```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
plt.style.use(['dark_background'])

import seaborn as sns
sns.set(color_codes=True)
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarni
  import pandas.util.testing as tm
```

```
import matplotlib.pyplot as plt
from matplotlib import style
plt.style.use(['dark_background'])

import urllib.request
import json

import seaborn as sns
sns.set(color_codes=True)
```

```
data = pd.read_excel('train.xlsx')
```

```
data.head()
```

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

```
data.describe()
```

	ID	Salary	10percentage	12graduation	12percentage	Colle
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.00
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.85
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.26
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.00
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.00
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.00
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.00
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.00

```
data.dtypes
```

```
Unnamed: 0                    object
ID                             int64
Salary                         int64
DOJ                   datetime64[ns]
DOL                           object
Designation                   object
JobCity                       object
Gender                        object
DOB                   datetime64[ns]
10percentage                 float64
10board                       object
12graduation                   int64
12percentage                 float64
12board                       object
CollegeID                      int64
CollegeTier                    int64
Degree                        object
Specialization                object
collegeGPA                   float64
CollegeCityID                  int64
CollegeCityTier                int64
CollegeState                  object
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
```

```
df = data
```

```
col_sal = data[['Salary','CollegeTier', 'GraduationYear', 'CollegeState']]
```

```
col_sal
```

	Salary	CollegeTier	GraduationYear	CollegeState
0	420000	2	2011	Andhra Pradesh
1	500000	2	2012	Madhya Pradesh
2	325000	2	2014	Uttar Pradesh
3	1100000	1	2011	Delhi
4	200000	2	2012	Uttar Pradesh
...	...	...	...	...
3993	280000	2	2010	Haryana
3994	100000	2	2013	Telangana
3995	320000	2	2012	Orissa
3996	200000	2	2014	Karnataka
3997	400000	2	2012	Tamil Nadu

3998 rows × 4 columns

```
col_sal.apply(lambda x: x.count(), axis=1)
```

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

```
col_sal.isnull().sum(axis = 0) # checking if any calumn
```

```
Salary           0
CollegeTier      0
GraduationYear   0
CollegeState     0
dtype: int64
```

```
col_sal.isnull().sum(axis = 1) #cheackinh each row
```

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

```
col_sal.plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0a98ce5eb8>
```


```
data.columns
```

```
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')
```

```
data.set_index("ID" ,inplace=True)
```

```
data.head()
```

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

```
col_sal.set_index("CollegeState", inplace=True)
```

343523	train	200000	[illegible]	03-01	get	Manesar	m	[illegible]	

```
df = col_sal

col_sal
```

	Salary	CollegeTier	GraduationYear	CollegeState
0	420000	2	2011	Andhra Pradesh
1	500000	2	2012	Madhya Pradesh
2	325000	2	2014	Uttar Pradesh
3	1100000	1	2011	Delhi
4	200000	2	2012	Uttar Pradesh
...	...	...	...	...
3993	280000	2	2010	Haryana
3994	100000	2	2013	Telangana
3995	320000	2	2012	Orissa
3996	200000	2	2014	Karnataka
3997	400000	2	2012	Tamil Nadu

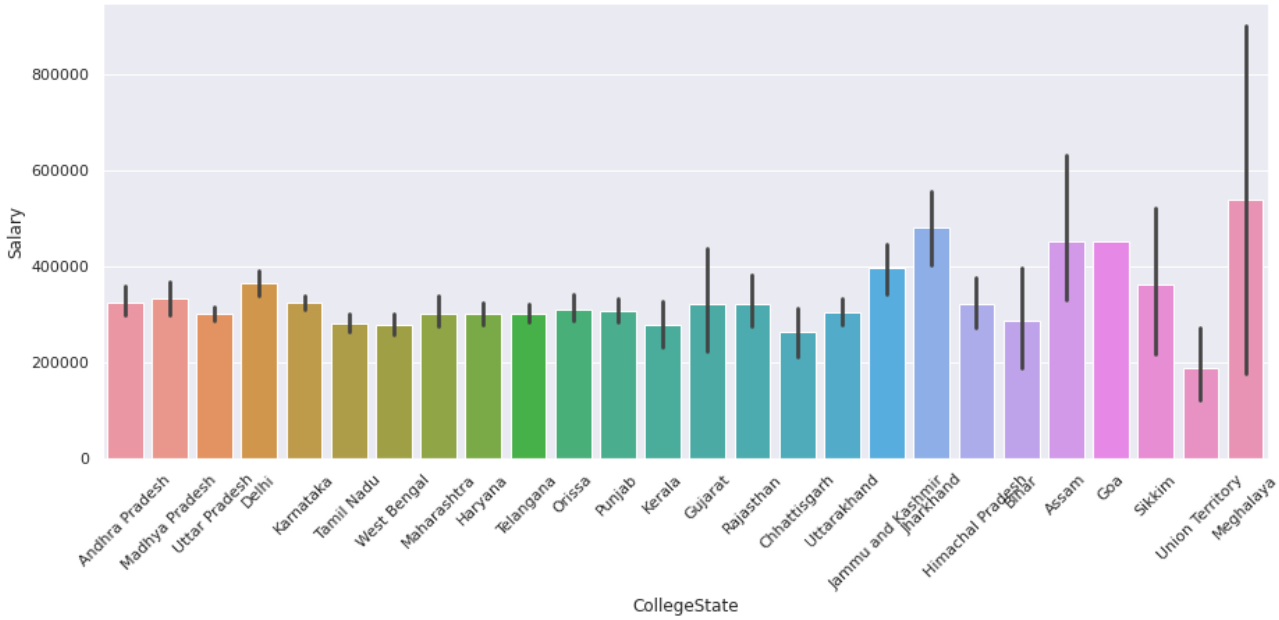
3998 rows × 4 columns

```
df = df.groupby('CollegeState').nunique()

df
```

CollegeState	Salary	CollegeTier	GraduationYear	CollegeState
Andhra Pradesh	79	2	9	1
Assam	5	2	3	1
Bihar	8	2	6	1
Chhattisgarh	19	2	4	1
Delhi	56	2	7	1
Goa	1	1	1	1
Gujarat	19	2	5	1
Haryana	69	2	7	1
Himachal Pradesh	13	2	2	1
Jammu and Kashmir	6	2	3	1
Jharkhand	25	2	6	1
Karnataka	94	2	8	1
Kerala	25	2	5	1
Madhya Pradesh	80	2	7	1
Maharashtra	81	2	7	1
Meghalaya	2	1	1	1
Orissa	68	2	7	1
Punjab	65	2	7	1
Rajasthan	63	2	7	1
Sikkim	3	1	2	1
Tamil Nadu	91	2	8	1
Telangana	85	2	8	1
Union Territory	5	1	2	1

```
fig = plt.gcf();
fig.set_size_inches(15, 6);
ax=sns.barplot(x="CollegeState", y="Salary", data=col_sal);
#ax.set(xlabel="CollegeState", ylabel = "Salary")
ax.set_xticklabels(ax.get_xticklabels(), rotation=45);
```




```
for i in range(1,8):
    seriesObj = data.apply(lambda x: True if x['Salary'] <= 250000*i  else False , axis=
    # Count number of True in series
    numOfRows = len(seriesObj[seriesObj == True].index)

    print('Number of Rows in dataframe in which Salary %d  : '%((250000*i)), numOfRows)
```

```
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
```

```
sal_high = data[data.Salary >980000]
```

```
sal_high
```

2152	train	323688	1200000	07-01	present	software engineer	Bangalore	m
2182	train	41147	4000000	2010-01-01	2011-12-01 00:00:00	automation engineer	gurgaon	m
2216	train	202950	1800000	2012-12-01	2015-02-01 00:00:00	client services associate	Bangalore	f
2230	train	107796	1200000	2011-01-01	present	engineer	Mumbai	m
2412	train	42943	1000000	2010-09-01	present	assistant manager	Vadodara	m
2472	train	1283372	1000000	2015-01-01	present	software engineer	Noida	m
2493	train	644790	1745000	2013-07-01	present	java software engineer	Johannesburg	m
2541	train	1045685	2000000	2015-06-01	present	data scientist	LONDON	m
2565	train	1254777	1800000	2015-01-01	2015-04-01 00:00:00	salesforce developer	Panchkula	f
2764	train	108231	1200000	2010-08-01	2011-06-01 00:00:00	office coordinator	Hyderabad	f
2880	train	608841	1030000	2013-07-01	present	senior software developer	Bangalore	f
3126	train	87319	1210000	2010-10-01	2011-09-01 00:00:00	get	Bhopal	m
3247	train	768298	1500000	2015-06-01	2015-06-01 00:00:00	software engineer	Bangalore	m
3276	train	34551	1100000	2010-06-01	present	design engineer	Greater Noida	m
3484	train	615010	2000000	2013-09-01	2014-06-01 00:00:00	it technician	Noida	m
3490	train	803778	2000000	2013-07-01	2014-10-01 00:00:00	technical lead	Pune	m
3710	train	271904	1100000	2011-11-01	present	senior software engineer	Bangalore	m

```
data['DOB']=pd.to_datetime(data['DOB'])
data['Dyear']=data['DOB'].dt.year
data.drop(columns=['DOB'],axis=1,inplace=True)
```

```
data.head()
```

