender_counts.index
sizes = gender_counts.values

plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90, colors=['skyblue',
plt.title('Gender Distribution', fontsize=20)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.tight_layout()
plt.show()
```



10board & 12board

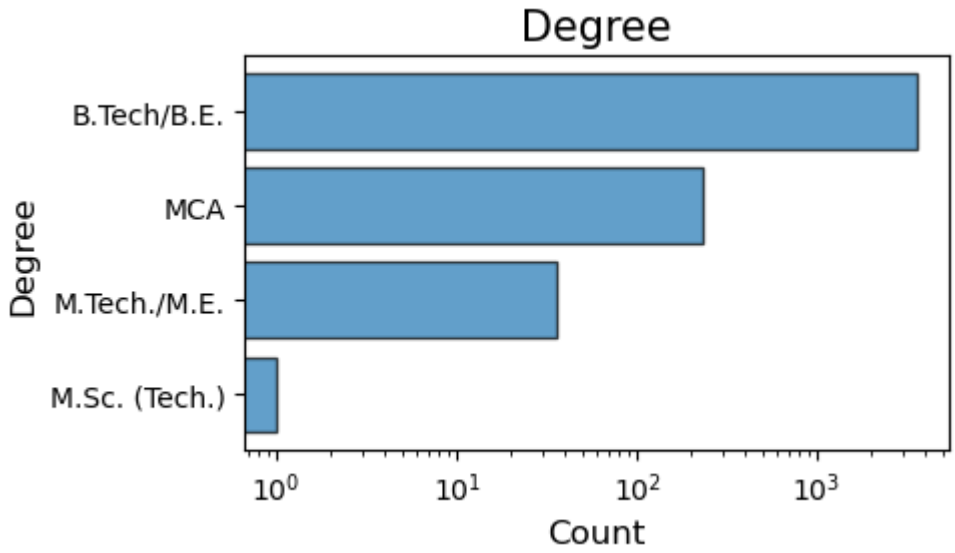
In [75]:
```
top_10_10th = df['10board'].str.upper().value_counts().nlargest(10)
top_10_12th = df['12board'].str.upper().value_counts().nlargest(10)

# Plotting
plt.figure(figsize=(8, 6))
plt.subplot(2, 1, 1)
sns.barplot(y=top_10_10th.index, x=top_10_10th.values, color='skyblue')
plt.title('10th Boards', fontsize=15)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Board', fontsize=12)
```

```
plt.subplot(2, 1, 2)
sns.barplot(y=top_10_12th.index, x=top_10_12th.values, color='lightcoral')
plt.title('12th Boards', fontsize=15)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Board', fontsize=12)

plt.tight_layout()
plt.show()
```

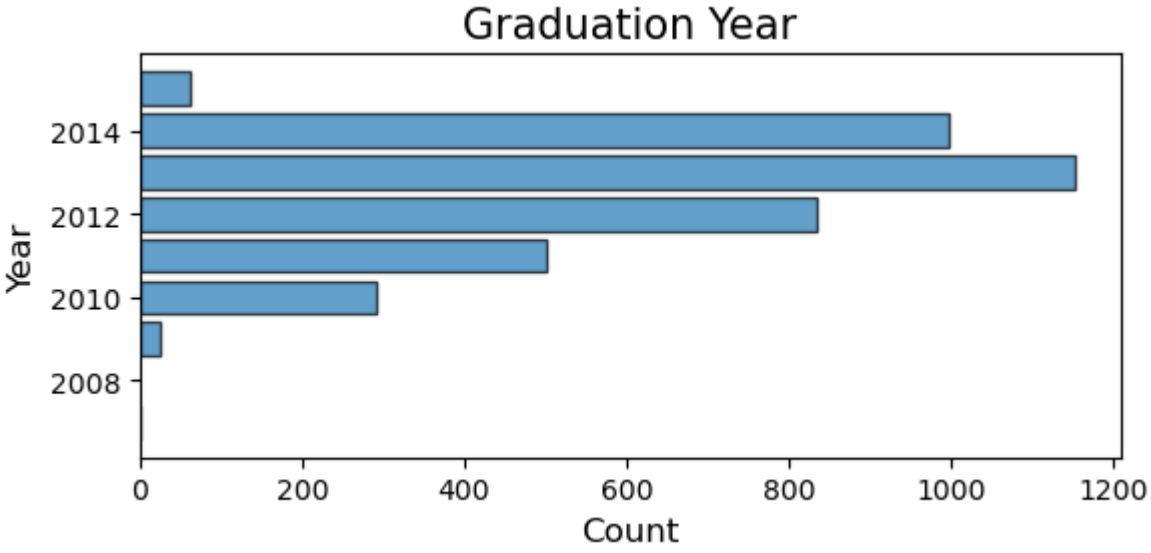


Degree

In [76]:
```
degree_counts = df['Degree'].value_counts().sort_values(ascending=True)

