a-project-analysis-of-amcat-data

May 9, 2024

1 EDA Project - Analysis of AMCAT Data

1.1 Columns Summary Table for dataset

- ID: Candidate ID
- Salary: Salary of the candidate
- DOJ: Date of joining the job
- DOL: Date of leaving the job
- Designation: Job designation/title
- JobCity: City where the job is located
- Gender: Gender of the candidate
- DOB: Date of birth of the candidate
- 10percentage: Percentage score in 10th grade
- 12percentage: Percentage score in 12th grade
- CollegeID: College ID of the candidate
- CollegeTier: Tier of the college
- Degree: Degree pursued by the candidate
- Specialization: Specialization pursued by the candidate
- CollegeGPA: Grade Point Average in college
- CollegeCityID: ID of the college city
- CollegeCityTier: Tier of the college city
- CollegeState: State where the college is located
- GraduationYear: Year of graduation
- Domain: Domain knowledge score
- ComputerProgramming: Score in computer programming
- ElectronicsAndSemicon: Score in electronics and semiconductors
- ComputerScience: Score in computer science
- MechanicalEngg: Score in mechanical engineering
- ElectricalEngg: Score in electrical engineering
- TelecomEngg: Score in telecommunications engineering
- CivilEngg: Score in civil engineering
- Conscientiousness, Agreeableness, Extraversion, Neuroticism, Openness_to_experience: Personality trait scores

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

```
import warnings
from scipy.stats import chi2_contingency
```

[2]:
```
df = pd.read_csv(r"C:\Users\Manikanta\Data Science Innomatics\Internship␣
↪Projects - Tasks\Exploratory Data Analysis(EDA)\data.xlsx - Sheet1.csv")
```

[3]:
```
df.head()
```

[3]:
```
  Unnamed: 0      ID     Salary          DOJ          DOL  \
0      train  203097   420000.0  6/1/12 0:00      present
1      train  579905   500000.0  9/1/13 0:00      present
2      train  810601   325000.0  6/1/14 0:00      present
3      train  267447  1100000.0  7/1/11 0:00      present
4      train  343523   200000.0  3/1/14 0:00  3/1/15 0:00

                Designation    JobCity Gender           DOB  10percentage  \
0   senior quality engineer  Bangalore      f  2/19/90 0:00          84.3
1         assistant manager     Indore      m  10/4/89 0:00          85.4
2          systems engineer    Chennai      f   8/3/92 0:00          85.0
3  senior software engineer    Gurgaon      m  12/5/89 0:00          85.6
4                       get    Manesar      m  2/27/91 0:00          78.0

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

   conscientiousness agreeableness extraversion  nueroticism  \
0             0.9737        0.8128       0.5269      1.35490
1            -0.7335        0.3789       1.2396     -0.10760
2             0.2718        1.7109       0.1637     -0.86820
3             0.0464        0.3448      -0.3440     -0.40780
4            -0.8810       -0.2793      -1.0697      0.09163

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

[5 rows x 39 columns]
```

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

[4]: (3998, 39)

[5]:
```
df.columns
```

[5]:
```
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')
```

[6]:
```
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
```

[7]:
```
# Remove any duplicate rows
df.drop_duplicates(inplace=True)
```

[8]:
```
# Check for missing values
missing_values = df.isnull().sum()
missing_values
```

[8]:
```
Unnamed: 0             0
ID                     0
Salary                 0
DOJ                    0
DOL                    0
Designation            0
JobCity                0
Gender                 0
DOB                    0
10percentage           0
10board                0
12graduation           0
12percentage           0
12board                0
CollegeID              0
CollegeTier            0
Degree                 0
Specialization         0
collegeGPA             0
CollegeCityID          0
CollegeCityTier        0
CollegeState           0
GraduationYear         0
```

```
English                   0
Logical                   0
Quant                     0
Domain                    0
ComputerProgramming       0
ElectronicsAndSemicon     0
ComputerScience           0
MechanicalEngg            0
ElectricalEngg            0
TelecomEngg               0
CivilEngg                 0
conscientiousness         0
agreeableness             0
extraversion              0
nueroticism               0
openess_to_experience     0
dtype: int64
```

[9]:
```
# Assuming your DataFrame is named df
df.drop(columns=['Unnamed: 0'], inplace=True)
```

[10]:
```
df.describe()
```

[10]:

	ID	Salary	10percentage	12graduation	12percentage	\
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	

	CollegeID	CollegeTier	collegeGPA	CollegeCityID	CollegeCityTier	\
count	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	
mean	5156.851426	1.925713	71.486171	5156.851426	0.300400	
std	4802.261482	0.262270	8.167338	4802.261482	0.458489	
min	2.000000	1.000000	6.450000	2.000000	0.000000	
25%	494.000000	2.000000	66.407500	494.000000	0.000000	
50%	3879.000000	2.000000	71.720000	3879.000000	0.000000	
75%	8818.000000	2.000000	76.327500	8818.000000	1.000000	
max	18409.000000	2.000000	99.930000	18409.000000	1.000000	

	…	ComputerScience	MechanicalEngg	ElectricalEngg	TelecomEngg	\
count	…	3998.000000	3998.000000	3998.000000	3998.000000	
mean	…	90.742371	22.974737	16.478739	31.851176	
std	…	175.273083	98.123311	87.585634	104.852845	

```
min    …        -1.000000       -1.000000       -1.000000    -1.000000
25%    …        -1.000000       -1.000000       -1.000000    -1.000000
50%    …        -1.000000       -1.000000       -1.000000    -1.000000
75%    …        -1.000000       -1.000000       -1.000000    -1.000000
max    …       715.000000      623.000000      676.000000   548.000000

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

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

[8 rows x 27 columns]
```

[ ]:

[11]:
```
df
```

[11]:
```
          ID     Salary           DOJ           DOL  \
0     203097   420000.0   6/1/12 0:00       present
1     579905   500000.0   9/1/13 0:00       present
2     810601   325000.0   6/1/14 0:00       present
3     267447  1100000.0   7/1/11 0:00       present
4     343523   200000.0   3/1/14 0:00   3/1/15 0:00
…      …        …            …            …
3993   47916   280000.0  10/1/11 0:00  10/1/12 0:00
3994  752781   100000.0   7/1/13 0:00   7/1/13 0:00
3995  355888   320000.0   7/1/13 0:00       present
3996  947111   200000.0   7/1/14 0:00   1/1/15 0:00
3997  324966   400000.0   2/1/13 0:00       present

                     Designation           JobCity Gender           DOB  \
0         senior quality engineer         Bangalore      f  2/19/90 0:00
```

```
1               assistant manager            Indore      m  10/4/89 0:00
2                systems engineer           Chennai      f   8/3/92 0:00
3        senior software engineer           Gurgaon      m  12/5/89 0:00
4                             get           Manesar      m  2/27/91 0:00
…                             …              …    …          …
3993            software engineer        New Delhi       m  4/15/87 0:00
3994             technical writer         Hyderabad      f  8/27/92 0:00
3995  associate software engineer         Bangalore      m   7/3/91 0:00
3996           software developer  Asifabadbanglore      f  3/20/92 0:00
3997      senior systems engineer           Chennai      f  2/26/91 0:00

      10percentage                          10board  …  ComputerScience  \
0            84.30  board ofsecondary education,ap  …               -1
1            85.40                            cbse  …               -1
2            85.00                            cbse  …               -1
3            85.60                            cbse  …               -1
4            78.00                            cbse  …               -1
…            …                                …  …             …
3993         52.09                            cbse  …               -1
3994         90.00                     state board  …               -1
3995         81.86                      bse,odisha  …               -1
3996         78.72                     state board  …              438
3997         70.60                            cbse  …               -1

      MechanicalEngg ElectricalEngg  TelecomEngg  CivilEngg conscientiousness  \
0                 -1             -1           -1         -1            0.9737
1                 -1             -1           -1         -1           -0.7335
2                 -1             -1           -1         -1            0.2718
3                 -1             -1           -1         -1            0.0464
4                 -1             -1           -1         -1           -0.8810
…               …             …          …         …               …
3993              -1             -1           -1         -1           -0.1082
3994              -1             -1           -1         -1           -0.3027
3995              -1             -1           -1         -1           -1.5765
3996              -1             -1           -1         -1           -0.1590
3997              -1             -1           -1         -1           -1.1128

     agreeableness  extraversion  nueroticism  openess_to_experience
0           0.8128        0.5269      1.35490                -0.4455
1           0.3789        1.2396     -0.10760                 0.8637
2           1.7109        0.1637     -0.86820                 0.6721
3           0.3448       -0.3440     -0.40780                -0.9194
4          -0.2793       -1.0697      0.09163                -0.1295
…           …           …          …                    …
3993        0.3448        0.2366      0.64980                -0.9194
3994        0.8784        0.9322      0.77980                -0.0943
3995       -1.5273       -1.5051     -1.31840                -0.7615
```

```
3996        0.0459       -0.4511      -0.36120                -0.0943
3997       -0.2793       -0.6343       1.32553                -0.6035

[3998 rows x 38 columns]
```

[ ]:

1.2 Univariate Analysis

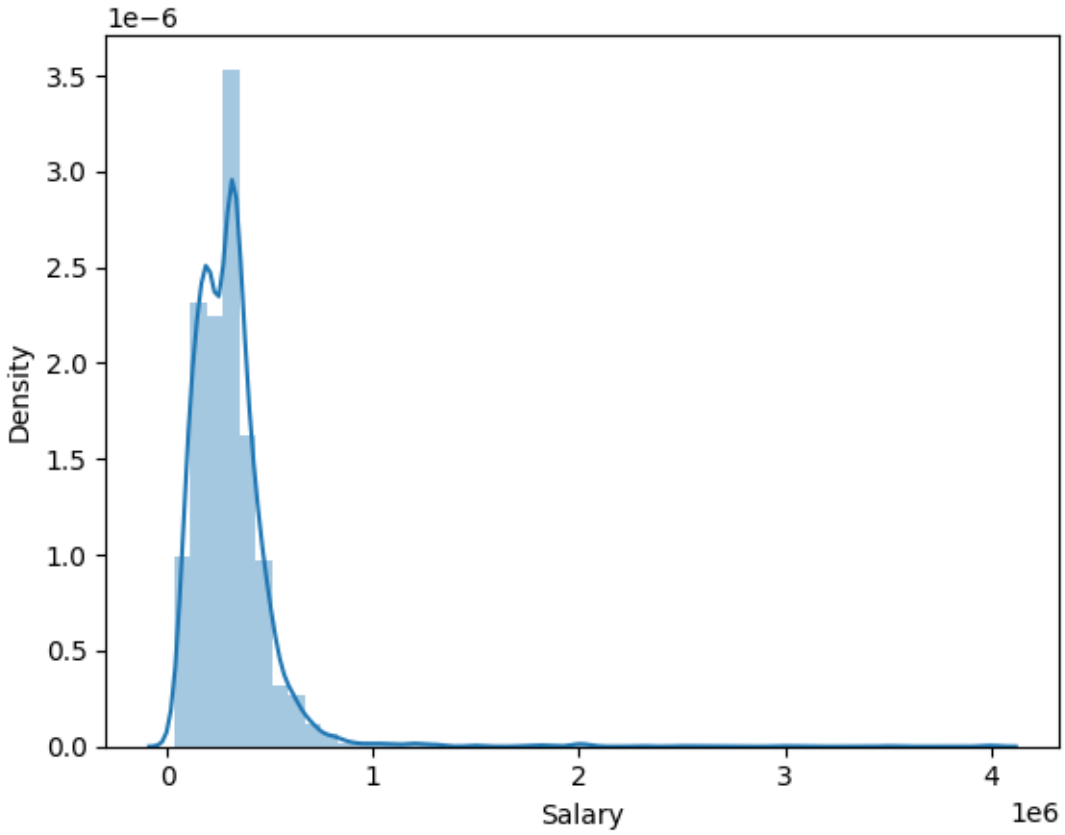
[12]:
```
df['Salary']
```

[12]:
```
0        420000.0
1        500000.0
2        325000.0
3       1100000.0
4        200000.0
           …
3993     280000.0
3994     100000.0
3995     320000.0
3996     200000.0
3997     400000.0
Name: Salary, Length: 3998, dtype: float64
```

[13]:
```
sns.distplot(df['Salary'])
```

```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\3249851952.py:1:
UserWarning:

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

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

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

  sns.distplot(df['Salary'])
```

[13]:
```
<Axes: xlabel='Salary', ylabel='Density'>
```



- From above graph we can observe there is a outlier, Salary >10,00,000 is very rare, especially in the first job. So these are considered as outliers and removed.

[14]:
```
# Initialize a dictionary to store the count of rows for each salary threshold
salary_counts = {250000 * i: (df['Salary'] <= 250000 * i).sum() for i in␣
 ↪range(1, 8)}

# Print the counts for each threshold
for threshold, count in salary_counts.items():
    print(f"Number of Rows in dataframe in which Salary <= {threshold}:␣
 ↪{count}")
```

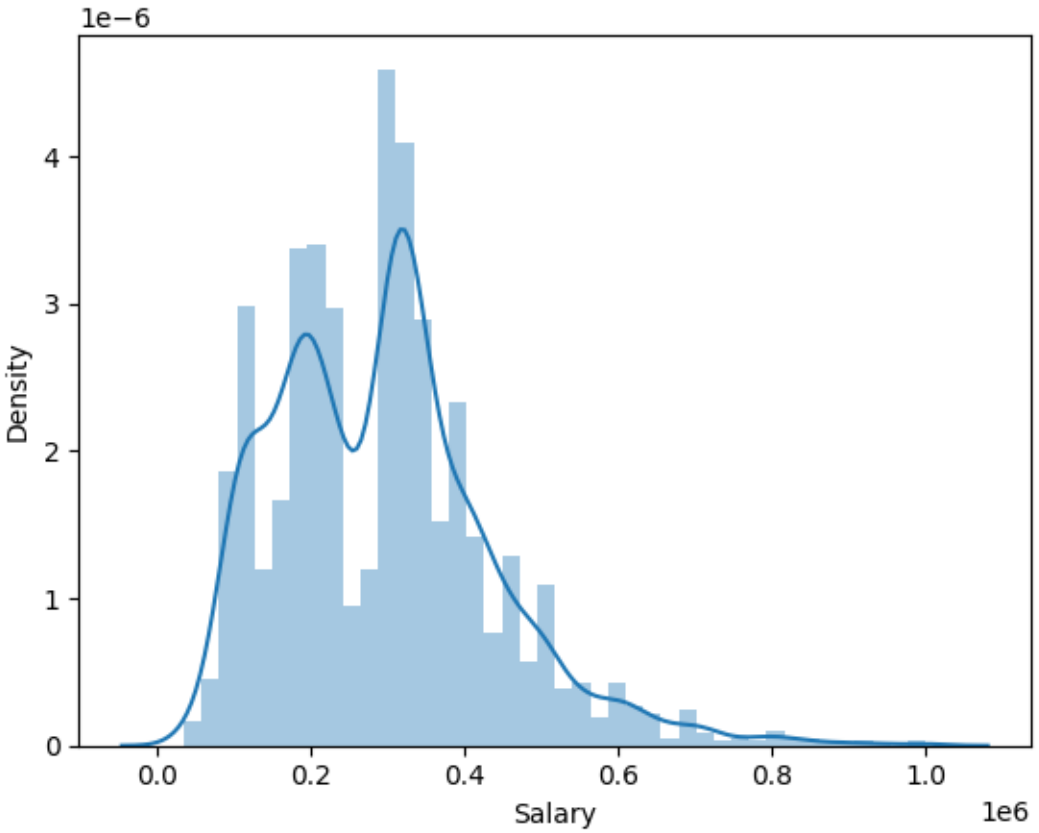
```
Number of Rows in dataframe in which Salary <= 250000: 1710
Number of Rows in dataframe in which Salary <= 500000: 3683
Number of Rows in dataframe in which Salary <= 750000: 3929
Number of Rows in dataframe in which Salary <= 1000000: 3962
Number of Rows in dataframe in which Salary <= 1250000: 3975
Number of Rows in dataframe in which Salary <= 1500000: 3981
Number of Rows in dataframe in which Salary <= 1750000: 3982
```

[15]:
```
indexNames = df[ df['Salary'] > 1000000 ].index
# Delete these row indexes from dataFrame
df.drop(indexNames , inplace=True)
df.shape
```

[15]: (3962, 38)

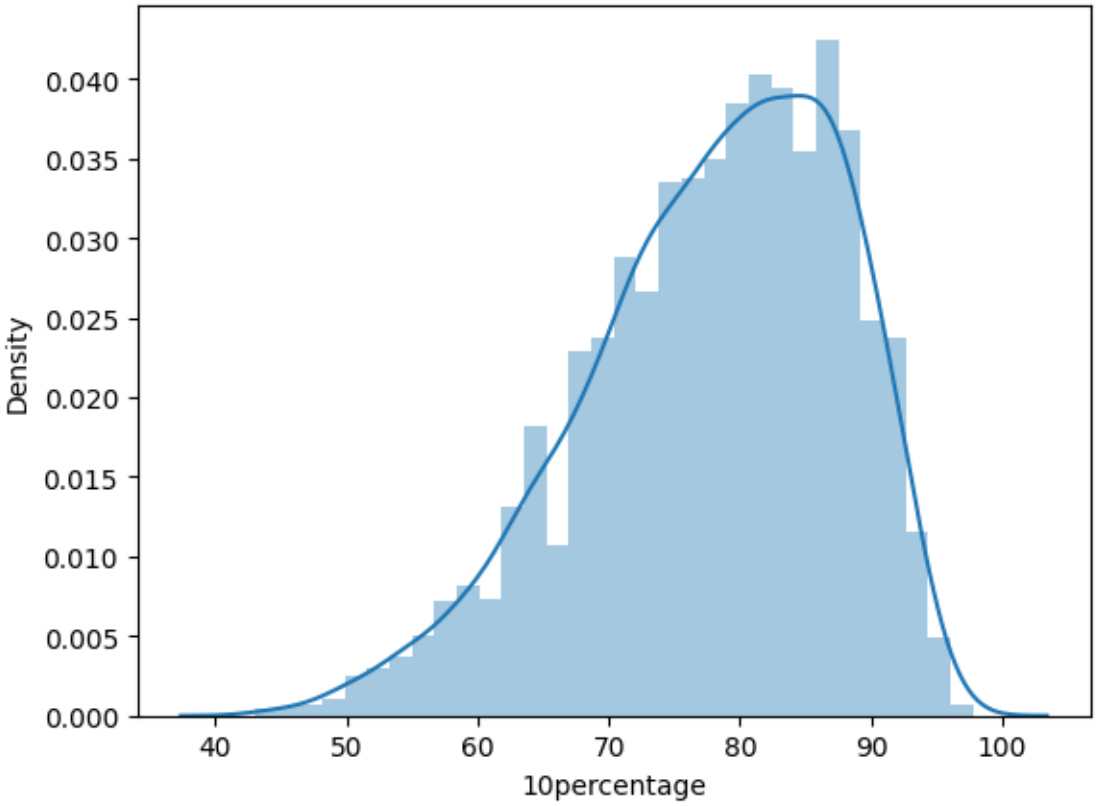
[16]:
```
sns.distplot(df['Salary']);
```

```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\2970929619.py:1:
UserWarning:

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

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

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

  sns.distplot(df['Salary']);
```