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

```
data.drop(['Unnamed: 0'], axis=1, inplace=True)
```

```
data.set_index("ID", inplace=True)
```

```
data.head()
```

```
            Salary    DOJ      DOL  Designation   JobCity  Gender  10percentage     10l
        ID
```

```
data['DOJ']=pd.to_datetime(data['DOJ'])
data['JYEAR']=data['DOJ'].dt.year
data.drop(columns=['DOJ'],axis=1,inplace=True)
```

```
                    2013                assistant
```

```
data.head()
```

ge	10board	12graduation	12percentage	12board	CollegeID	CollegeTier	De
.3	board ofsecondary education,ap	2007	95.8	board of intermediate education,ap	1141	2	B.Tech
.4	cbse	2007	85.0	cbse	5807	2	B.Tech
.0	cbse	2010	68.2	cbse	64	2	B.Tech
.6	cbse	2007	83.6	cbse	6920	1	B.Tech
.0	cbse	2008	76.8	cbse	11368	2	B.Tech

```
data['Specialization'].unique()
```

```
array(['computer engineering',
       'electronics and communication engineering',
       'information technology', 'computer science & engineering',
       'mechanical engineering', 'electronics and electrical engineering',
       'electronics & telecommunications',
       'instrumentation and control engineering', 'computer application',
       'electronics and computer engineering', 'electrical engineering',
```

```
specialization_map = \
{'electronics and communication engineering' : 'EC',
 'computer science & engineering' : 'CS',
 'information technology' : 'CS' ,
 'computer engineering' : 'CS',
 'computer application' : 'CS',
 'mechanical engineering' : 'ME',
 'electronics and electrical engineering' : 'EC',
 'electronics & telecommunications' : 'EC',
 'electrical engineering' : 'EL',
 'electronics & instrumentation eng' : 'EC',
 'civil engineering' : 'CE',
 'electronics and instrumentation engineering' : 'EC',
 'information science engineering' : 'CS',
 'instrumentation and control engineering' : 'EC',
 'electronics engineering' : 'EC',
 'biotechnology' : 'other',
 'other' : 'other',
 'industrial & production engineering' : 'other',
 'chemical engineering' : 'other',
 'applied electronics and instrumentation' : 'EC',
 'computer science and technology' : 'CS',
 'telecommunication engineering' : 'EC',
 'mechanical and automation' : 'ME',
 'automobile/automotive engineering' : 'ME',
 'instrumentation engineering' : 'EC',
 'mechatronics' : 'ME',
 'electronics and computer engineering' : 'CS',
 'aeronautical engineering' : 'ME',
 'computer science' : 'CS',
 'metallurgical engineering' : 'other',
 'biomedical engineering' : 'other',
 'industrial engineering' : 'other',
 'information & communication technology' : 'EC',
 'electrical and power engineering' : 'EL',
 'industrial & management engineering' : 'other',
 'computer networking' : 'CS',
 'embedded systems technology' : 'EC',
 'power systems and automation' : 'EL',
 'computer and communication engineering' : 'CS',
 'information science' : 'CS',
 'internal combustion engine' : 'ME',
 'ceramic engineering' : 'other',
 'mechanical & production engineering' : 'ME',
 'control and instrumentation engineering' : 'EC',
 'polymer technology' : 'other',
 'electronics' : 'EC'}
```

```
for i in range(1,8):
    seriesObj = data.apply(lambda x: True if x['Salary'] <= 250000*i  else False , axis=
    # Count number of True in series
    numOfRows = len(seriesObj[seriesObj == True].index)

    print('Number of Rows in dataframe in which Salary %d  : '%((250000*i)), numOfRows)
```

```
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
```

```
indexNames = data[ data['Salary'] > 1000000 ].index

# Delete these row indexes from dataFrame
data.drop(indexNames , inplace=True)
data.shape
```

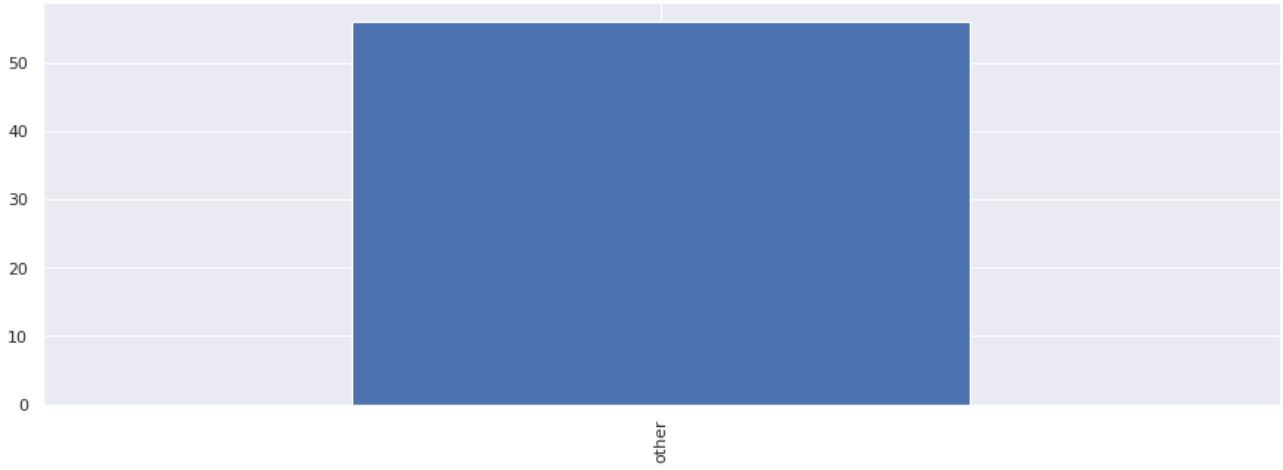
```
(3962, 36)
```

```
data['Specialization'] = data['Specialization'].map(specialization_map)

data['Specialization'].value_counts().plot(kind='bar', figsize=(15,5))

print(data['Specialization'].unique())
```

```
[nan 'other']
```




```
data.head()
```

onicsAndSemicon	ComputerScience	MechanicalEngg	ElectricalEngg	TelecomEngg	CivilEr
-1	-1	-1	-1	-1	
466	-1	-1	-1	-1	
-1	-1	-1	-1	-1	
233	-1	-1	-1	-1	

```
data.drop(columns=['DOL'], inplace=True)

clean_data = pd.DataFrame()

clean_data=data[data['ComputerScience']>0]


clean_data
```

ID	Salary	Designation	JobCity	Gender	10percentage	10board	12gradu
1027655	300000	system engineer	Hyderabad	m	89.92	state board	

```
clean_data.drop(columns=[
       'CollegeCityTier', '10board','CollegeState','ElectronicsAndSemicon',
        'MechanicalEngg', 'ElectricalEngg', 'TelecomEngg',
       'CivilEngg'],axis=1,inplace=True)
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:3997: SettingWithCopyWar
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
  errors=errors,
```

analyst board

```
clean_data.head()
```

ID	Salary	Designation	JobCity	Gender	10percentage	10board	12graduation
1027655	300000	system engineer	Hyderabad	m	89.92	state board	201(
947847	300000	java software engineer	Banglore	m	86.08	state board	201(
1279958	300000	java software engineer	Bangalore	m	81.20	state board	2008
874596	250000	associate software developer	Gurgaon	m	60.80	cbse	2006
963123	335000	programmer analyst	Hyderabad	m	88.00	state board	201(

```
clean_data.to_csv('trial_data_1.csv')
```

```
male_count = clean_data['Gender'].where(clean_data['Gender']=='m').count()
```

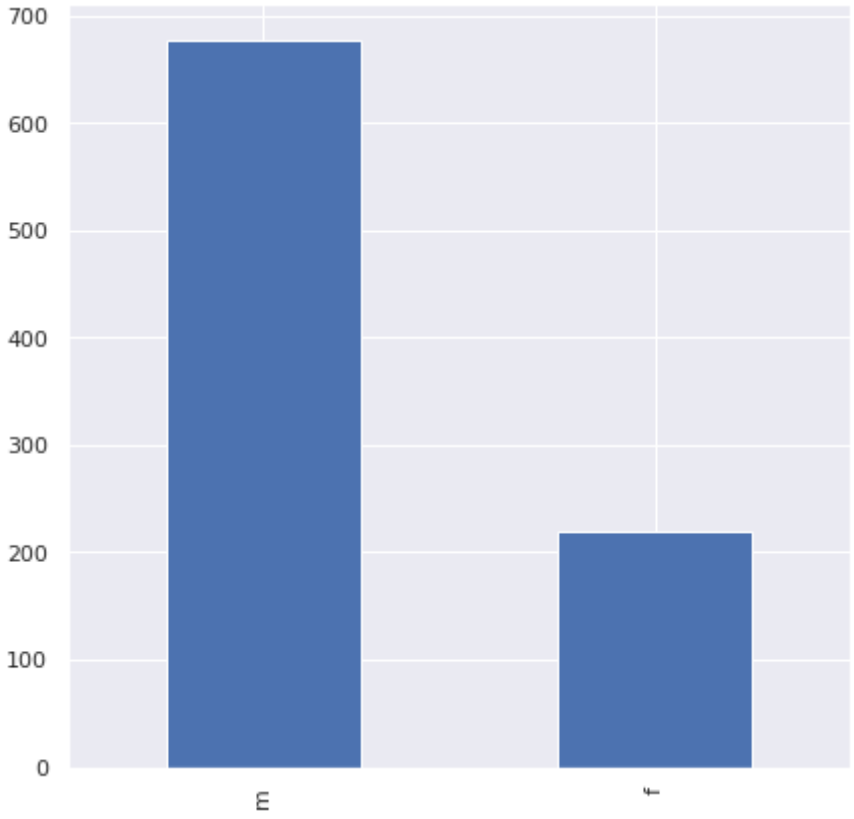
```
female_count = clean_data['Gender'].where(clean_data['Gender']=='f').count()
```

```
print(male_count)
print(female_count)
```

```
677
219
```

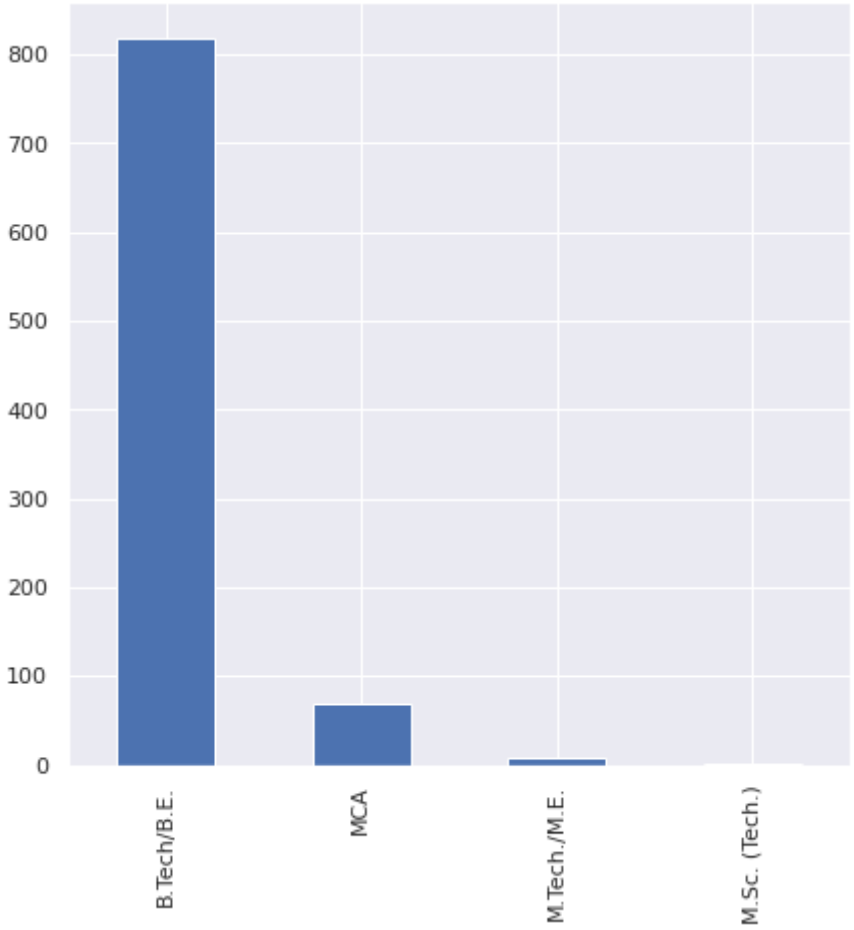
```
clean_data['Gender'].value_counts().plot(kind='bar', figsize=(7, 7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed2967f438>
```



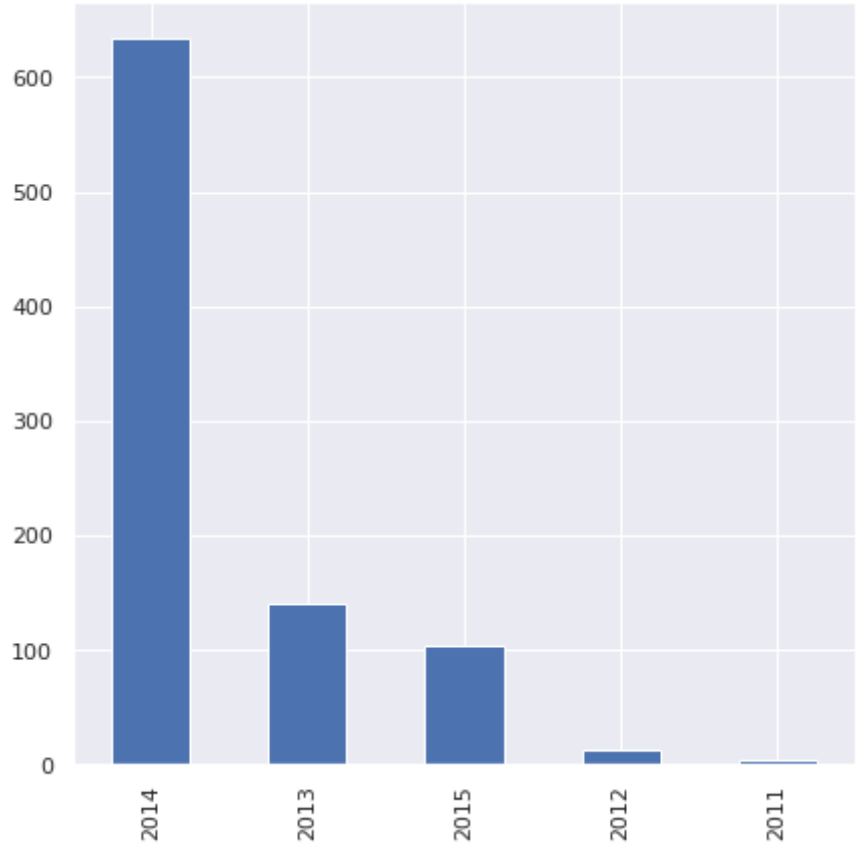

```
clean_data['Degree'].value_counts().plot(kind='bar', figsize=(7,7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed296811d0>
```