# Plotting
plt.figure(figsize=(5, 3))
plt.barh(degree_counts.index, degree_counts.values, edgecolor='k', alpha=0.7)
plt.title('Degree', fontsize=15)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Degree', fontsize=12)
plt.xscale('log')
plt.tight_layout()
plt.show()
```



In [77]:
```
graduation_year_counts = df['GraduationYear'].value_counts().sort_values(ascending=

# Plotting
plt.figure(figsize=(6, 3))
plt.barh(graduation_year_counts.index, graduation_year_counts.values, edgecolor='k'
plt.title('Graduation Year', fontsize=15)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Year', fontsize=12)
plt.tight_layout()
plt.show()
```



Removing Outliers

In [78]:
```
def outlier_treatment(datacolumn):

    Q1 = datacolumn.quantile(0.25)
    Q3 = datacolumn.quantile(0.75)


    IQR = Q3 - Q1


    lower_range = Q1 - (1.5 * IQR)
    upper_range = Q3 + (1.5 * IQR)
```

```
    return lower_range, upper_range
```

In [79]:
```
df.columns
```

Out[79]:
```
Index(['Salary', 'DOJ', 'DOL', 'Designation', 'JobCity', 'Gender', 'DOB',
       '10percentage', '10board', '12graduation', '12percentage', '12board',