[17]:
```
sns.distplot(df["10percentage"]);
```

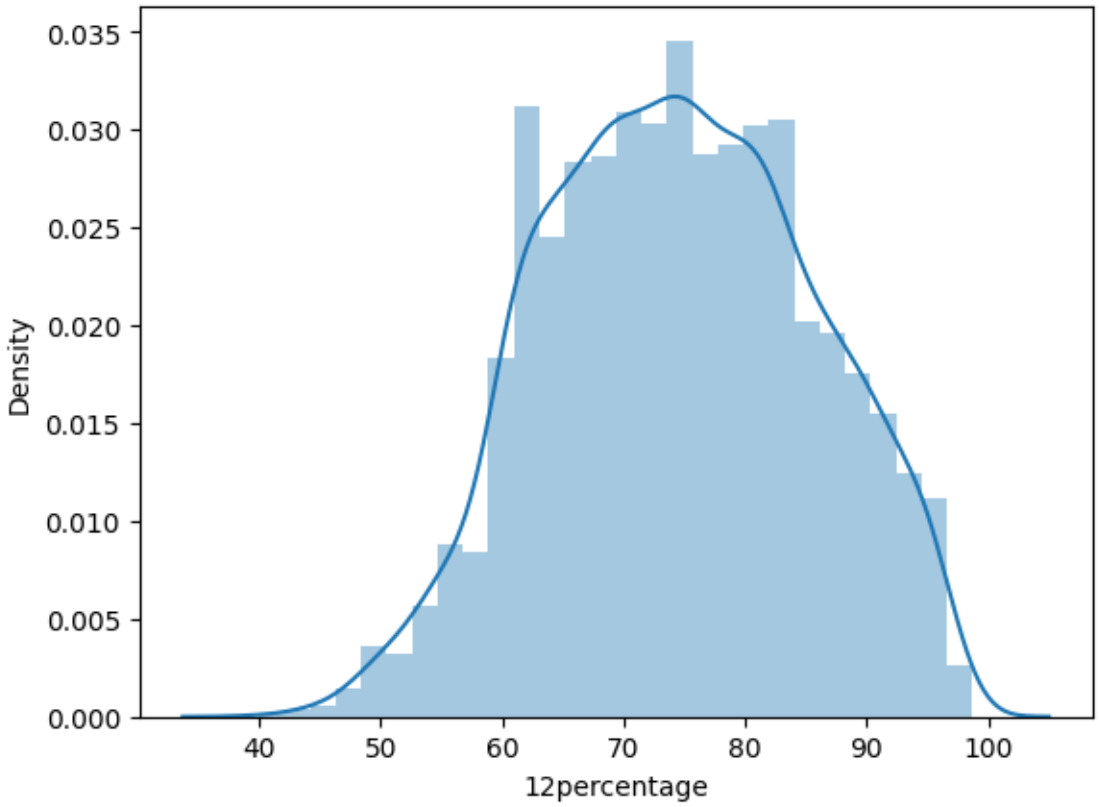
```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\3025552207.py:1:
UserWarning:

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

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

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

  sns.distplot(df["10percentage"]);
```



- The 10percentage column is not normally distributed and is Right Skewed and the max is in range 80-90

[18]:
```
sns.distplot(df["12percentage"]);
```

```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\729720608.py:1:
UserWarning:

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

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

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

  sns.distplot(df["12percentage"]);
```

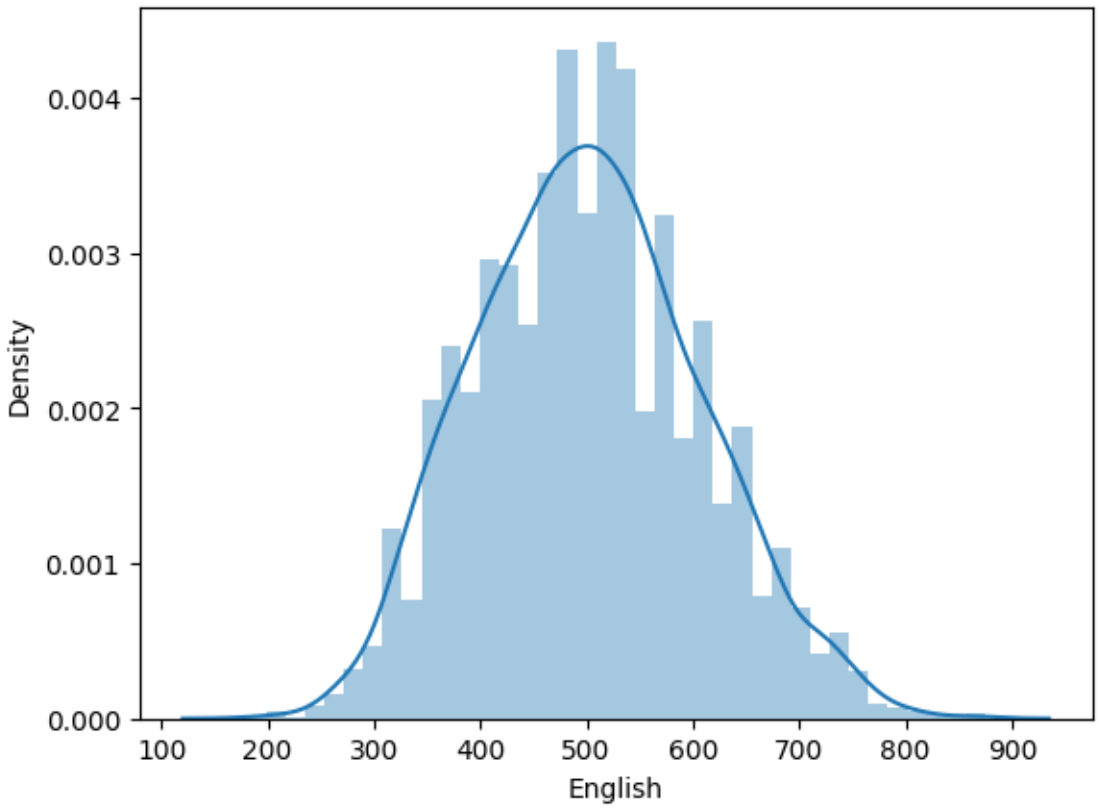


- The 12percentage column is not normally distributed and outliers and the max is in range 70-80

[19]:
```
sns.distplot(df["English"]);
```

```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\1283016064.py:1:
UserWarning:

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

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

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

  sns.distplot(df["English"]);
```

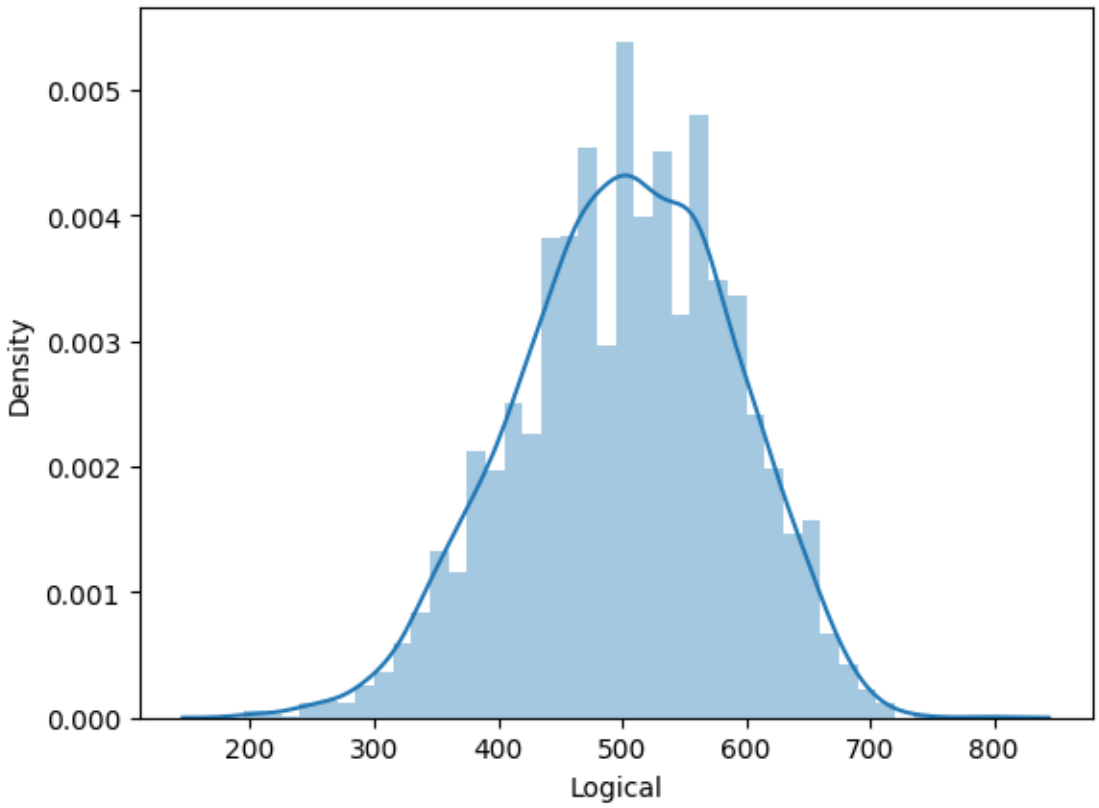


- The English column is normally distributed and has max in range 400-600

[20]:
```
sns.distplot(df["Logical"]);
```

```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\4192468787.py:1:
UserWarning:

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

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

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

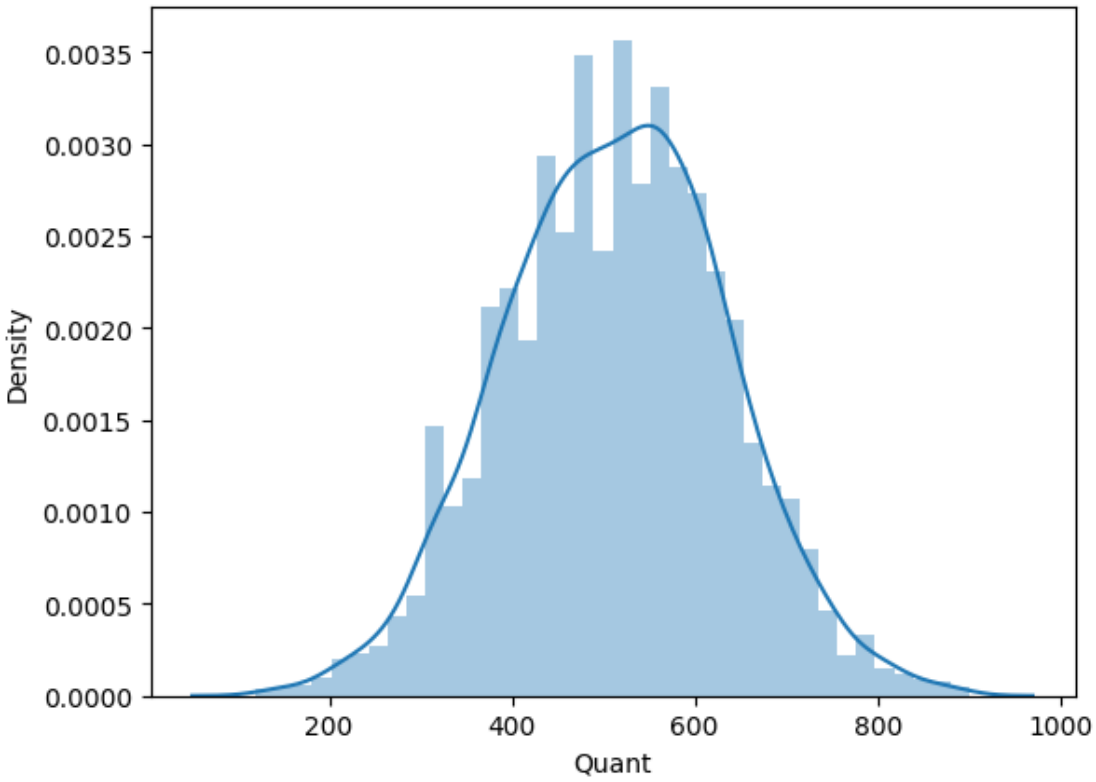
```
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Logical"]);
```



- The Logical column is normally distributed and has max in range 400-700

[21]:
```
sns.distplot(df["Quant"]);
```

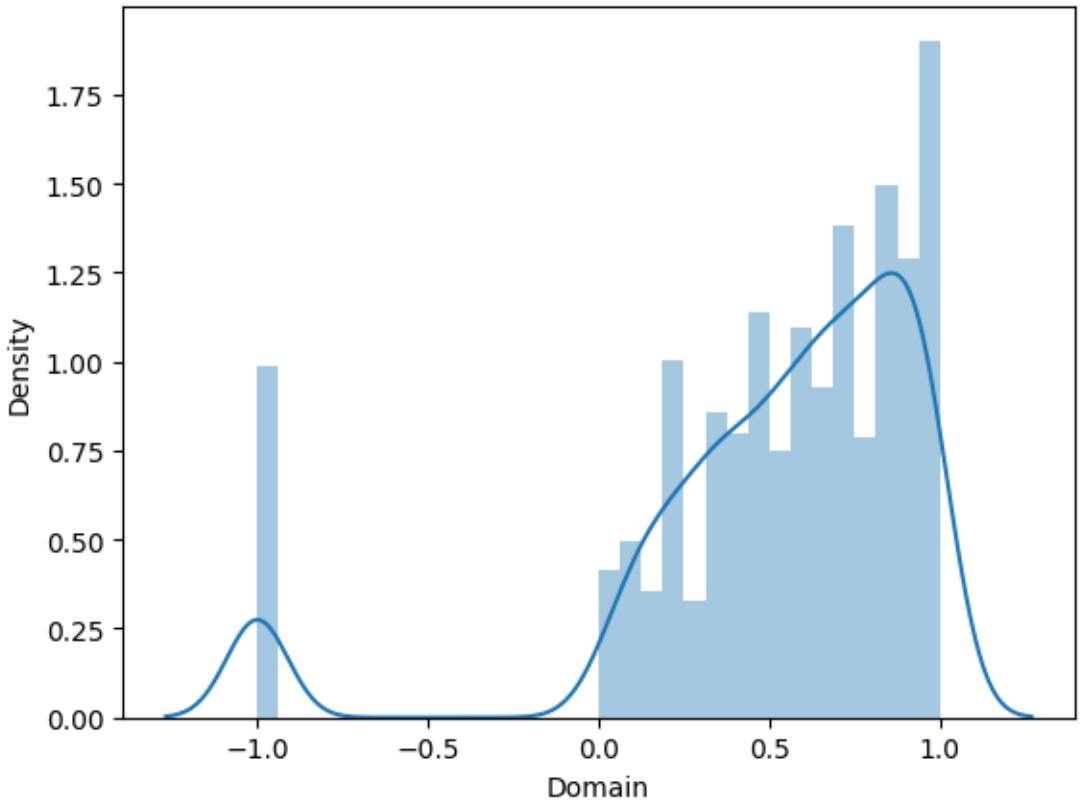
```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\2999565833.py:1:
UserWarning:

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

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

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

  sns.distplot(df["Quant"]);
```



- The Quant column is normally distributed and has max in range 400-800

[22]:
```
sns.distplot(df["Domain"]);
```

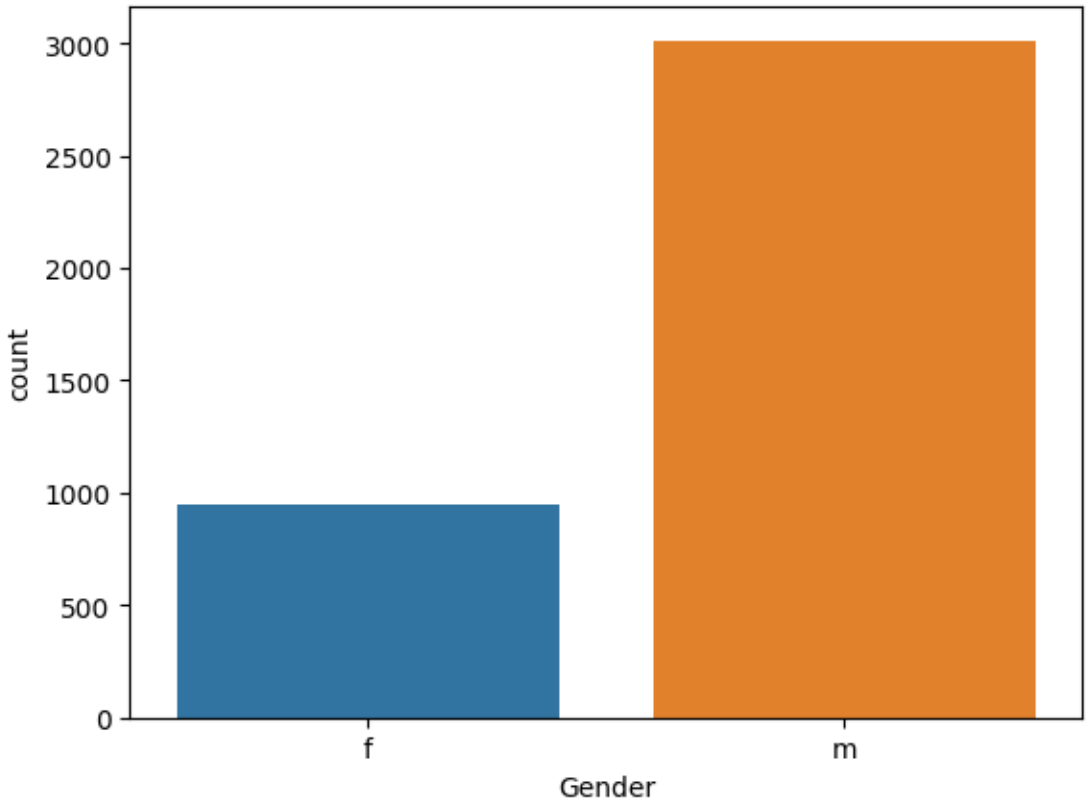
```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\1138102185.py:1:
UserWarning:

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

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

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

  sns.distplot(df["Domain"]);
```



- The collegecityid column is not normally distributed and has outliers

[23]:
```
sns.countplot(x='Gender',data=df)
```

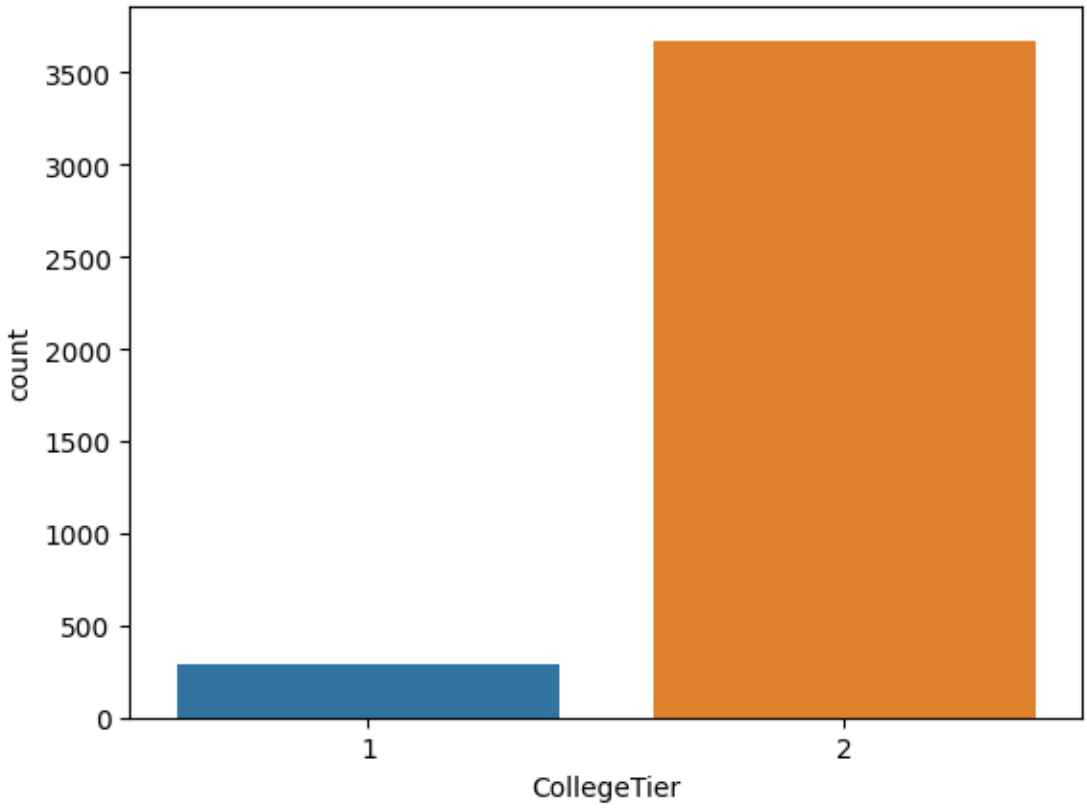
[23]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



- From above graph the count of males are more compared to females

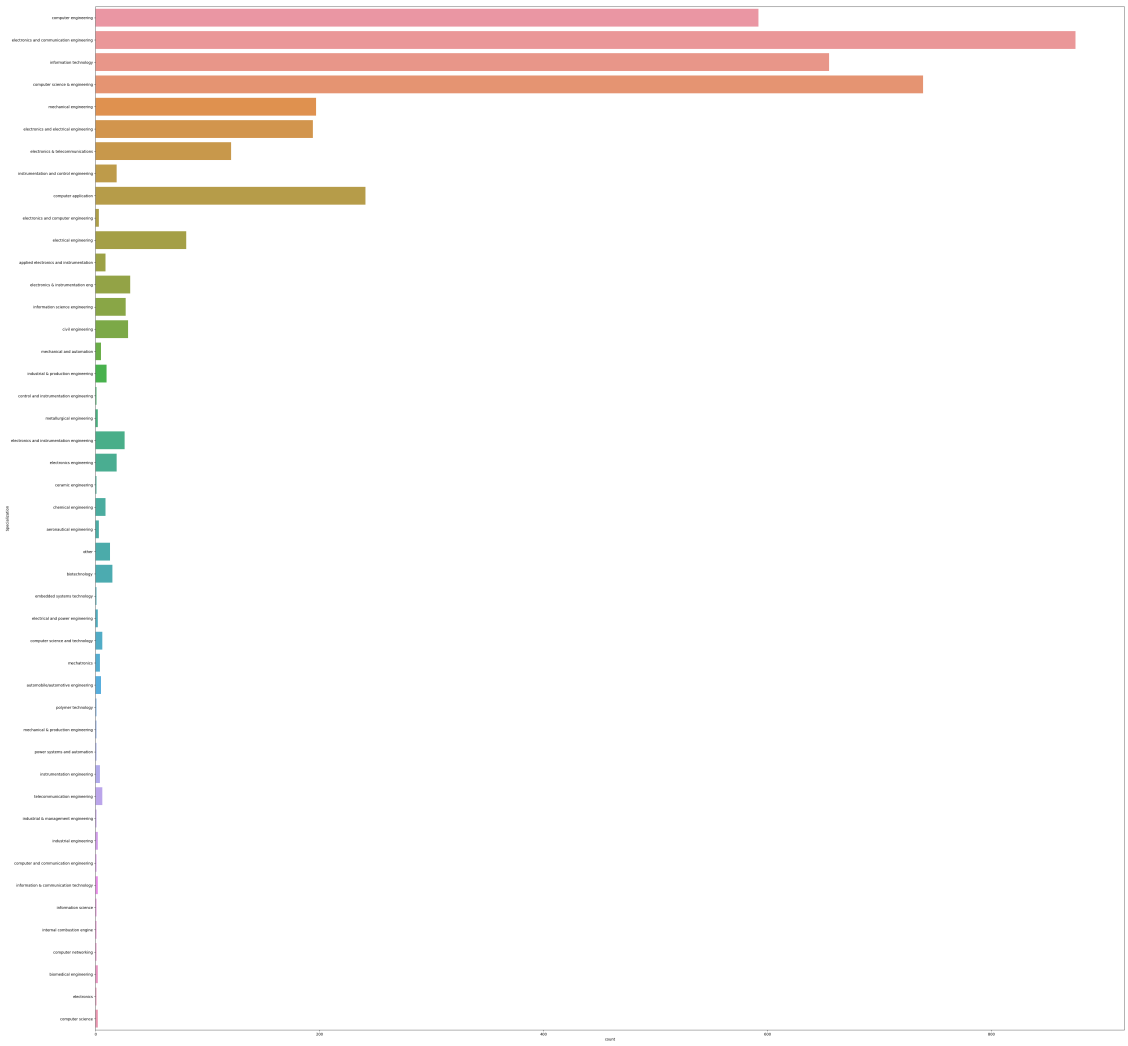
[24]:
```
sns.countplot(x='CollegeTier',data=df)
```

[24]:
```
<Axes: xlabel='CollegeTier', ylabel='count'>
```



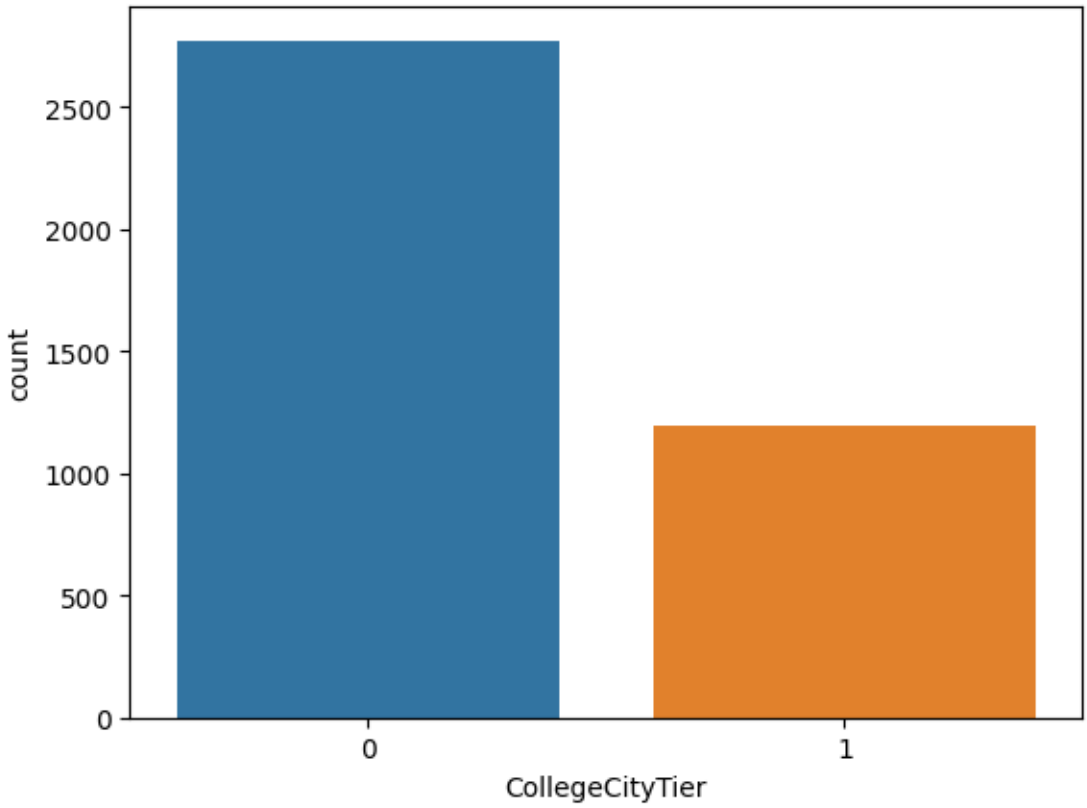
- The count of tier 2 college is more!!

[25]:
```
plt.figure(figsize=(50, 50))
ax = sns.countplot(y="Specialization", data=df)
```



[26]:
```
sns.countplot(x='CollegeCityTier',data=df)
```

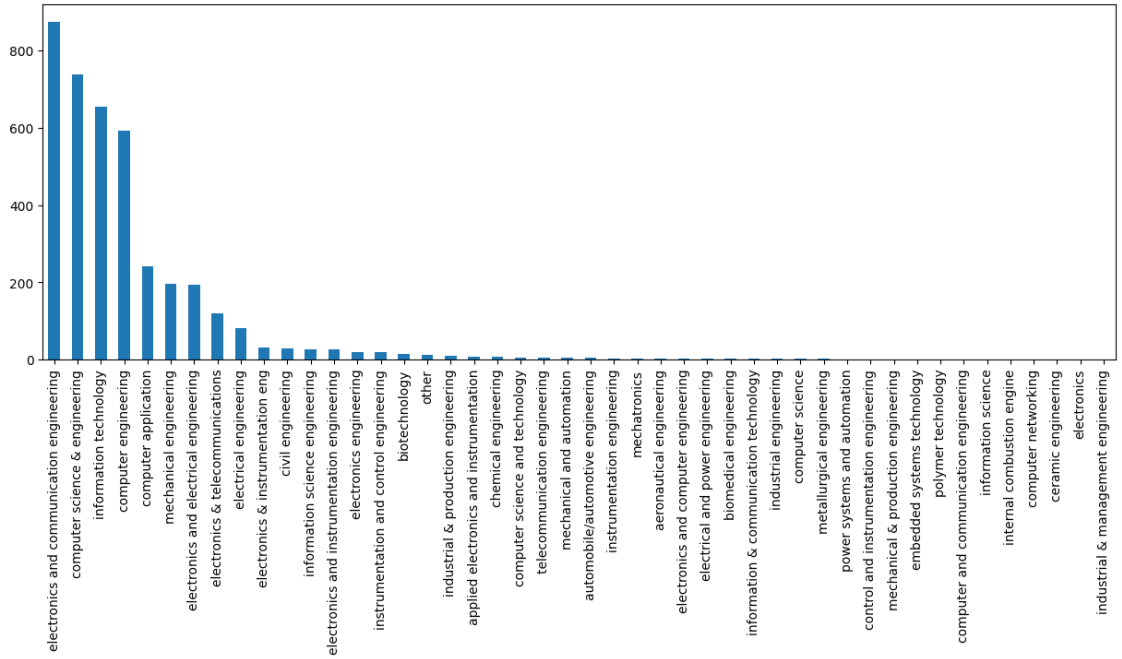
[26]: <Axes: xlabel='CollegeCityTier', ylabel='count'>



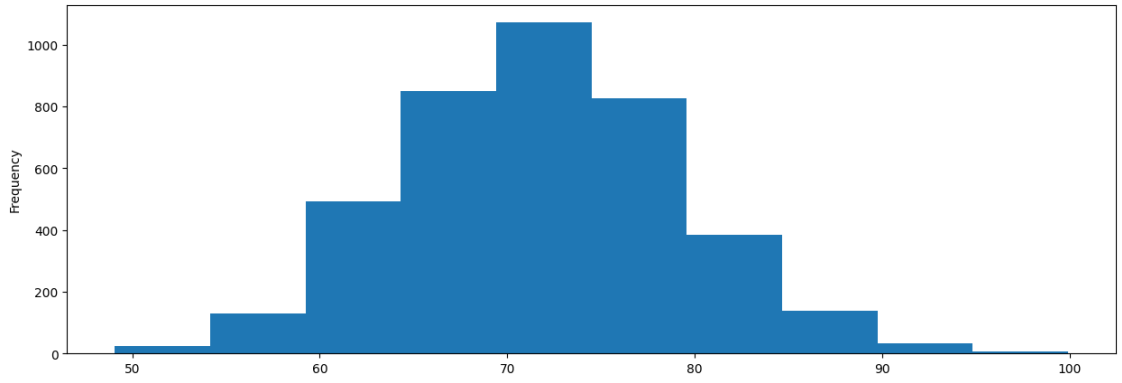
- The Collegecitytier 0 has high frequency to that of collegecitytier 1

[27]:
```
specialization_freq = df['Specialization'].value_counts()
specialization_freq.plot(kind='bar', figsize=(15,5))
```

[27]: <Axes: >



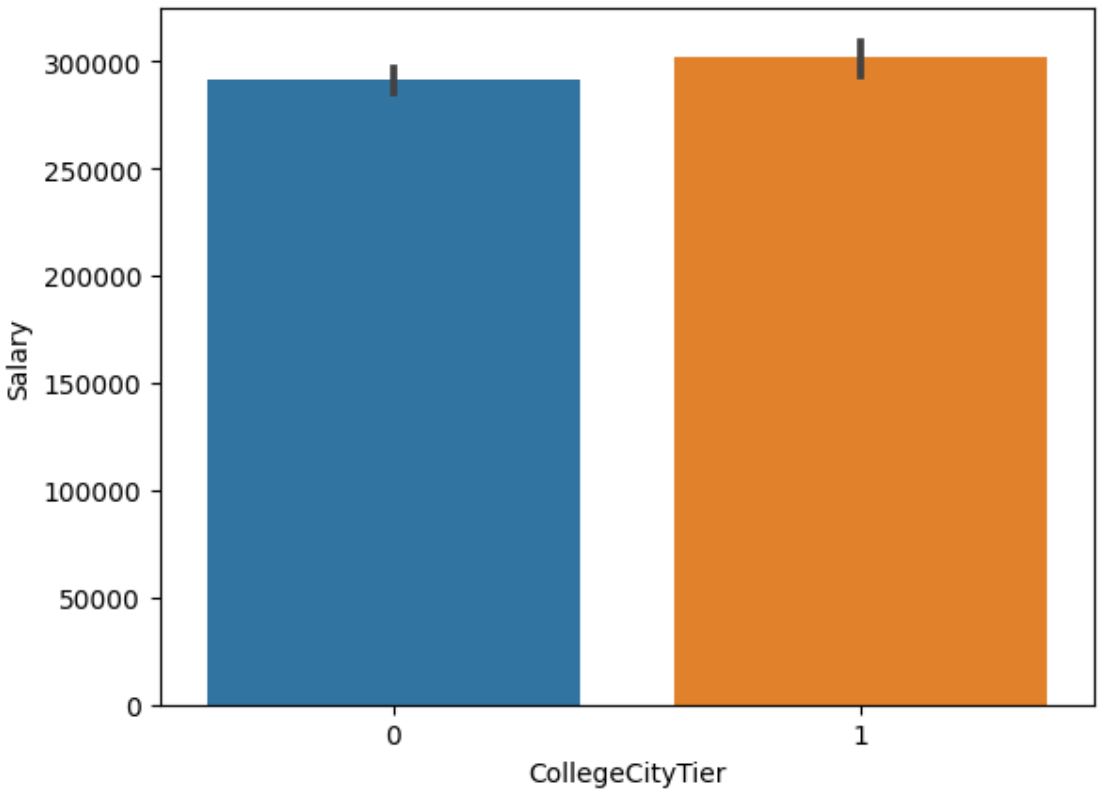
[28]:
```
df.loc[df['collegeGPA'] <= 10, 'collegeGPA'] *= 10
df['collegeGPA'].plot(kind='hist', figsize=(15,5));
```



- Bringing the CGPA to a 0-100 scale

[29]:
```
sns.barplot(x='CollegeCityTier',y='Salary',data=df)
```

[29]:
```
<Axes: xlabel='CollegeCityTier', ylabel='Salary'>
```



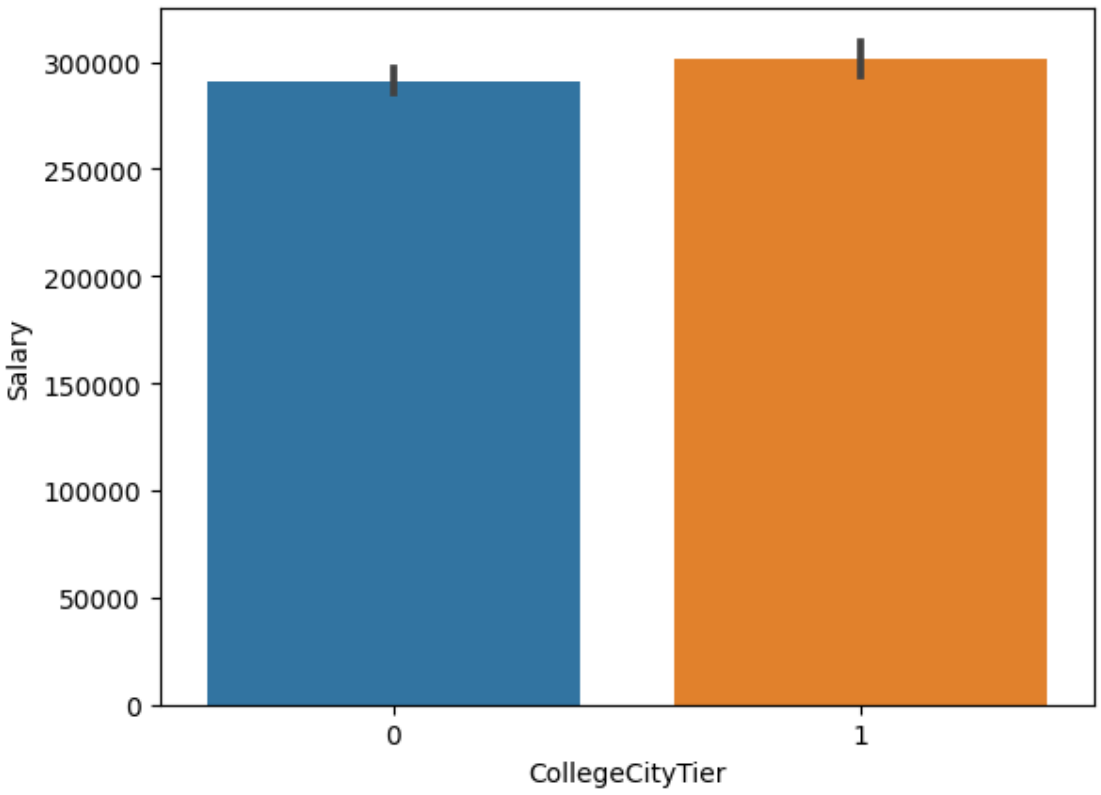
[ ]:

[ ]:

1.3 Bivariate Analysis

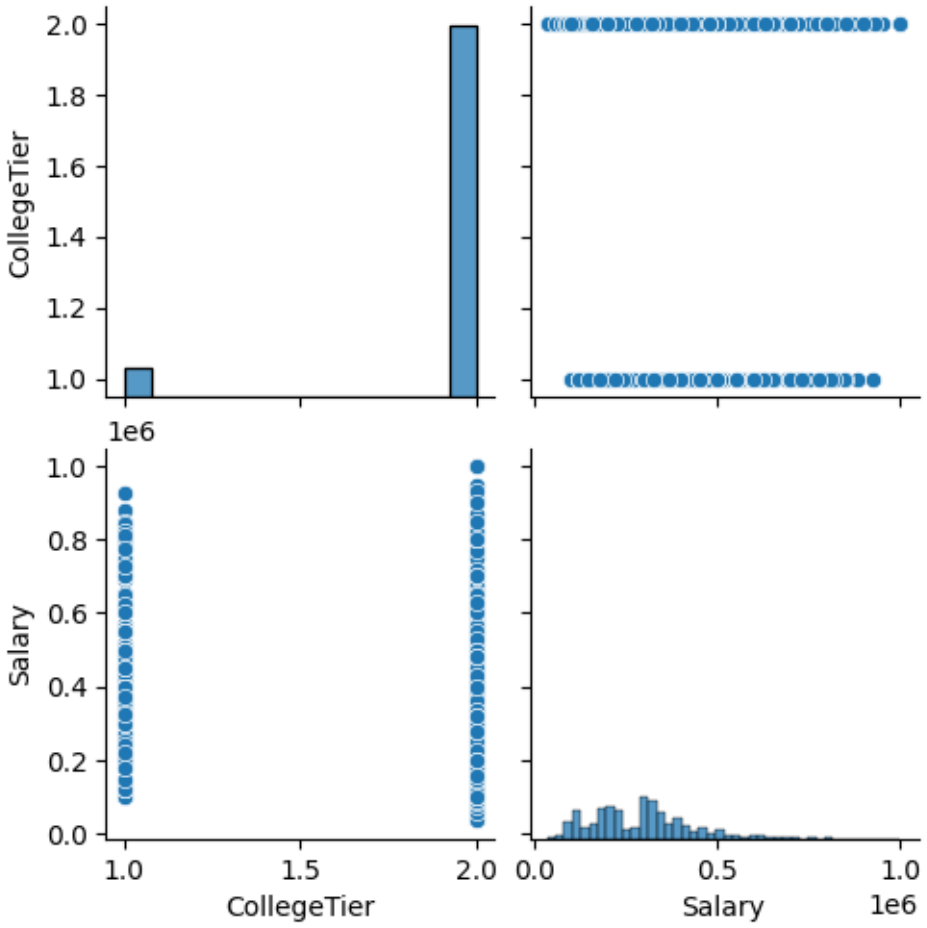
[30]:
```
sns.barplot(x='CollegeCityTier',y='Salary',data=df)
```

[30]: <Axes: xlabel='CollegeCityTier', ylabel='Salary'>



[31]:
```
sns.pairplot(df,vars=['CollegeTier', 'Salary'])
```

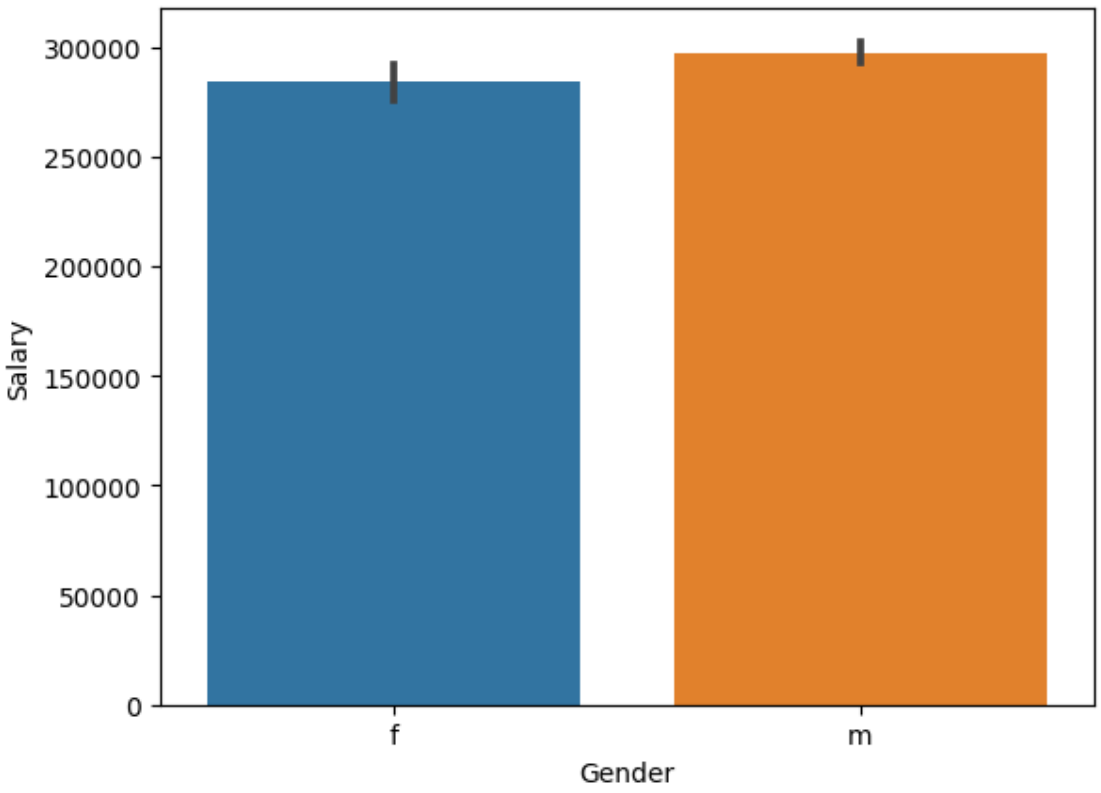
[31]: <seaborn.axisgrid.PairGrid at 0x2934cf8f640>



- From the graph we observe that collegecitytier 1 has bagged with highest salary , and also to be noted that collegecity tier 0 also provide the same salary expectations

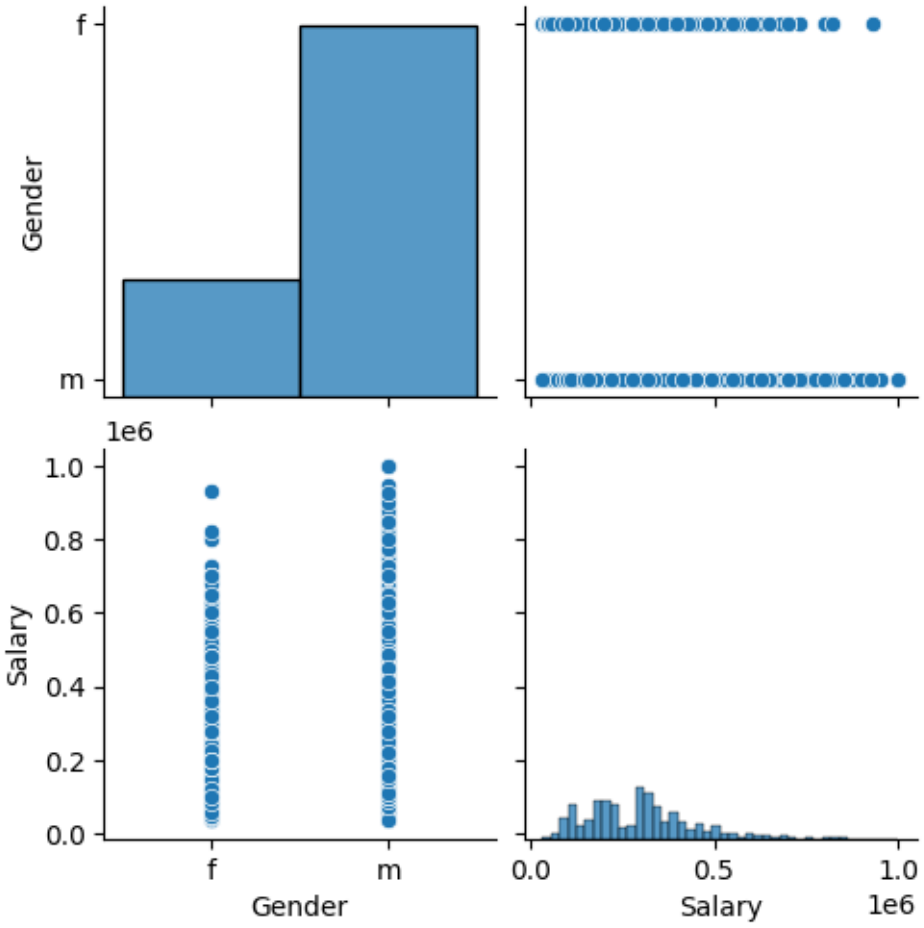
[32]:
```
sns.barplot(x='Gender',y='Salary',data=df)
```

[32]:
```
<Axes: xlabel='Gender', ylabel='Salary'>
```



[33]:
```
sns.pairplot(df,vars=['Gender', 'Salary'])
```

[33]: <seaborn.axisgrid.PairGrid at 0x2934ea06d70>



- Males and females take the salary more or less the same

[34]:
```
l = []
for i in df['Designation']:
    if 'senior' in i and 'engineer' not in i:
        l.append('senior')
    elif 'trainee' in i and 'engineer' not in i:
        l.append('trainee')
    elif 'engineer' in i and 'senior' not in i:
        l.append('engineer')
    elif 'associate' in i and 'senior' not in i:
        l.append('associate')
    elif 'developer' in i and 'senior' not in i:
        l.append('developer')
    elif 'manager' in i and 'senior' not in i:
        l.append('manager')
    elif 'analyst' in i:
        l.append('analyst')
```

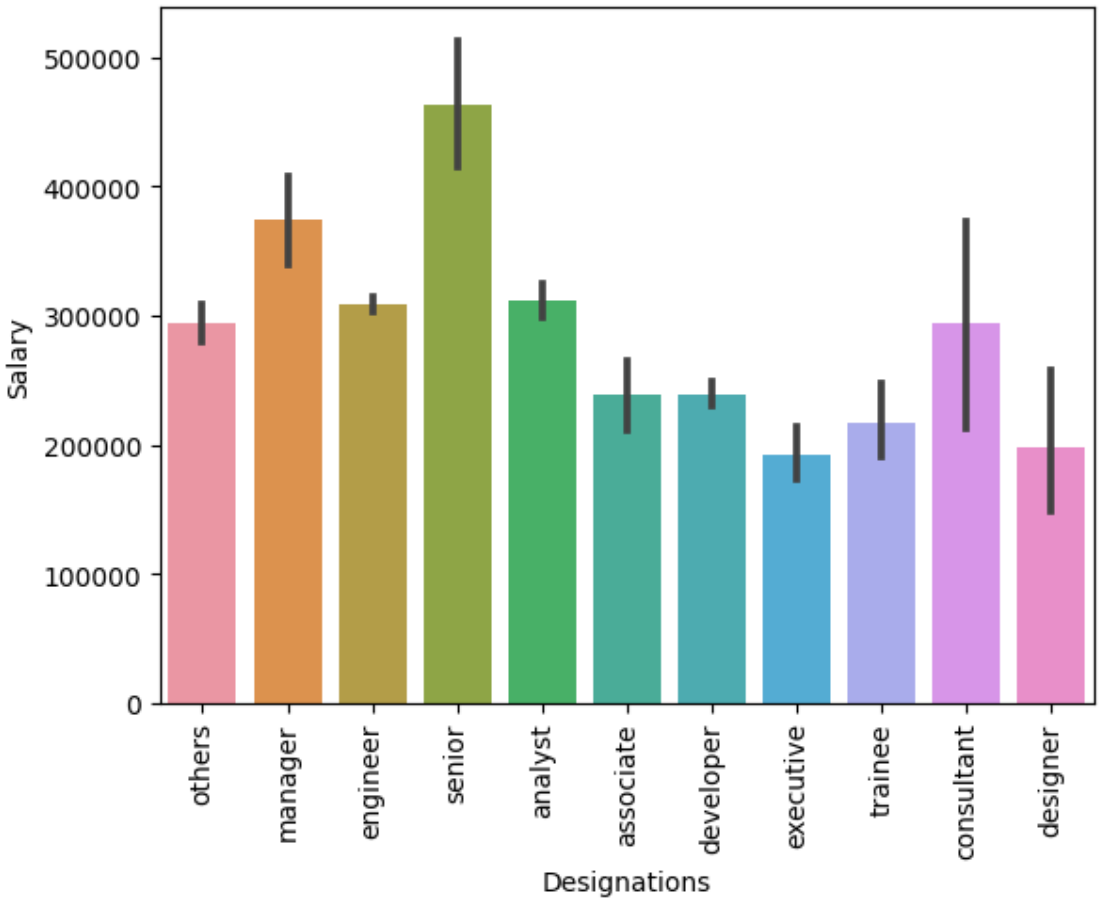
```
    elif 'consultant' in i:
        l.append('consultant')
    elif 'executive' in i:
        l.append('executive')
    elif 'designer' in i:
        l.append('designer')
    else:
        l.append('others')
```

[35]:
```
df['Designations']=l
df['Designations'].value_counts()
```

[35]:
```
engineer      1984
developer      663
others         528
analyst        401
manager        119
associate       65
executive       62
trainee         57
senior          41
designer        23
consultant      19
Name: Designations, dtype: int64
```

[36]:
```
sns.barplot(x='Designations',y='Salary',data=df)
 plt.xticks(rotation=90)
```

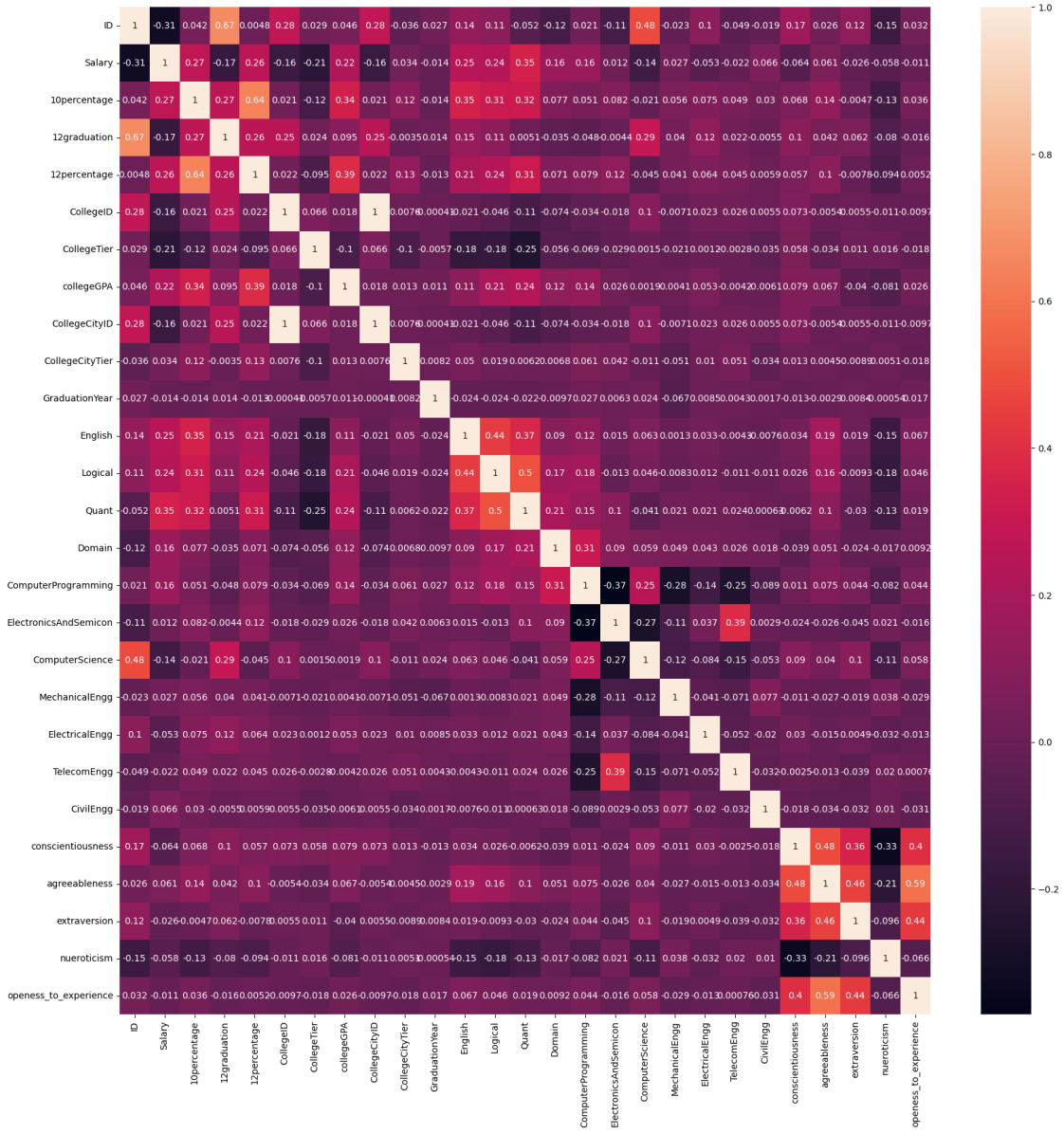
[36]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 [Text(0, 0, 'others'),
  Text(1, 0, 'manager'),
  Text(2, 0, 'engineer'),
  Text(3, 0, 'senior'),
  Text(4, 0, 'analyst'),
  Text(5, 0, 'associate'),
  Text(6, 0, 'developer'),
  Text(7, 0, 'executive'),
  Text(8, 0, 'trainee'),
  Text(9, 0, 'consultant'),
  Text(10, 0, 'designer')])
```



[37]:
```
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(),annot=True)
```

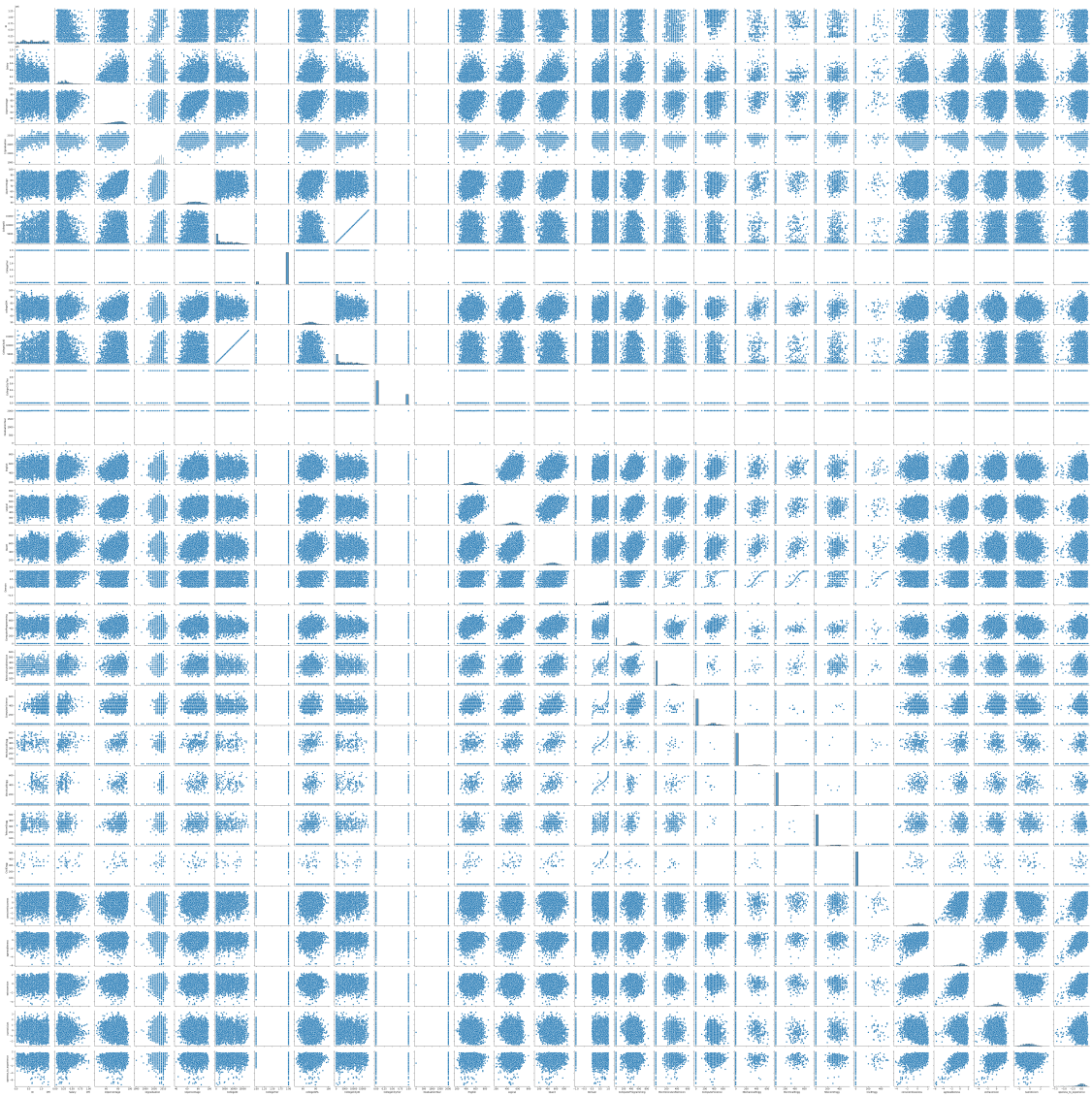
```
C:\Users\Manikanta\AppData\Local\Temp\ipykernel_7808\2710085310.py:2:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

[37]: <Axes: >



[39]:
```
sns.pairplot(data = df)
```

[39]: <seaborn.axisgrid.PairGrid at 0x2934cd5bd60>



[40]:
```
sns.pairplot(data = df, diag_kind='kde')
```

[40]: <seaborn.axisgrid.PairGrid at 0x2934ef8f5b0>

1.4 Research Questions

- Research Question 1: Testing the Claim about Computer Science Engineering Jobs
- Research Question 2: Relationship between Gender and Specialization

[ ]:
```
# Computer Science Engineers with specified job titles
computer_science_jobs = df[(df['Degree'] == 'ComputerScience') &
                           (df['Designation'].isin(['Programming Analyst',␣
↪'Software Engineer',
                                                    'Hardware Engineer',␣
↪'Associate Engineer']))]
```

[ ]:
```
# Calculate average salary
average_salary = computer_science_jobs['Salary'].mean()
average_salary
```

[ ]:
```
# Gender and Specialization
contingency_table = pd.crosstab(df['Gender'], df['Specialization'])
contingency_table
```

[ ]:
```
# Chi-square test of independence
chi2, p, dof, expected = chi2_contingency(contingency_table)
```

[ ]:
```
chi2
```

[ ]:
```
p
```

1.5 Conclusion

- The dataset comprises candidate details including ID, salary, job tenure, designation, location, gender, education, and personality traits.
- Insights cover salary trends, demographics, education profiles, and personality influences on careers.
- Understanding these facets informs recruitment, roles, and salary dynamics.

1.6 Bonus

- Correlation between salary and education level.
- Distribution of salaries across different cities or job titles.
- Impact of college tier on salary expectations.
- Trends in job tenure and salary growth over time.
- Gender diversity and its impact on salary and job roles.
- Comparison of salary distributions across different industries or sectors.

1.7 My Own research

- Numerical Features: Perform Column Standardization by subtracting the mean and dividing by the standard deviation.
- Categorical Features: For categories more than 2, use dummy variables (one-hot encoding). For binary categories, convert to 0 or 1.
- Use scikit-learn's 'StandardScaler' for numerical standardization.
- Utilize pandas' 'get_dummies()' function for one-hot encoding and map function for binary conversion.

[ ]:
```

```