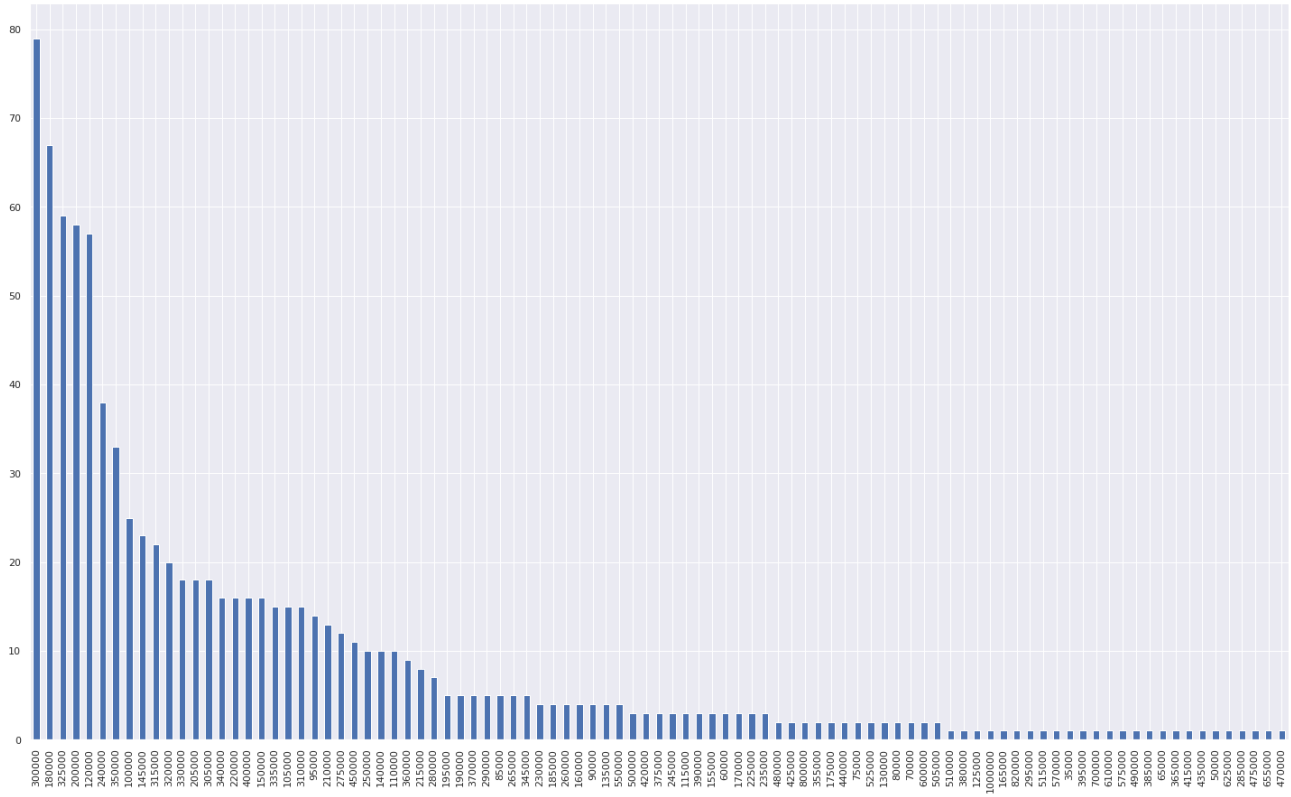
```
clean_data['JYEAR'].value_counts().plot(kind='bar', figsize=(7, 7))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed297aaeb8>
```

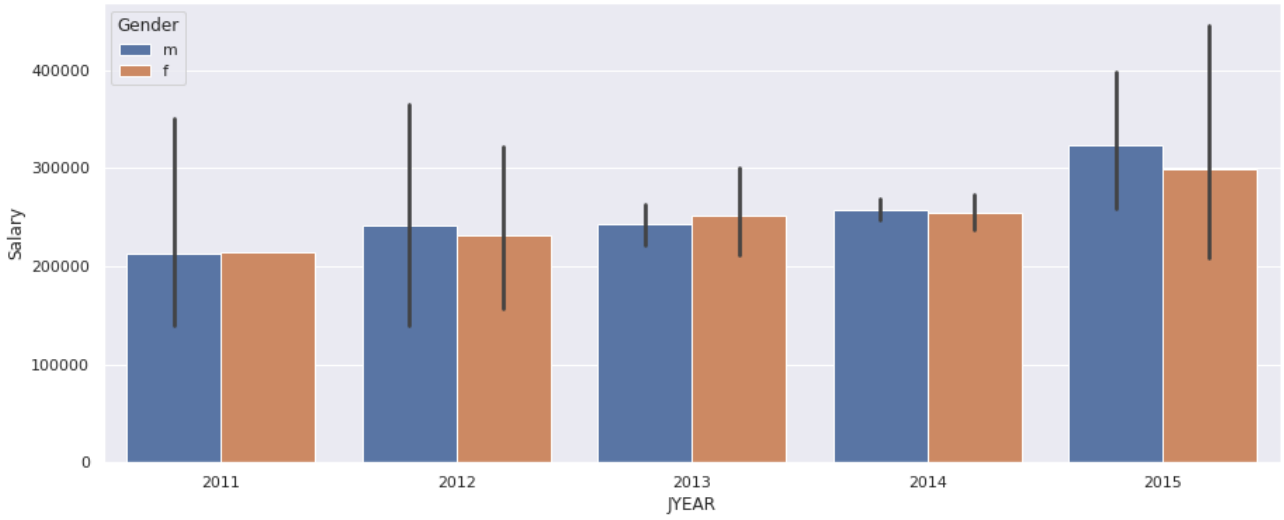



```
clean_data['Salary'].value_counts().plot(kind='bar', figsize=(25, 15))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29826240>
```

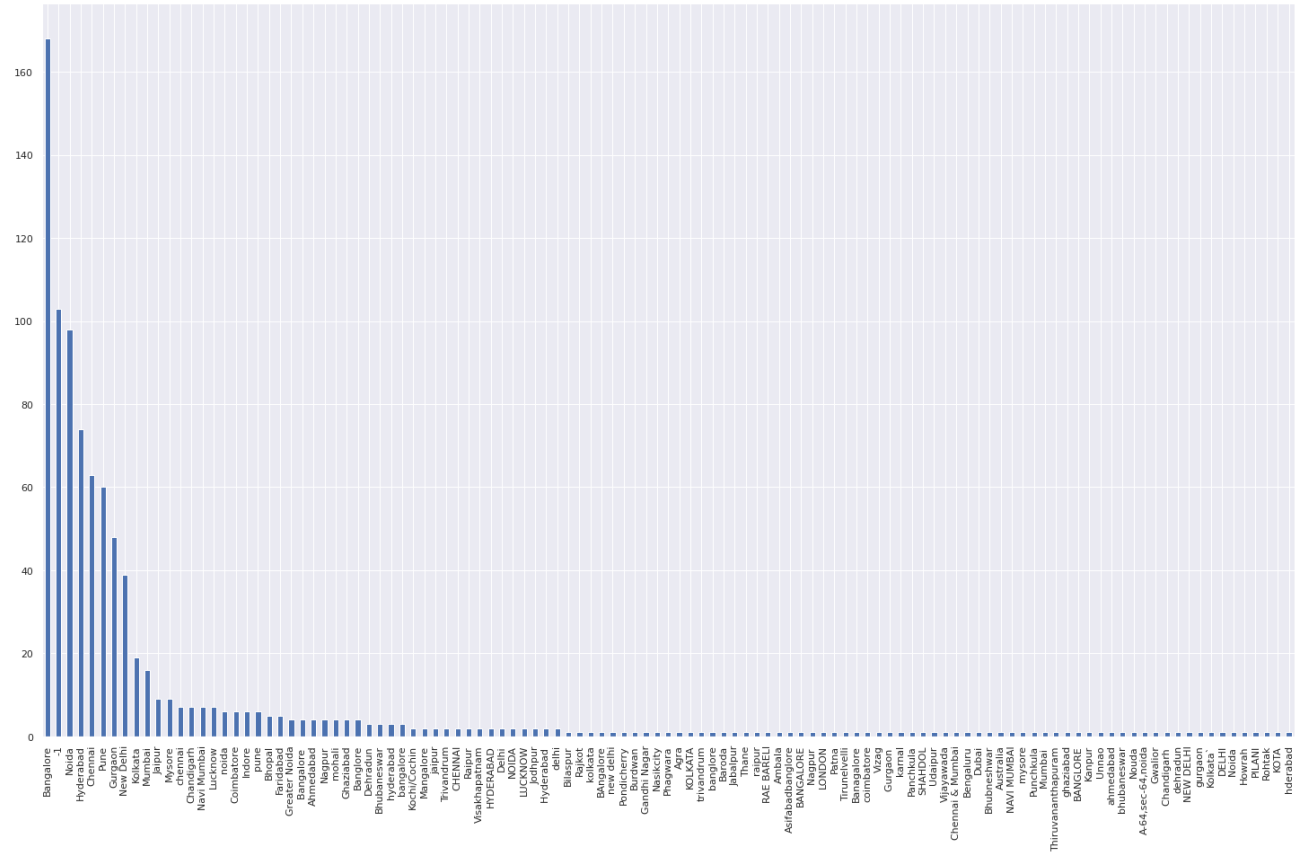


```
fig = plt.gcf();
fig.set_size_inches(15, 6);
ax=sns.barplot(x="JYEAR",y="Salary", data=clean_data, hue="Gender")
#ax.set(xlabel="CollegeState", ylabel = "Salary")
# ax.set_xticklabels(ax.get_xticklabels(), rotation=45);
```



```
clean_data['JobCity'].value_counts().plot(kind='bar', figsize=(25, 15))
```

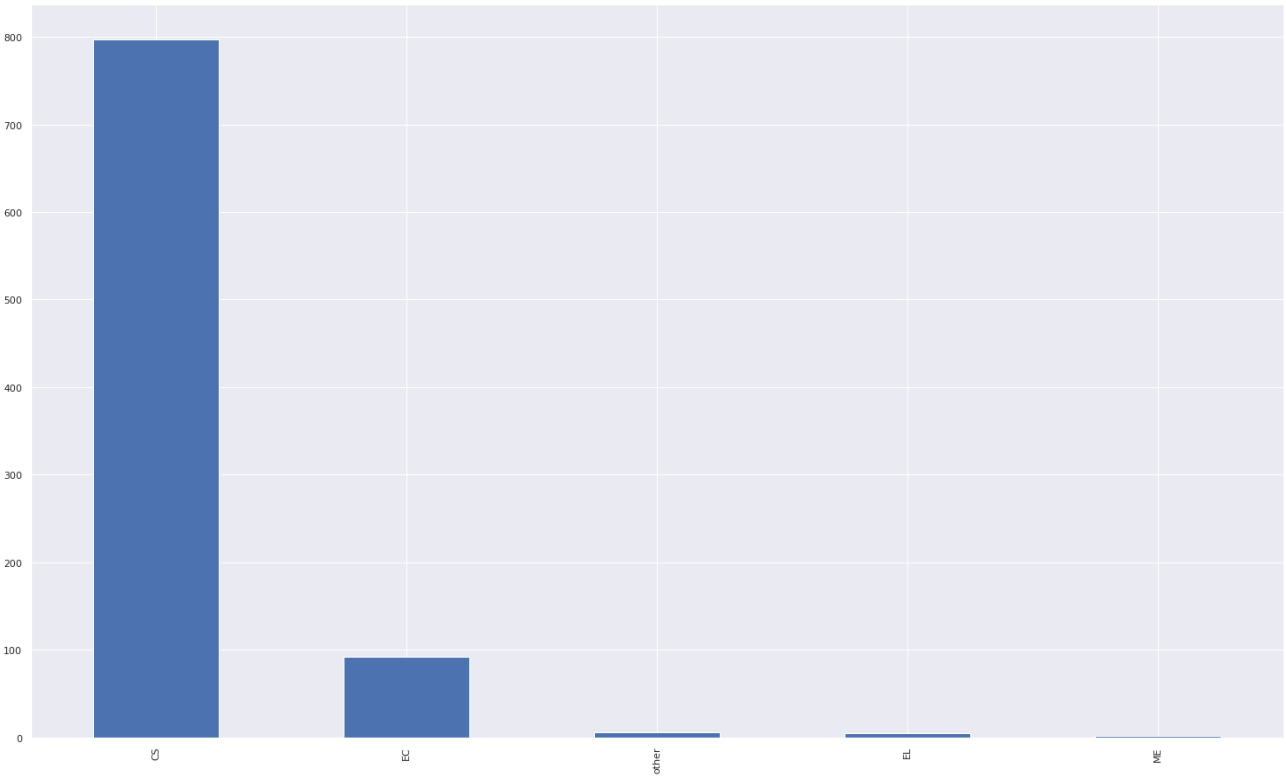
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed2a44dba8>
```




```
clean_data['Specialization'].value_counts().plot(kind='bar', figsize=(25, 15))
```

```
clean_data['Specialization'].value_counts().plot(kind='bar',figsize=(25, 15))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed2a4259b0>
```

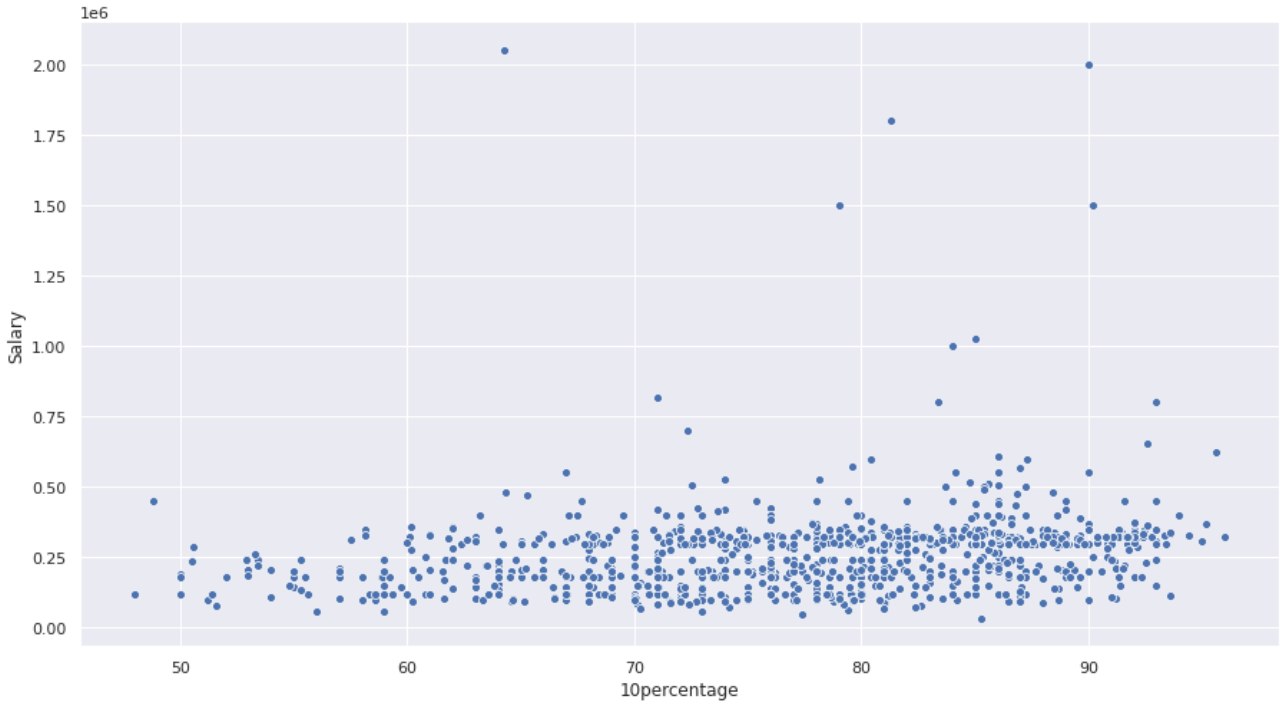


RELATIONSHIP BETWEEN THE FEILDS

```
fig = plt.gcf()
```

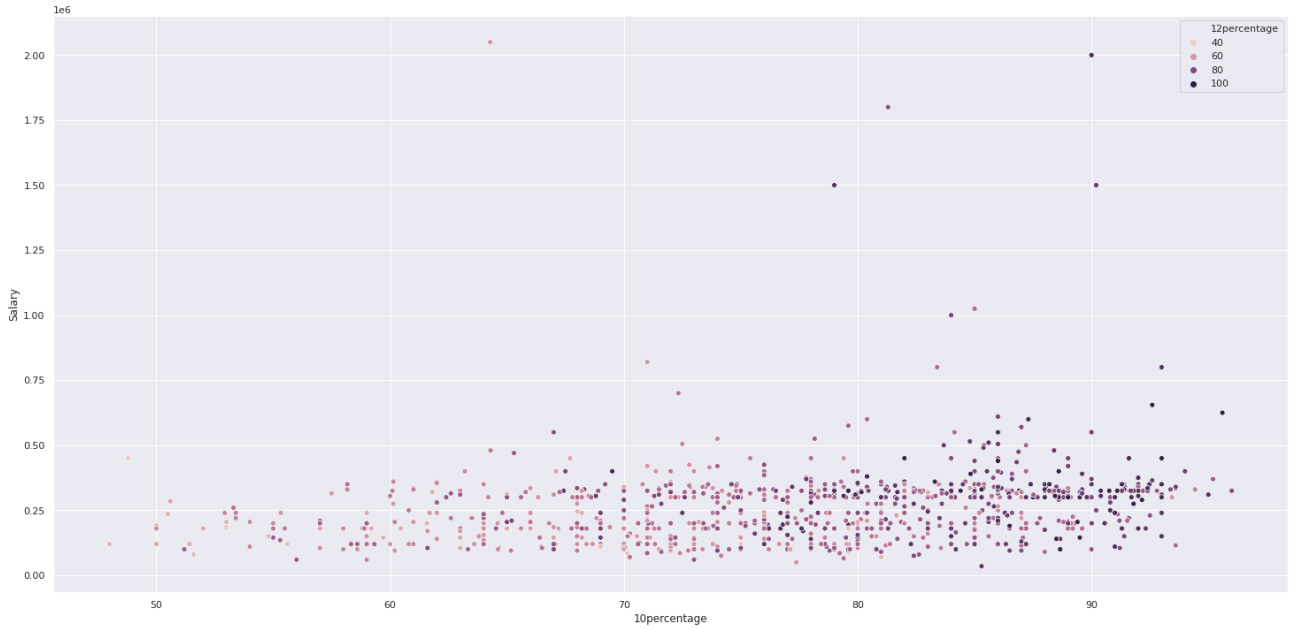
```
fig.set_size_inches(15, 8)
sns.scatterplot(x="10percentage", y="Salary", data=clean_data)
#10 percent vs salary
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed2d2906d8>
```



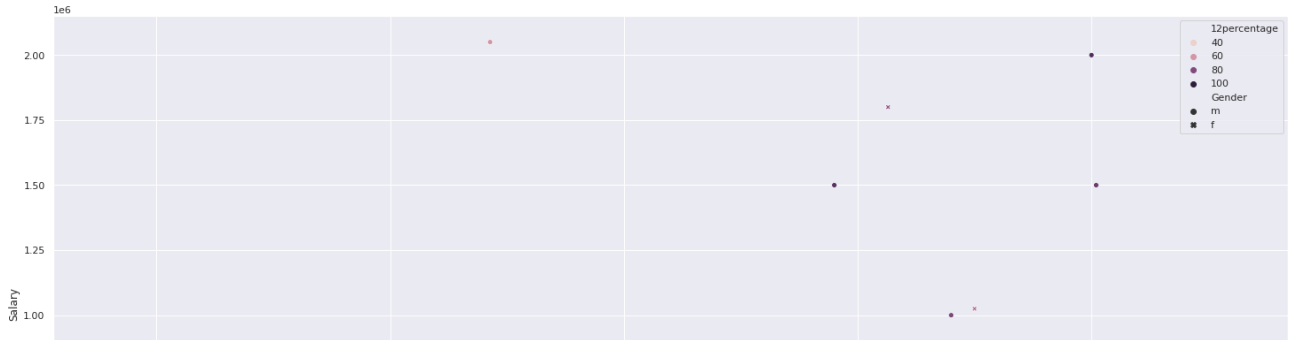

```
fig = plt.gcf()
fig.set_size_inches(25, 12)
sns.scatterplot(x="10percentage", y="Salary", data=clean_data, hue="12percentage")
#10, 12 percent vs salary
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29e16b38>
```



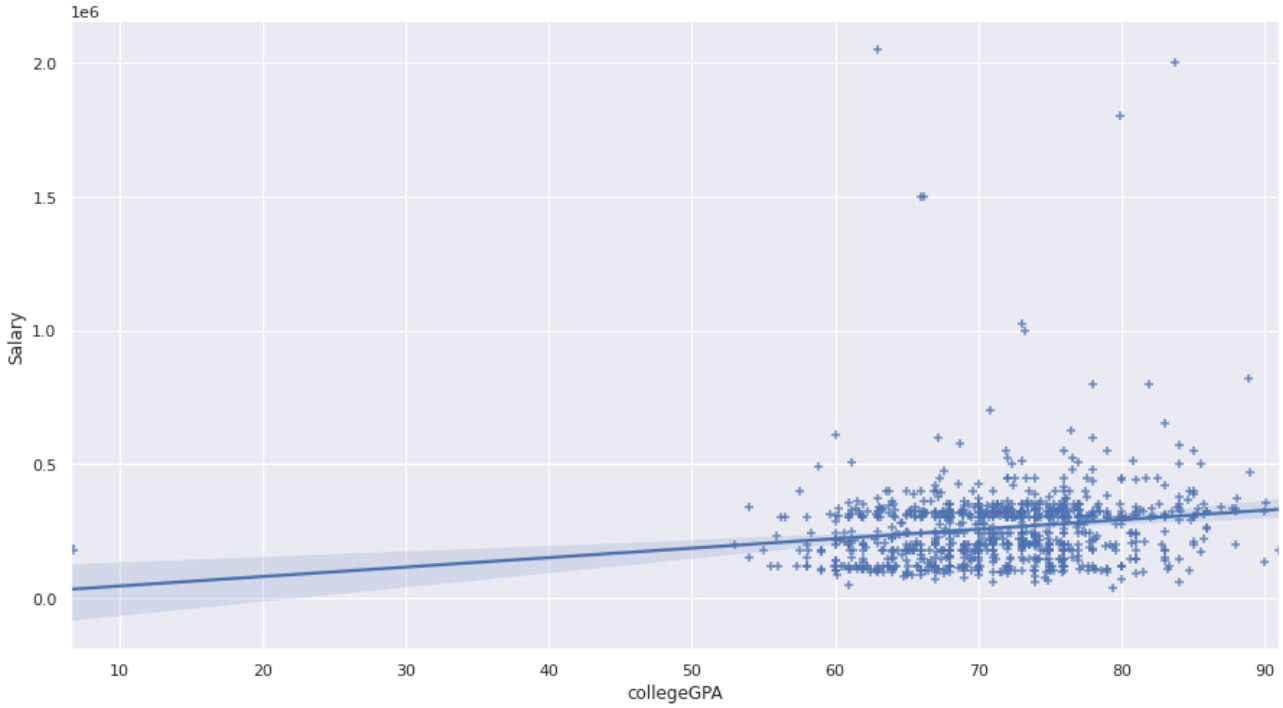

```
fig = plt.gcf()
fig.set_size_inches(25, 12)
sns.scatterplot(x="10percentage", y="Salary", data=clean_data, hue="12percentage", style
#10, 12 percent vs salary for male and female
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29dfcd68>
```



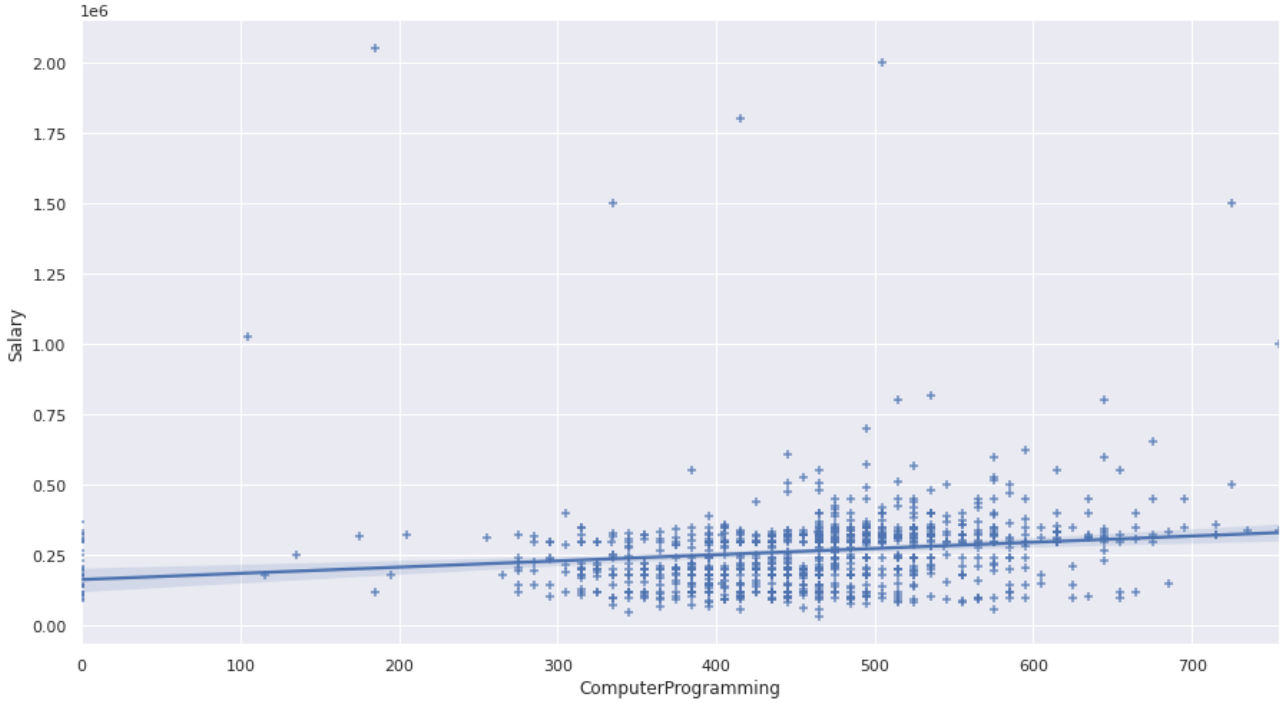

```
fig = plt.gcf()
fig.set_size_inches(15, 8)
sns.regplot(x="collegeGPA", y="Salary", data=clean_data, marker="+")
#college gpa vs salary
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29b224a8>
```




```
fig = plt.gcf()
fig.set_size_inches(15, 8)
sns.regplot(x="ComputerProgramming", y="Salary", data=clean_data, marker="+")
#college gpa vs salary
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29a74080>
```



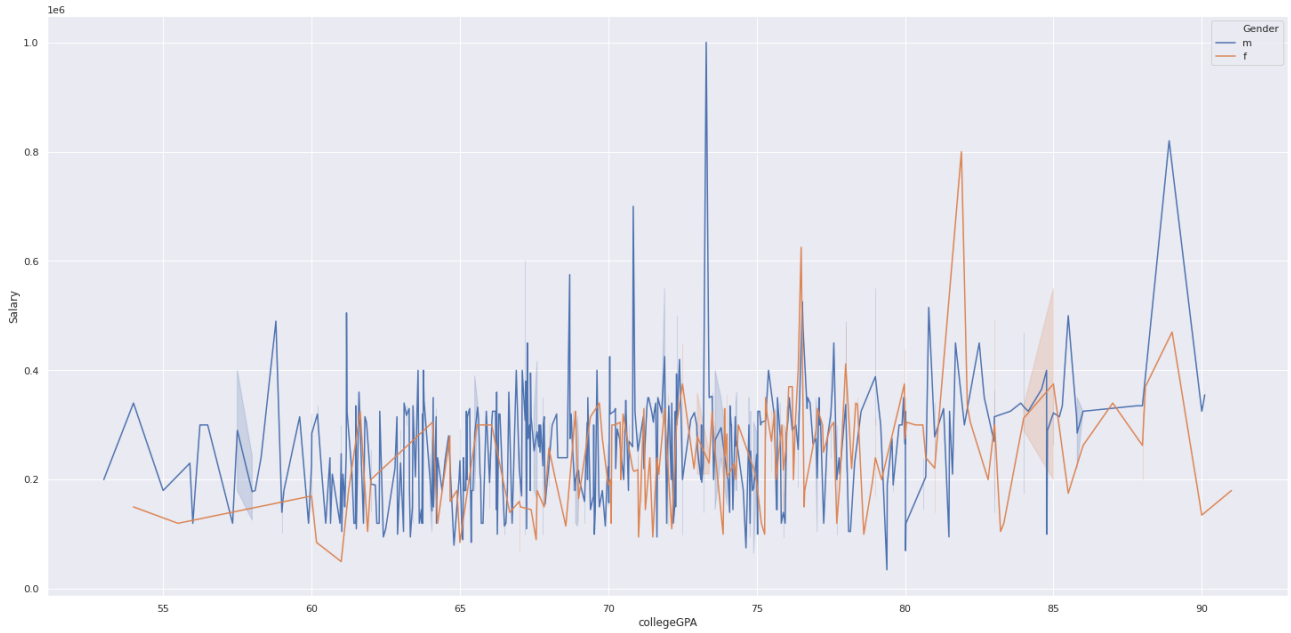
line plot relation between possiable factors

```
indexNames = clean_data[ clean_data['collegeGPA'] < 50 ].index

# Delete these row indexes from dataFrame
clean_data.drop(indexNames , inplace=True)
clean_data.shape


fig = plt.gcf()
fig.set_size_inches(25, 12)
sns.lineplot(x="collegeGPA", y="Salary", data=clean_data, hue="Gender")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29e82b00>
```

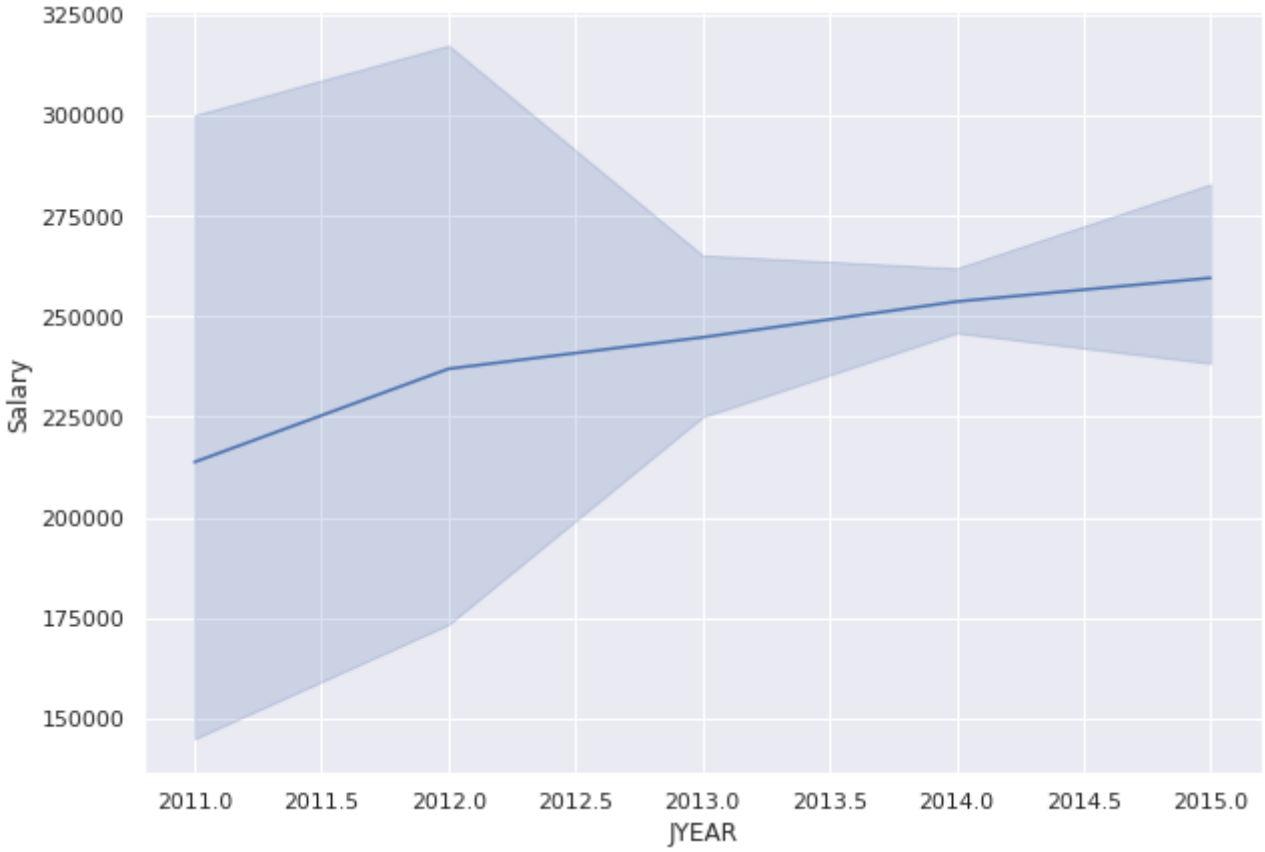



```
clean_data
```

```
CollegeCityID  GraduationYear  English  Logical  Quant    Domain  ComputerProgramming
```

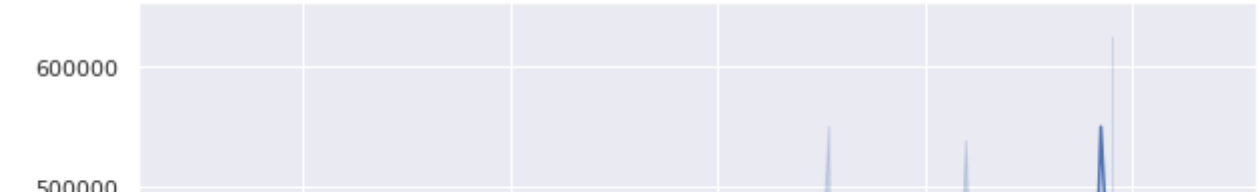
```
fig = plt.gcf()
fig.set_size_inches(10, 7)
sns.lineplot(x="JYEAR", y="Salary", data=clean_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed2ac5a5f8>
```




```
fig = plt.gcf()
fig.set_size_inches(10, 7)
sns.lineplot(x="Logical", y="Salary", data=clean_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29289ef0>
```



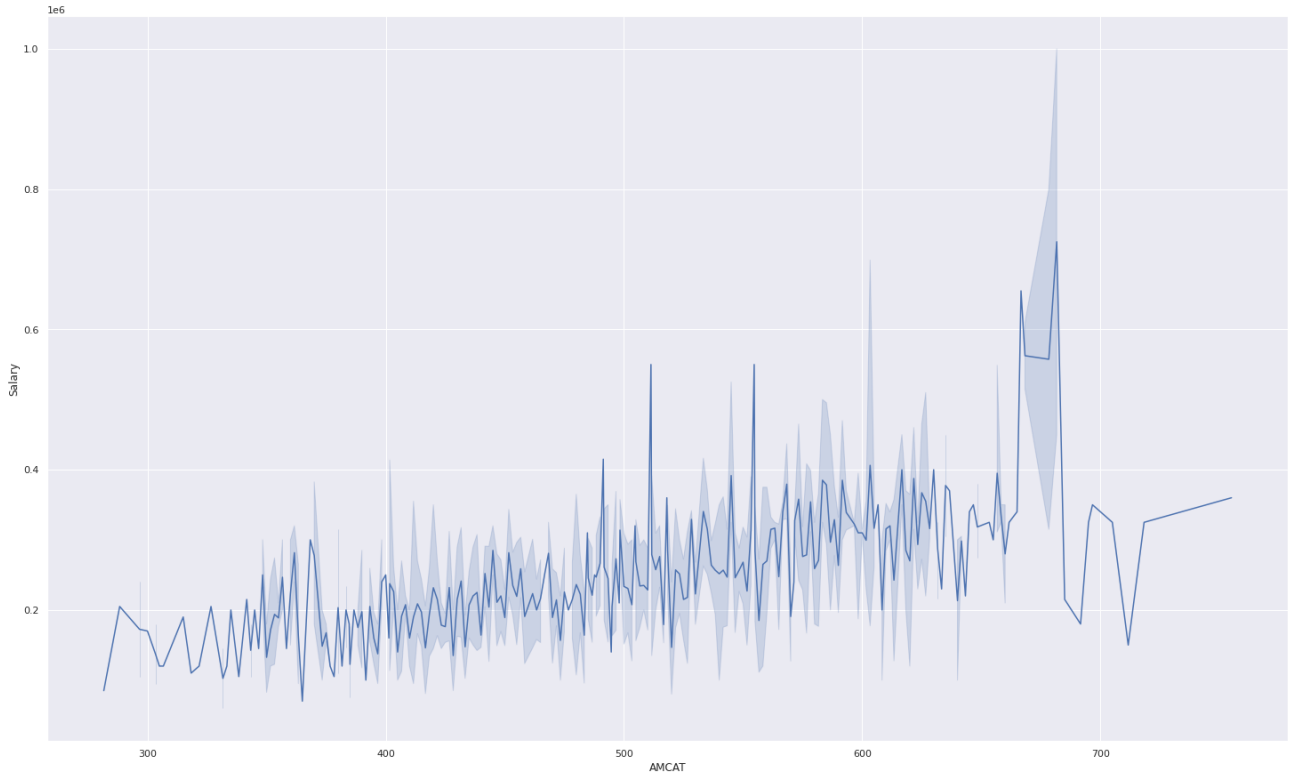

```
#creating a  column for average macat score and average academics score
clean_data['AMCAT']=(clean_data['Logical']+clean_data['Quant']+clean_data['English'])/3
clean_data['ACAD']=clean_data['10percentage']+clean_data['12percentage']+clean_data['col
```

```
clean_data
```

ish	Logical	Quant	Domain	ComputerProgramming	ComputerScience	conscientiousness
560	555	620	-1.000000	645	407	-0.3027
590	435	380	0.356536	405	346	1.7081
395	565	645	-1.000000	495	376	0.7027
465	585	515	0.911395	545	500	0.8463
525	555	630	0.356536	475	346	0.4155
...	...	...	...	...	...	..
550	445	475	0.649390	435	407	-1.1644
510	505	595	0.978799	455	561	0.4155
560	420	645	0.953900	575	530	0.1282
500	480	500	0.356536	465	346	0.1282
450	410	320	0.744758	445	438	-0.1590

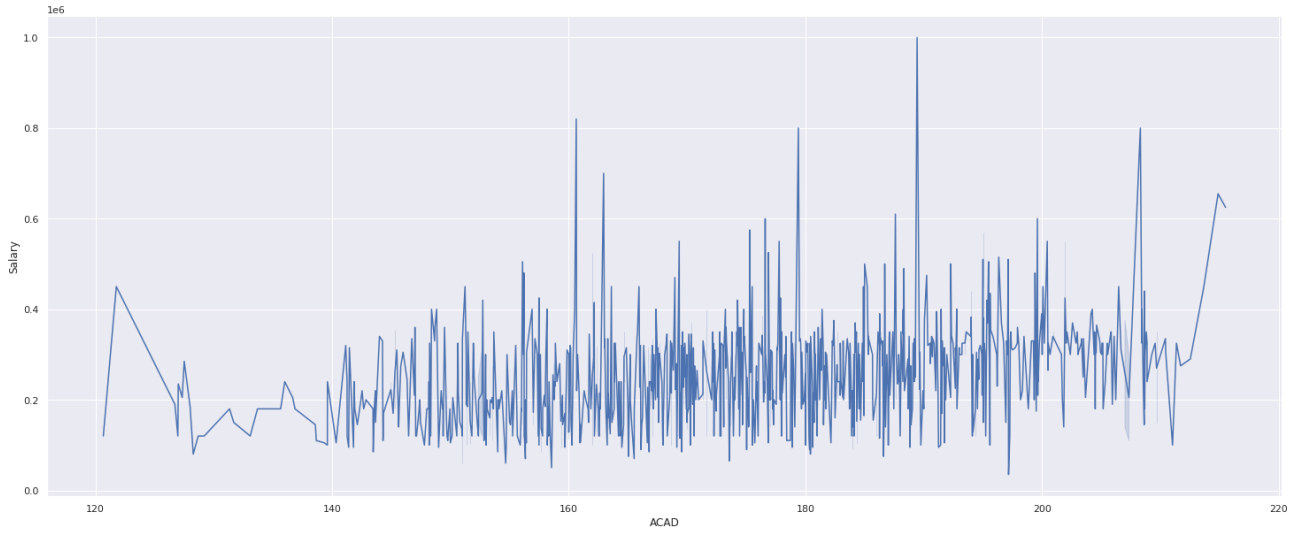
```
fig = plt.gcf()
fig.set_size_inches(25, 15)
sns.lineplot(x="AMCAT", y="Salary", data=clean_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed29192ac8>
```




```
fig = plt.gcf()
fig.set_size_inches(25, 10)
sns.lineplot(x="ACAD", y="Salary", data=clean_data)
```

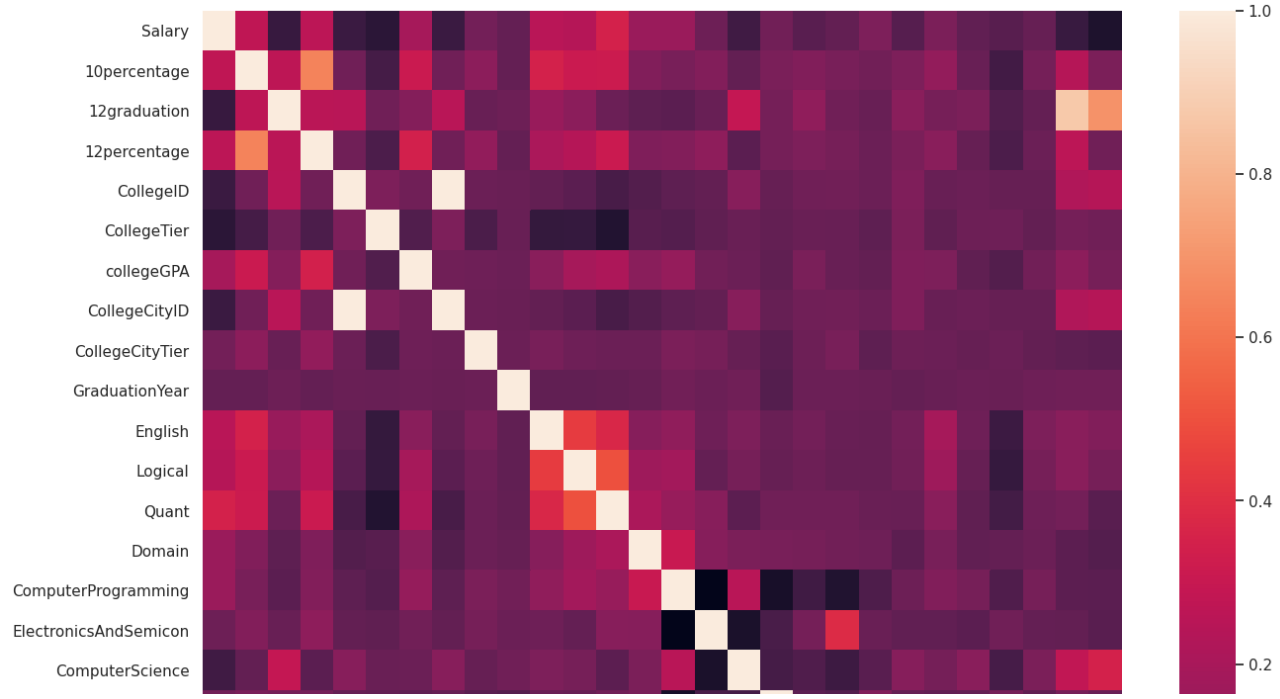
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed291c8518>
```




```
fig = plt.gcf()
fig.set_size_inches(15, 10)
sns.regplot(x="AMCAT", y="Salary", data=clean_data);
```


```
plt.figure(figsize= (15,15), dpi=100)
sns.heatmap(data.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed28d2e8d0>
```




```
fig = plt.gcf();
fig.set_size_inches(25, 15)
sns.heatmap(clean_data.corr(), annot=True);
```