       'CollegeTier', 'Degree', 'Specialization', 'collegeGPA',
       'CollegeCityTier', 'CollegeState', 'GraduationYear', 'English',
       'Logical', 'Quant', 'Domain', 'ComputerProgramming',
       'ElectronicsAndSemicon', 'ComputerScience', 'conscientiousness',
       'agreeableness', 'extraversion', 'nueroticism', 'openess_to_experience',
       'Age', 'Tenure'],
      dtype='object')
```

In [80]:
```
columns = ['Salary','10percentage','12percentage','English','Logical',
           'Quant','Domain', 'ComputerProgramming','ElectronicsAndSemicon',
           'ComputerScience', 'conscientiousness','agreeableness',
           'extraversion', 'nueroticism', 'openess_to_experience','Age',
           'Tenure']
df1 = df.copy()
```

In [81]:
```
for cols in columns:
    lowerbound, upperbound = outlier_treatment(df1[cols])
    df1 = df1.drop(df1[(df1[cols] < lowerbound) | (df1[cols] > upperbound)].
                   index)
```

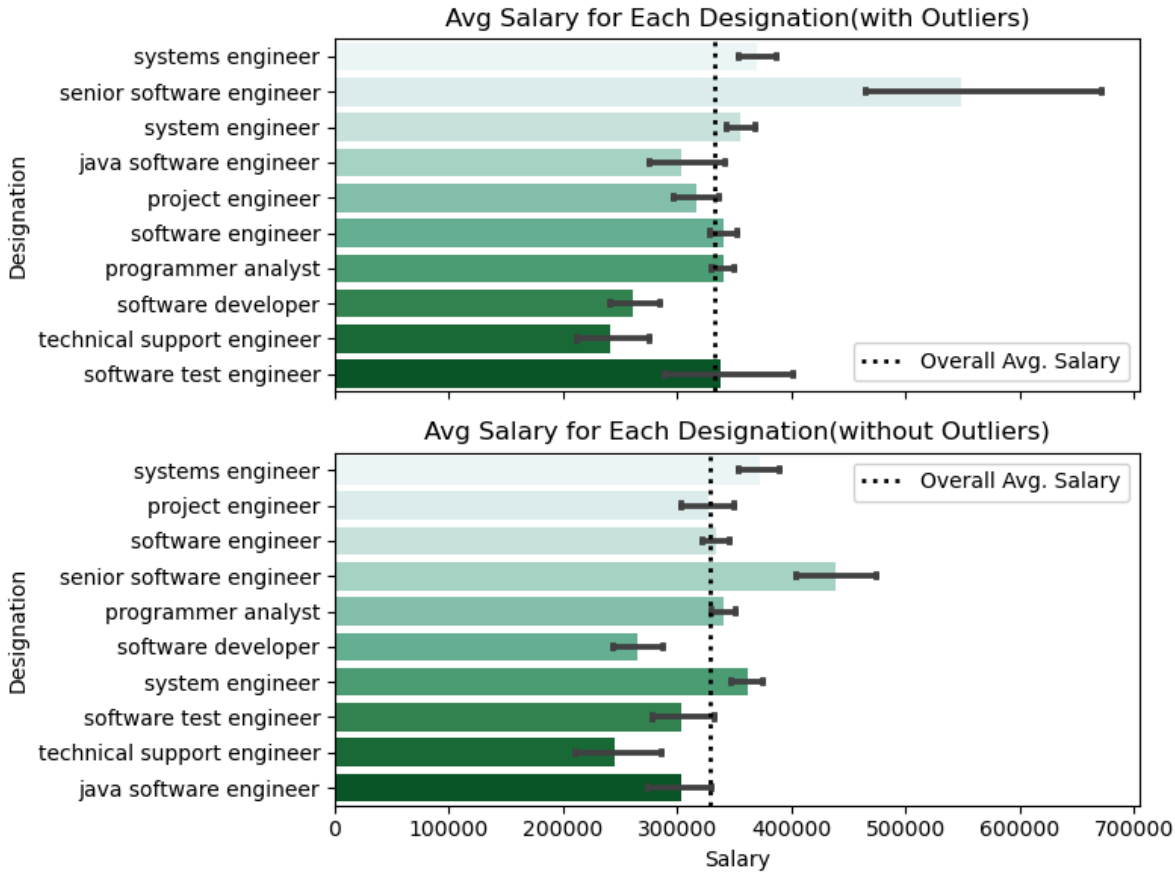
In [82]:
```
print(f'Number of observation with outliers: {df.shape[0]}')
print(f'Number of observations without outliers: {df1.shape[0]}')
```

```
Number of observation with outliers: 3864
Number of observations without outliers: 2490
```

Bivariate Analysis

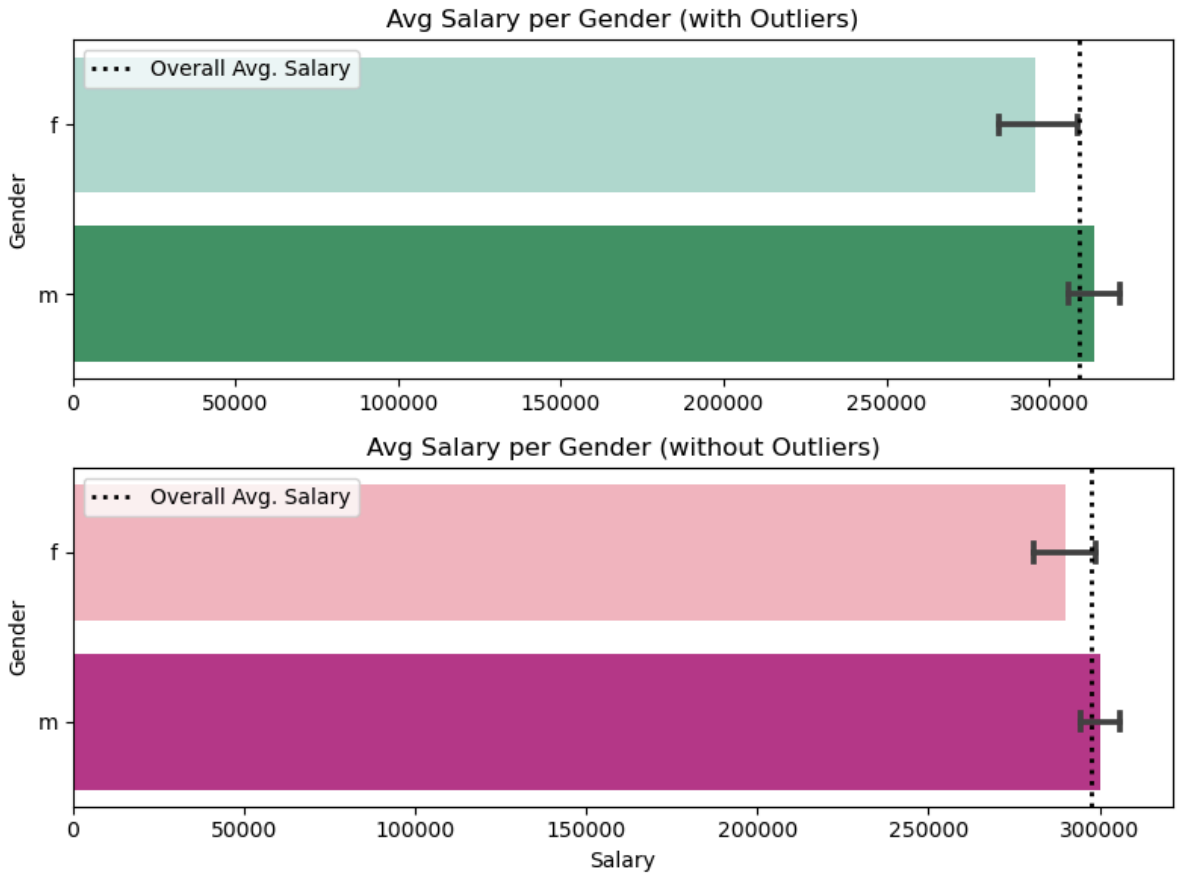
In [83]:
```
# Get top 10 designations with outliers
top_10_designations_with_outliers = df['Designation'].value_counts().nlargest(10).i

# Filter dataframe for top 10 designations with outliers
df_top_10_with_outliers = df[df['Designation'].isin(top_10_designations_with_outlie

# Get top 10 designations without outliers
top_10_designations_without_outliers = df1['Designation'].value_counts().nlargest(1

# Filter dataframe for top 10 designations without outliers
df_top_10_without_outliers = df1[df1['Designation'].isin(top_10_designations_withou

# Plotting
fig, ax = plt.subplots(2, 1, figsize=(8, 6), sharex=True)

# Plotting with outliers
sns.barplot(x='Salary', y='Designation', data=df_top_10_with_outliers, palette='Bu
ax[0].axvline(df_top_10_with_outliers['Salary'].mean(), color='k', linestyle=':', l
ax[0].set_title('Avg Salary for Each Designation(with Outliers)')
ax[0].legend()
ax[0].set_xlabel('')

# Plotting without outliers
sns.barplot(x='Salary', y='Designation', data=df_top_10_without_outliers, palette='
ax[1].axvline(df_top_10_without_outliers['Salary'].mean(), color='k', linestyle=':'
ax[1].set_title('Avg Salary for Each Designation(without Outliers)')
```

```
ax[1].legend()
ax[1].set_xlabel('Salary')

plt.tight_layout()
plt.show()
```

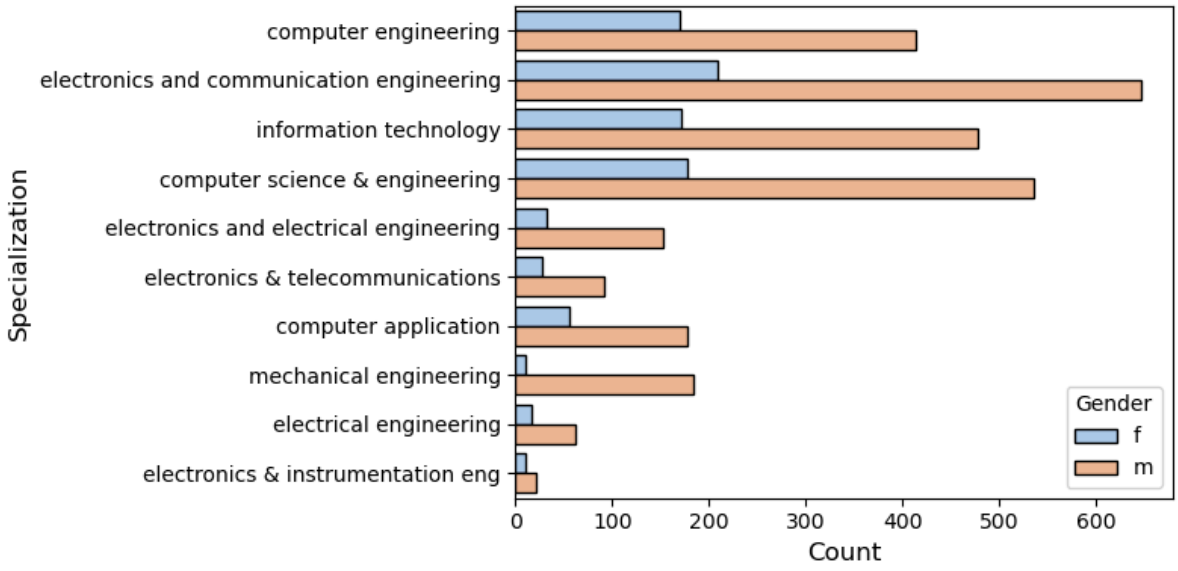


In [84]:

```
# Plotting with outliers
plt.figure(figsize=(8, 6))
plt.subplot(2, 1, 1)
sns.barplot(x='Salary', y='Gender', data=df, palette='BuGn', capsize=0.1)
plt.axvline(df['Salary'].mean(), color='k', linestyle=':', linewidth=2, label='Over
plt.title('Avg Salary per Gender (with Outliers)')
plt.legend()
plt.xlabel('')

# Plotting without outliers
plt.subplot(2, 1, 2)
sns.barplot(x='Salary', y='Gender', data=df1, palette='RdPu', capsize=0.1)
plt.axvline(df1['Salary'].mean(), color='k', linestyle=':', linewidth=2, label='Ove
plt.title('Avg Salary per Gender (without Outliers)')
plt.legend()
plt.xlabel('Salary')

plt.tight_layout()
plt.show()
```



In [85]:
```
# Get top 10 specializations
top_10_specializations = df['Specialization'].value_counts().nlargest(10).index

# Filter dataframe for top 10 specializations
df_top_10_specializations = df[df['Specialization'].isin(top_10_specializations)]

# Plotting
plt.figure(figsize=(8, 4))
sns.countplot(y='Specialization', hue='Gender', data=df_top_10_specializations, pal
plt.xlabel('Count', fontsize=12)
plt.ylabel('Specialization', fontsize=12)
plt.legend(title='Gender', fontsize=10)
plt.tight_layout()
plt.show()
```



Research Question

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.

In [86]:
```
df['Designation'] = df['Designation'].replace(['programmer analyst trainee',
                                               'programmer analyst'],
                                              'programmer analyst')
df['Designation'] = df['Designation'].replace(['software eng', 'software engg',
                                               'software engineer',
                                               'software engineere',
                                               'software enginner'],
                                              'software engineer')
```

In [87]:
```
df2 = df[(df["Designation"].isin(["programmer analyst",
                                  "software engineer",
                                  "hardware engineer",
                                  "associate engineer"]))&(df["Specia
```

In [88]:
```
job_group = df2.groupby('Designation')
job_salary_mean = job_group['Salary'].mean()
job_salary_std = job_group['Salary'].std()
```

In [89]:
```
print("Mean salaries for different job roles:")
print(job_salary_mean)
print("\nStandard deviation of salaries for different job roles:")
print(job_salary_std)
```

```
Mean salaries for different job roles:
Designation
associate engineer    281666.666667
programmer analyst    345267.857143
Name: Salary, dtype: float64

Standard deviation of salaries for different job roles:
Designation
associate engineer    89768.220063
programmer analyst    55844.098271
Name: Salary, dtype: float64
```

In [ ]: