In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('salary.csv')
```

In [3]:

```
data.head()
```

Out[3]:

| | timestamp | company | level | title | totalyearlycompensation | location | yearsofexperien |
|---|---|---|---|---|---|---|---|
| **0** | 06-07-2017 11:33 | Oracle | L3 | Product Manager | 127000 | Redwood City, CA | 1 |
| **1** | 06-10-2017 17:11 | eBay | SE 2 | Software Engineer | 100000 | San Francisco, CA | 5 |
| **2** | 06-11-2017 14:53 | Amazon | L7 | Product Manager | 310000 | Seattle, WA | 8 |
| **3** | 6/17/2017 0:23:14 | Apple | M1 | Software Engineering Manager | 372000 | Sunnyvale, CA | 7 |
| **4** | 6/20/2017 10:58:51 | Microsoft | 60 | Software Engineer | 157000 | Mountain View, CA | 5 |

5 rows × 29 columns

In [4]:

```
data.tail()
```

Out[4]:

| | timestamp | company | level | title | totalyearlycompensation | location | yearsofexperi |
|---|---|---|---|---|---|---|---|
| **62637** | 09-09-2018 11:52 | Google | T4 | Software Engineer | 327000 | Seattle, WA | |
| **62638** | 9/13/2018 8:23:32 | Microsoft | 62 | Software Engineer | 237000 | Redmond, WA | |
| **62639** | 9/13/2018 14:35:59 | MSFT | 63 | Software Engineer | 220000 | Seattle, WA | |
| **62640** | 9/16/2018 16:10:35 | Salesforce | Lead MTS | Software Engineer | 280000 | San Francisco, CA | |
| **62641** | 1/29/2019 5:12:59 | apple | ict3 | Software Engineer | 200000 | Sunnyvale, CA | |

5 rows × 29 columns

In [5]:

```
data.shape
```

Out[5]:

```
(62642, 29)
```

In [6]:

```
data.columns
```

Out[6]:

```
Index(['timestamp', 'company', 'level', 'title', 'totalyearlycompensation',
       'location', 'yearsofexperience', 'yearsatcompany', 'tag', 'basesalar
y',
       'stockgrantvalue', 'bonus', 'gender', 'otherdetails', 'cityid', 'dmai
d',
       'rowNumber', 'Masters_Degree', 'Bachelors_Degree', 'Doctorate_Degre
e',
       'Highschool', 'Some_College', 'Race_Asian', 'Race_White',
       'Race_Two_Or_More', 'Race_Black', 'Race_Hispanic', 'Race', 'Educatio
n'],
      dtype='object')
```

In [7]:

```
data.duplicated().sum()
```

Out[7]:

```
0
```

In [8]:

```
data.isnull().sum()
```

Out[8]:

```
timestamp                    0
company                      5
level                      119
title                        0
totalyearlycompensation      0
location                     0
yearsofexperience            0
yearsatcompany               0
tag                        854
basesalary                   0
stockgrantvalue              0
bonus                        0
gender                   19540
otherdetails             22505
cityid                       0
dmaid                        2
rowNumber                    0
Masters_Degree               0
Bachelors_Degree             0
Doctorate_Degree             0
Highschool                   0
Some_College                 0
Race_Asian                   0
Race_White                   0
Race_Two_Or_More             0
Race_Black                   0
Race_Hispanic                0
Race                     40215
Education                32272
dtype: int64
```

In [9]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62642 entries, 0 to 62641
Data columns (total 29 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   timestamp                62642 non-null  object 
 1   company                  62637 non-null  object 
 2   level                    62523 non-null  object 
 3   title                    62642 non-null  object 
 4   totalyearlycompensation  62642 non-null  int64  
 5   location                 62642 non-null  object 
 6   yearsofexperience        62642 non-null  float64
 7   yearsatcompany           62642 non-null  float64
 8   tag                      61788 non-null  object 
 9   basesalary               62642 non-null  int64  
 10  stockgrantvalue          62642 non-null  float64
 11  bonus                    62642 non-null  float64
 12  gender                   43102 non-null  object 
 13  otherdetails             40137 non-null  object 
 14  cityid                   62642 non-null  int64  
 15  dmaid                    62640 non-null  float64
 16  rowNumber                62642 non-null  int64  
 17  Masters_Degree           62642 non-null  int64  
 18  Bachelors_Degree         62642 non-null  int64  
 19  Doctorate_Degree         62642 non-null  int64  
 20  Highschool               62642 non-null  int64  
 21  Some_College             62642 non-null  int64  
 22  Race_Asian               62642 non-null  int64  
 23  Race_White               62642 non-null  int64  
 24  Race_Two_Or_More         62642 non-null  int64  
 25  Race_Black               62642 non-null  int64  
 26  Race_Hispanic            62642 non-null  int64  
 27  Race                     22427 non-null  object 
 28  Education                30370 non-null  object 
dtypes: float64(5), int64(14), object(10)
memory usage: 13.9+ MB
```

In [10]:

```
data.describe()
```

Out[10]:

| | totalyearlycompensation | yearsofexperience | yearsatcompany | basesalary | stockgrantval |
|---|---|---|---|---|---|
| **count** | 6.264200e+04 | 62642.000000 | 62642.000000 | 6.264200e+04 | 6.264200e+ |
| **mean** | 2.163004e+05 | 7.204135 | 2.702093 | 1.366873e+05 | 5.148608e+ |
| **std** | 1.380337e+05 | 5.840375 | 3.263656 | 6.136928e+04 | 8.187457e+ |
| **min** | 1.000000e+04 | 0.000000 | 0.000000 | 0.000000e+00 | 0.000000e+ |
| **25%** | 1.350000e+05 | 3.000000 | 0.000000 | 1.080000e+05 | 0.000000e+ |
| **50%** | 1.880000e+05 | 6.000000 | 2.000000 | 1.400000e+05 | 2.500000e+ |
| **75%** | 2.640000e+05 | 10.000000 | 4.000000 | 1.700000e+05 | 6.500000e+ |
| **max** | 4.980000e+06 | 69.000000 | 69.000000 | 1.659870e+06 | 2.800000e+ |

In [11]:

```
data.nunique()
```

Out[11]:

```
timestamp                 61755
company                    1631
level                      2916
title                        15
totalyearlycompensation      893
location                    1050
yearsofexperience             65
yearsatcompany                81
tag                         3058
basesalary                   482
stockgrantvalue              612
bonus                        335
gender                         4
otherdetails               12839
cityid                      1045
dmaid                        149
rowNumber                  62642
Masters_Degree                 2
Bachelors_Degree               2
Doctorate_Degree               2
Highschool                     2
Some_College                   2
Race_Asian                     2
Race_White                     2
Race_Two_Or_More               2
Race_Black                     2
Race_Hispanic                  2
Race                           5
Education                      5
dtype: int64
```

In [12]:

```
data_cat = data[['title', 'gender', 'Masters_Degree', 'Bachelors_Degree', 'Doctorate_Degree
                 'Highschool', 'Some_College', 'Race_Asian', 'Race_White','Race_Two_Or_More
                 'Race_Black', 'Race_Hispanic', 'Race', 'Education']]
```

In [13]:

```
for i in data_cat.columns:
    print(data_cat[i].unique())
```

```
['Product Manager' 'Software Engineer' 'Software Engineering Manager'
 'Data Scientist' 'Solution Architect' 'Technical Program Manager'
 'Human Resources' 'Product Designer' 'Marketing' 'Business Analyst'
 'Hardware Engineer' 'Sales' 'Recruiter' 'Mechanical Engineer'
 'Management Consultant']
[nan 'Male' 'Female' 'Other' 'Title: Senior Software Engineer']
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[0 1]
[nan 'White' 'Asian' 'Black' 'Two Or More' 'Hispanic']
[nan 'PhD' "Master's Degree" "Bachelor's Degree" 'Some College'
 'Highschool']
```

In [14]:

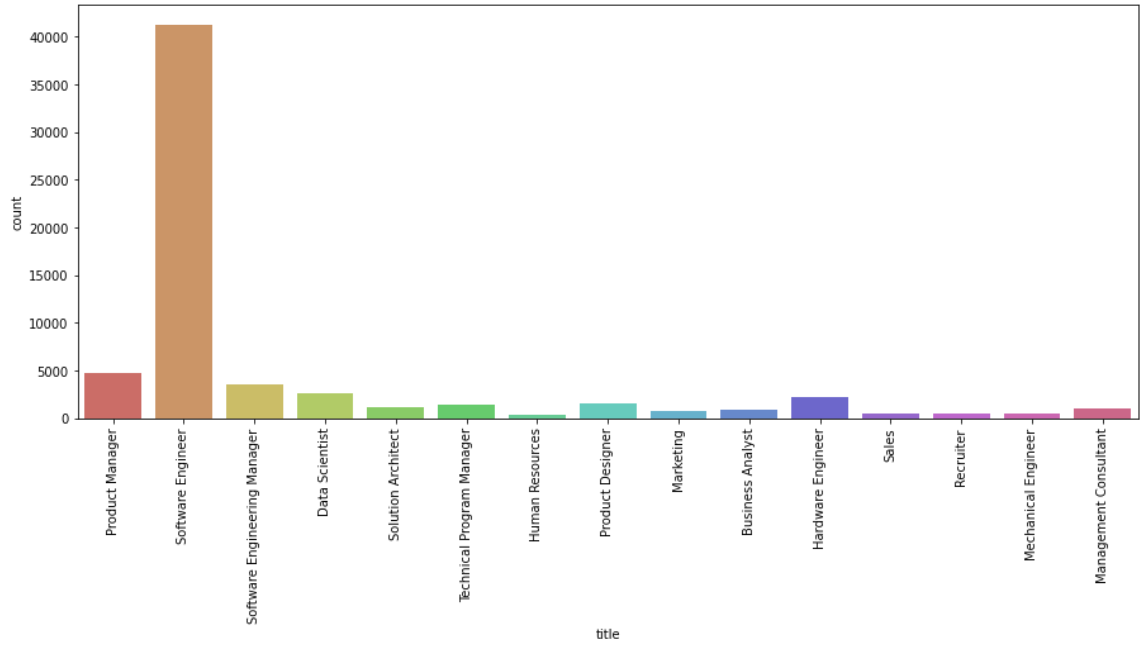
```
for i in data_cat.columns:
    print(data_cat[i].value_counts())
```

```
Software Engineer               41231
Product Manager                  4673
Software Engineering Manager     3569
Data Scientist                   2578
Hardware Engineer                2200
Product Designer                 1516
Technical Program Manager        1381
Solution Architect               1157
Management Consultant             976
Business Analyst                  885
Marketing                         710
Mechanical Engineer               490
Sales                             461
Recruiter                         451
Human Resources                   364
Name: title, dtype: int64
Male                               35702
Female                              6999
Other                                400
Title: Senior Software Engineer        1
Name: gender, dtype: int64
0    47251
1    15391
Name: Masters_Degree, dtype: int64
0    50037
1    12605
Name: Bachelors_Degree, dtype: int64
0    60839
1     1803
Name: Doctorate_Degree, dtype: int64
0    62322
1      320
Name: Highschool, dtype: int64
0    62287
1      355
Name: Some_College, dtype: int64
0    50870
1    11772
Name: Race_Asian, dtype: int64
0    54610
1     8032
Name: Race_White, dtype: int64
0    61838
1      804
Name: Race_Two_Or_More, dtype: int64
0    61952
1      690
Name: Race_Black, dtype: int64
0    61512
1     1130
Name: Race_Hispanic, dtype: int64
Asian          11772
White           8032
Hispanic        1129
Two Or More      804
```

```
Black            690
Name: Race, dtype: int64
Master's Degree      15391
Bachelor's Degree    12601
PhD                   1703
Some College           355
Highschool             320
Name: Education, dtype: int64
```

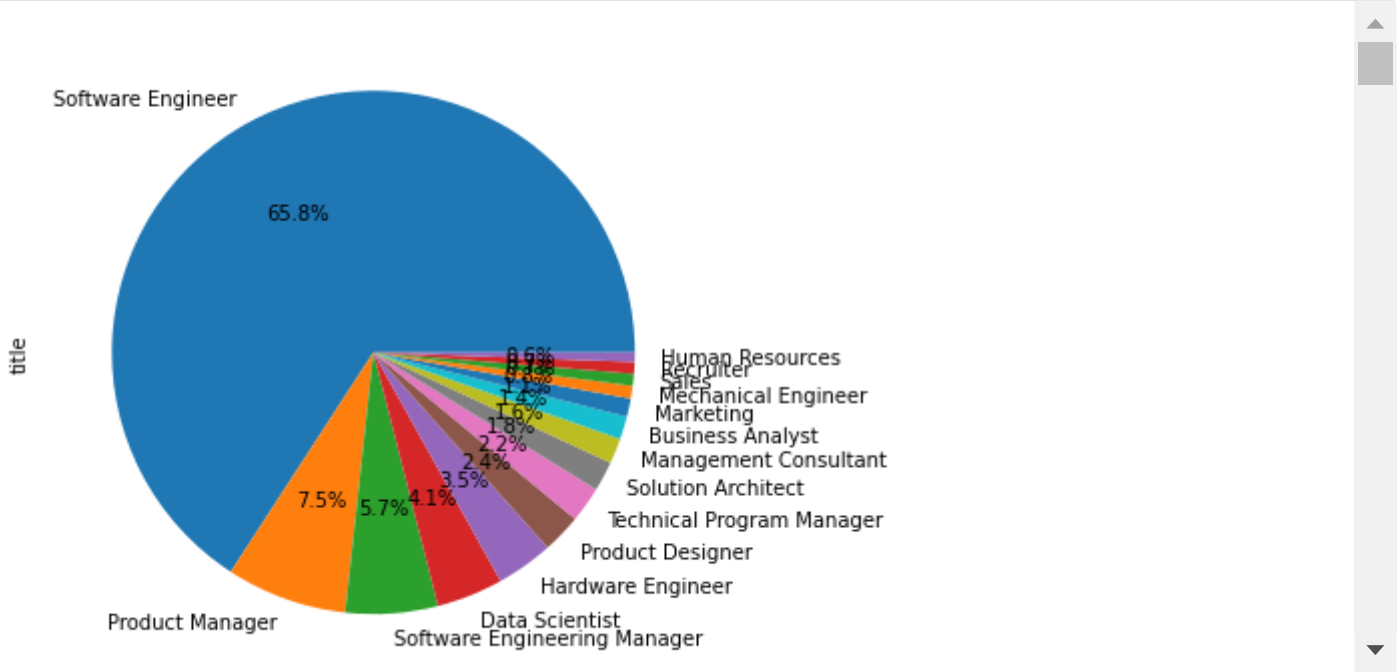
In [15]:

```
for i in data_cat.columns:
    plt.figure(figsize = (15,6))
    sns.countplot(data_cat[i], data = data_cat, palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()
```



In [16]:

```
for i in data_cat.columns:
    plt.figure(figsize = (15,6))
    data_cat[i].value_counts().plot(kind = 'pie', autopct = '%1.1f%%')
    plt.xticks(rotation = 90)
    plt.show()
```




In [17]:

```
data_salaries = data[['company', 'title', 'totalyearlycompensation','location',
                      'yearsofexperience', 'yearsatcompany', 'gender','Race', 'Education']]
```

In [18]:

```
data_salaries = pd.DataFrame(data_salaries)
```

In [19]:

```
data_salaries.head()
```

Out[19]:

| | company | title | totalyearlycompensation | location | yearsofexperience | yearsatcompan |
|---|---|---|---|---|---|---|
| **0** | Oracle | Product Manager | 127000 | Redwood City, CA | 1.5 | 1. |
| **1** | eBay | Software Engineer | 100000 | San Francisco, CA | 5.0 | 3. |
| **2** | Amazon | Product Manager | 310000 | Seattle, WA | 8.0 | 0. |
| **3** | Apple | Software Engineering Manager | 372000 | Sunnyvale, CA | 7.0 | 5. |
| **4** | Microsoft | Software Engineer | 157000 | Mountain View, CA | 5.0 | 3. |

In [20]:

```
data_salaries.tail()
```

Out[20]:

| | company | title | totalyearlycompensation | location | yearsofexperience | yearsatcomp |
|---|---|---|---|---|---|---|
| **62637** | Google | Software Engineer | 327000 | Seattle, WA | 10.0 | |
| **62638** | Microsoft | Software Engineer | 237000 | Redmond, WA | 2.0 | |
| **62639** | MSFT | Software Engineer | 220000 | Seattle, WA | 14.0 | |
| **62640** | Salesforce | Software Engineer | 280000 | San Francisco, CA | 8.0 | |
| **62641** | apple | Software Engineer | 200000 | Sunnyvale, CA | 0.0 | |

In [21]:

```
data_salaries['title'].value_counts()
```

Out[21]:

```
Software Engineer               41231
Product Manager                  4673
Software Engineering Manager     3569
Data Scientist                   2578
Hardware Engineer                2200
Product Designer                 1516
Technical Program Manager        1381
Solution Architect               1157
Management Consultant             976
Business Analyst                  885
Marketing                         710
Mechanical Engineer               490
Sales                             461
Recruiter                         451
Human Resources                   364
Name: title, dtype: int64
```

In [22]:

```
undesired_titles = ['Marketing', 'Mechanical Engineer', 'Sales', 'Recruiter', 'Human Resour
data_salaries_original = data_salaries.copy()
data_salaries = data_salaries[data_salaries['title'].apply(lambda x: x not in undesired_tit
data_salaries['title'].value_counts()
```

Out[22]:

```
Software Engineer               41231
Product Manager                  4673
Software Engineering Manager     3569
Data Scientist                   2578
Hardware Engineer                2200
Product Designer                 1516
Technical Program Manager        1381
Solution Architect               1157
Management Consultant             976
Business Analyst                  885
Name: title, dtype: int64
```

In [23]:

```
data_salaries.shape
```

Out[23]:

```
(60166, 9)
```

In [24]:

```
data_salaries.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 60166 entries, 0 to 62641
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   company                  60161 non-null  object 
 1   title                    60166 non-null  object 
 2   totalyearlycompensation  60166 non-null  int64  
 3   location                 60166 non-null  object 
 4   yearsofexperience        60166 non-null  float64
 5   yearsatcompany           60166 non-null  float64
 6   gender                   41614 non-null  object 
 7   Race                     21160 non-null  object 
 8   Education                28922 non-null  object 
dtypes: float64(2), int64(1), object(6)
memory usage: 4.6+ MB
```

In [25]:

```
data_salaries.isnull().sum()
```

Out[25]:

```
company                        5
title                          0
totalyearlycompensation        0
location                       0
yearsofexperience              0
yearsatcompany                 0
gender                     18552
Race                       39006
Education                  31244
dtype: int64
```

In [26]:

```
null_data = pd.DataFrame(data_salaries.isnull().sum(), columns = ['Count of Nulls'])
null_data.index.name = 'Column Name'
null_data[null_data ['Count of Nulls'] > 0].sort_values('Count of Nulls', ascending=False)
```

Out[26]:

| Column Name | Count of Nulls |
|---|---|
| **Race** | 39006 |
| **Education** | 31244 |
| **gender** | 18552 |
| **company** | 5 |

In [27]:

```
data_salaries.fillna({'company':'NA', 'gender':'NA','Race': 'NA', 'Education': 'NA'}, inpla
```

In [28]:

```
data_salaries.isnull().sum()
```

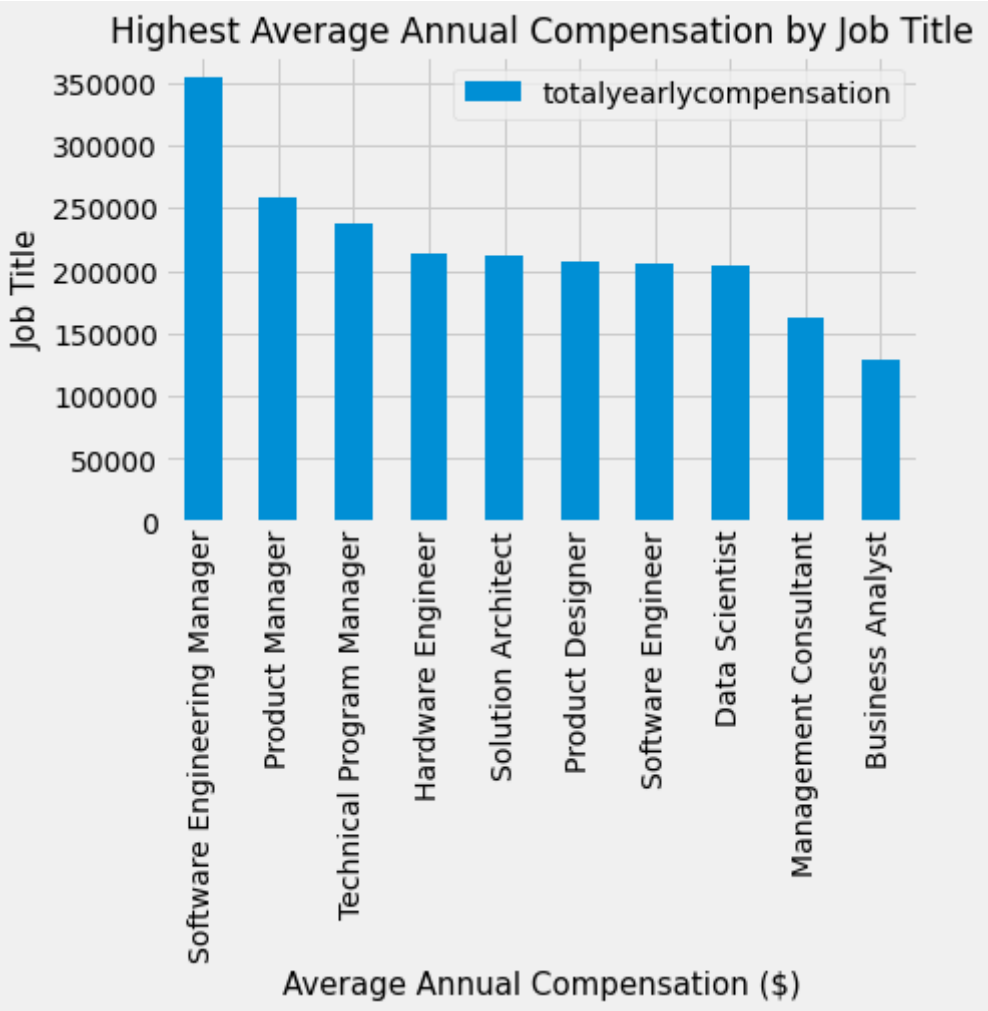
Out[28]:

```
company                  0
title                    0
totalyearlycompensation  0
location                 0
yearsofexperience        0
yearsatcompany           0
gender                   0
Race                     0
Education                0
dtype: int64
```

In [29]:

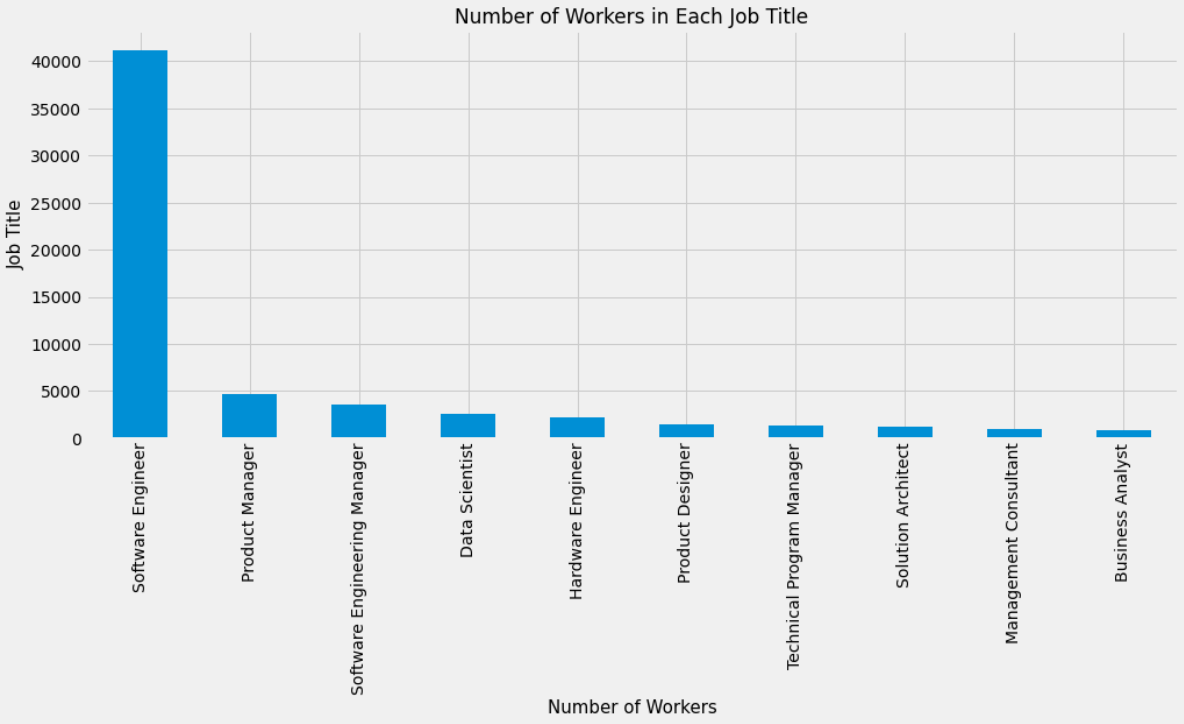
```
job_titles = data_salaries[['company','title', 'totalyearlycompensation']].groupby(['title'
                                                                                    .mean()\
                                                                                    .round(2)\
                                                                                    .sort_values('tot

plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
job_titles.plot.bar()
plt.title('Highest Average Annual Compensation by Job Title', size=17)
plt.xlabel('Average Annual Compensation ($)', size = 15)
plt.ylabel('Job Title', size = 15)
plt.show()
```

```
<Figure size 1080x432 with 0 Axes>
```



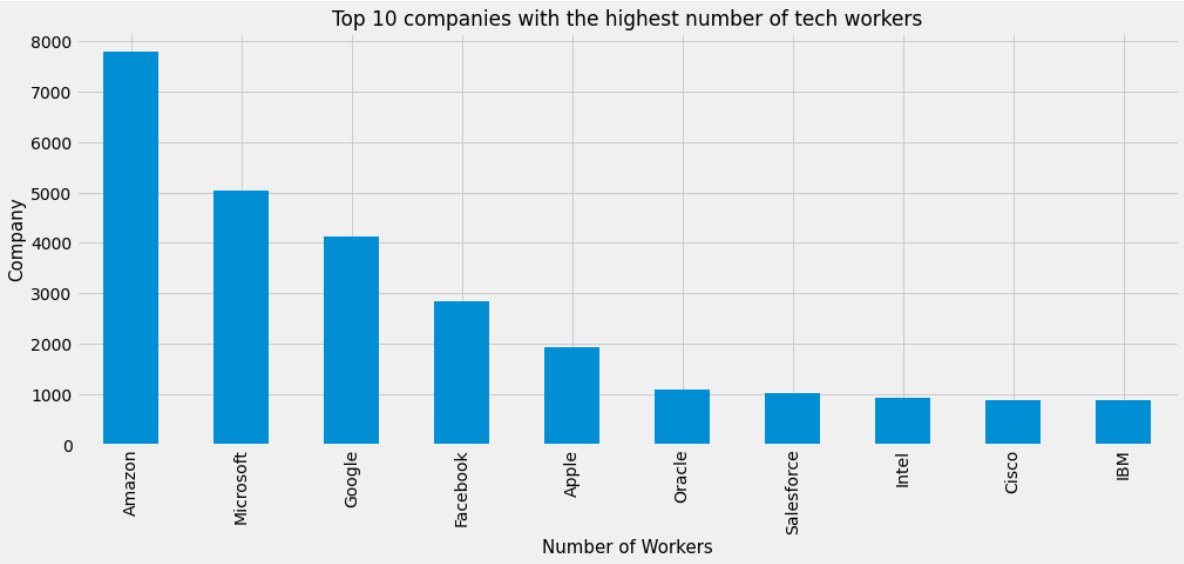
In [30]:

```
top_jobs = data_salaries['title'].value_counts()
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
top_jobs.plot.bar()
plt.title("Number of Workers in Each Job Title", size=17)
plt.xlabel('Number of Workers', size = 15)
plt.ylabel('Job Title', size = 15)
plt.show();
```



In [31]:

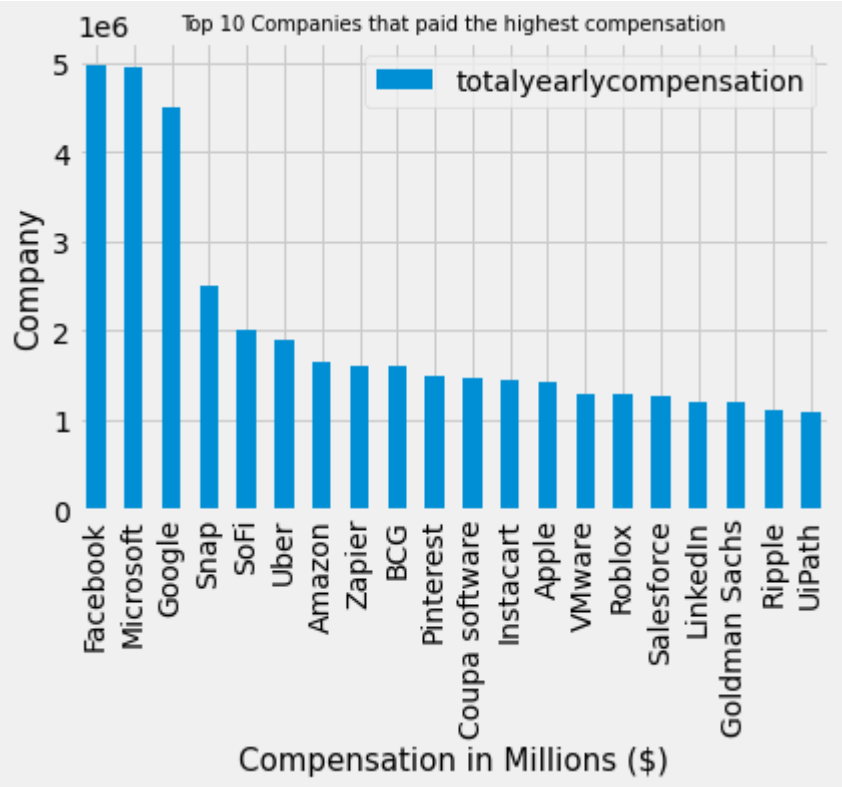
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
companies_with_most_tech_workers = data_salaries['company'].value_counts()[:10].plot.bar();
plt.title('Top 10 companies with the highest number of tech workers', size=17)
plt.xlabel('Number of Workers', size = 15)
plt.ylabel('Company', size = 15)
plt.show();
```



In [32]:

```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
highest_paying_companies = data_salaries[['company','title','totalyearlycompensation']].gro
                                                                                        .max
                                                                                        .hea
plt.title('Top 10 Companies that paid the highest compensation', size=10)
plt.xlabel('Compensation in Millions ($)', size = 15)
plt.ylabel('Company', size = 15)
plt.show();
```

```
<Figure size 1080x432 with 0 Axes>
```



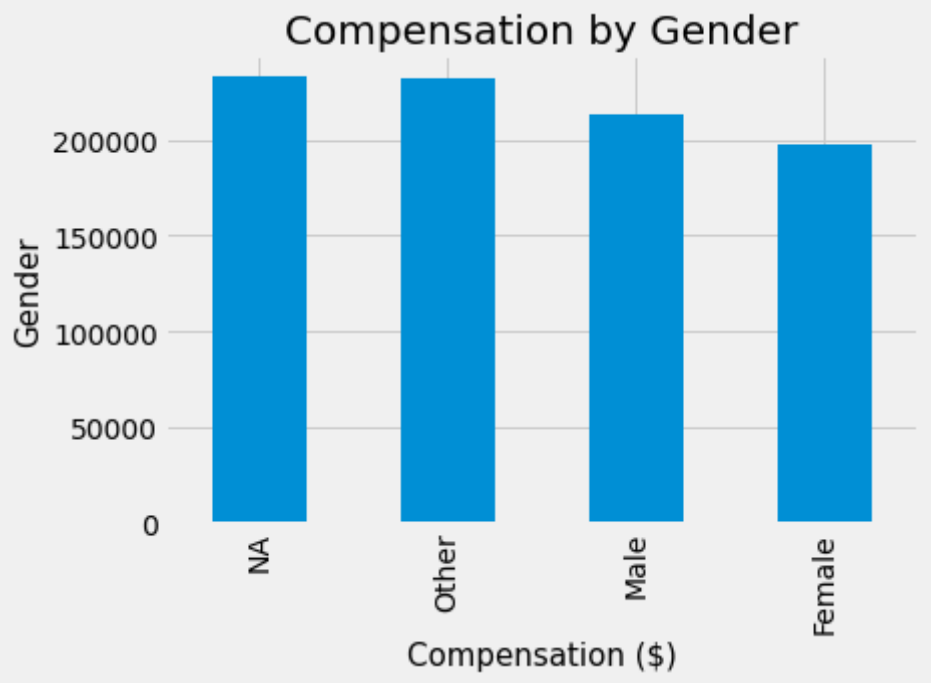
In [33]:

```
data_salaries['gender'].replace('Title: Senior Software Engineer', 'NA', inplace = True)
```

In [34]:

```
pay_by_gender = data_salaries[['totalyearlycompensation', 'gender']].groupby(['gender']).me
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
pay_by_gender.sort_values('totalyearlycompensation', ascending = False).head(10).plot.bar(l
plt.title('Compensation by Gender', size=20)
plt.xlabel('Compensation ($)', size = 15)
plt.ylabel('Gender', size = 15)
plt.show();
```

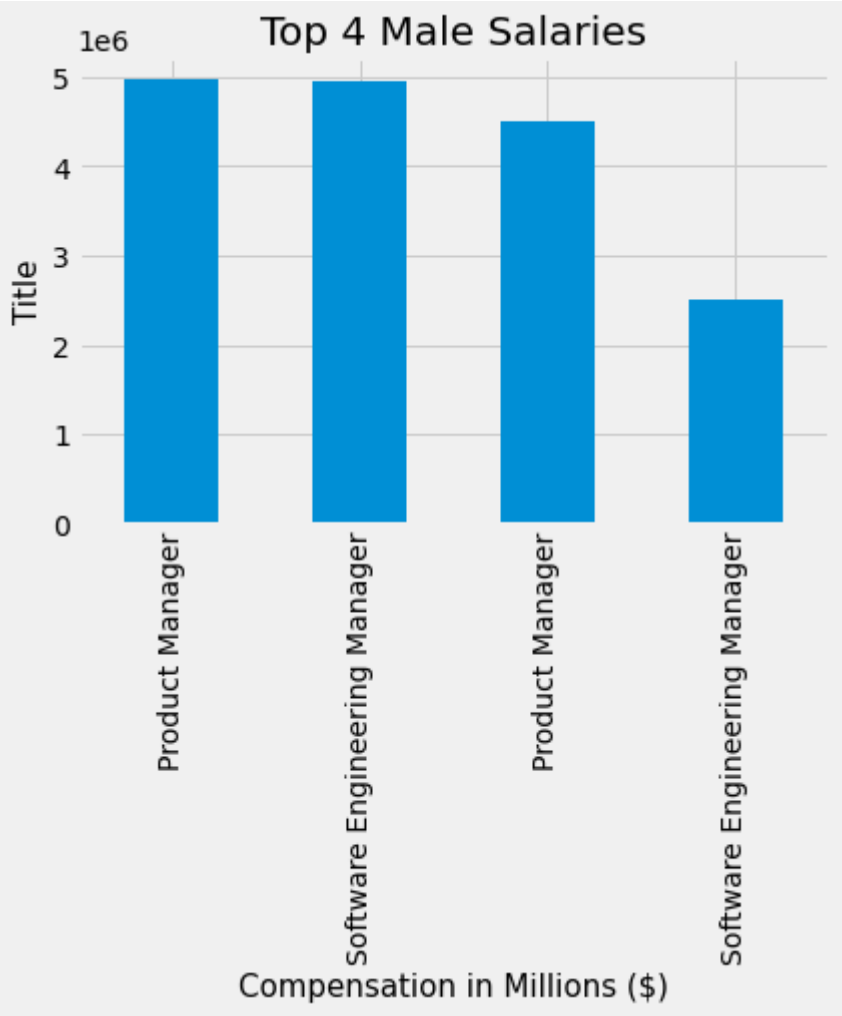
```
<Figure size 1080x432 with 0 Axes>
```



In [35]:

```
male_salaries = data_salaries[data_salaries.gender == 'Male'].copy()
top4_male_salaries = male_salaries.nlargest(4,'totalyearlycompensation')
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
top4_male_salaries.plot.bar(x = 'title', y = 'totalyearlycompensation', legend = False);
plt.title('Top 4 Male Salaries', size=20)
plt.xlabel('Compensation in Millions ($)', size = 15)
plt.ylabel('Title', size = 15)
plt.show();
```

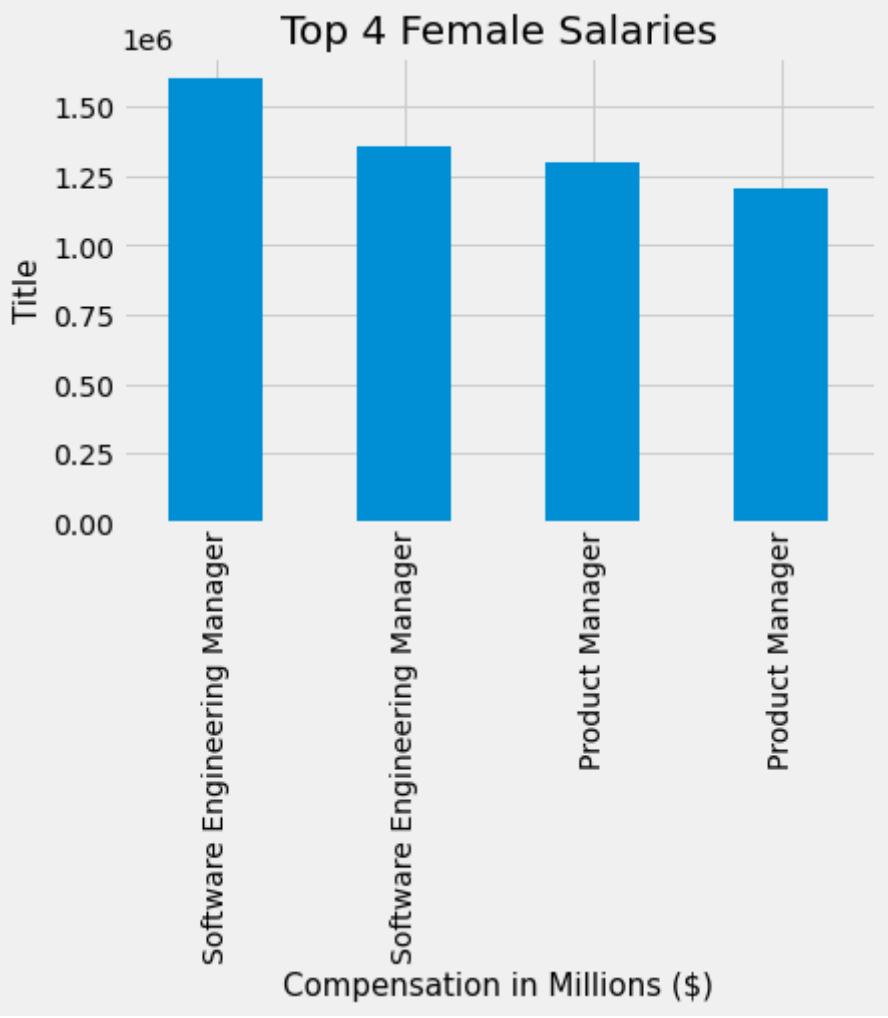
```
<Figure size 1080x432 with 0 Axes>
```



In [36]:

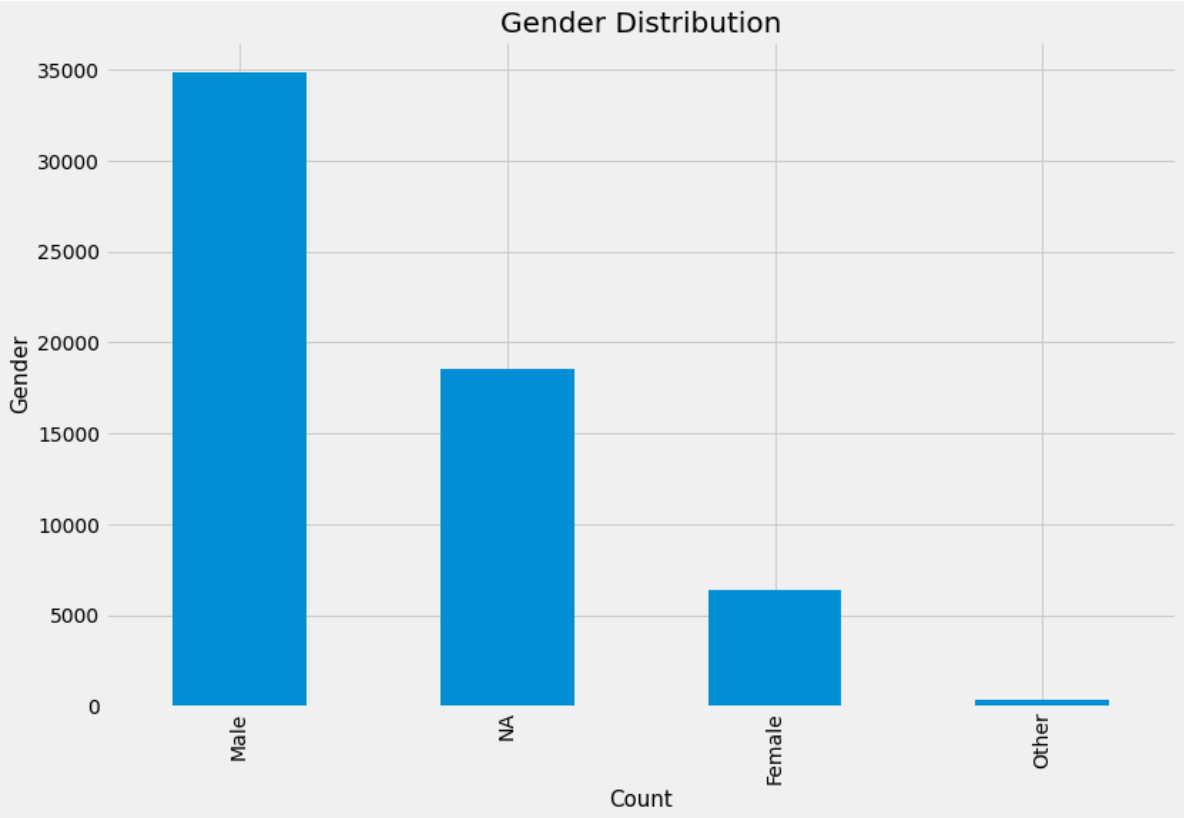
```
female_salaries = data_salaries[data_salaries.gender == 'Female'].copy()
top4_female_salaries = female_salaries.nlargest(4,'totalyearlycompensation')
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
top4_female_salaries.plot.bar(x = 'title', y = 'totalyearlycompensation', legend = False);
plt.title('Top 4 Female Salaries', size=20)
plt.xlabel('Compensation in Millions ($)', size = 15)
plt.ylabel('Title', size = 15)
plt.show();
```

```
<Figure size 1080x432 with 0 Axes>
```



In [37]:

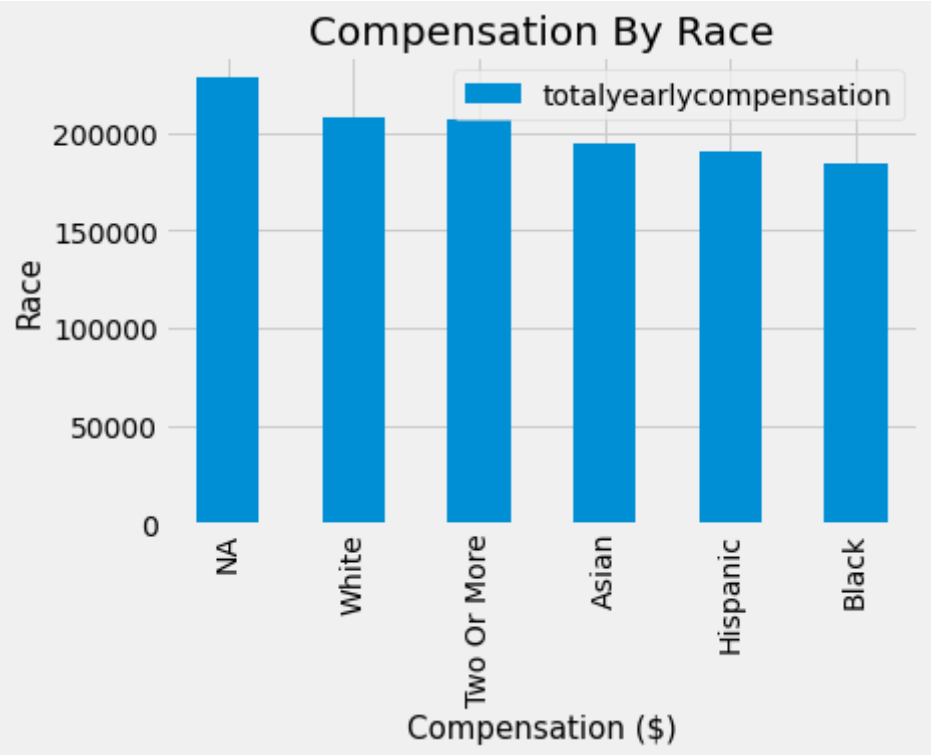
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
gender_distribution = data_salaries['gender'].value_counts().plot.bar(figsize = (12,8));
plt.title('Gender Distribution', size=20)
plt.xlabel('Count', size = 15)
plt.ylabel('Gender', size = 15)
plt.show();
```



In [38]:

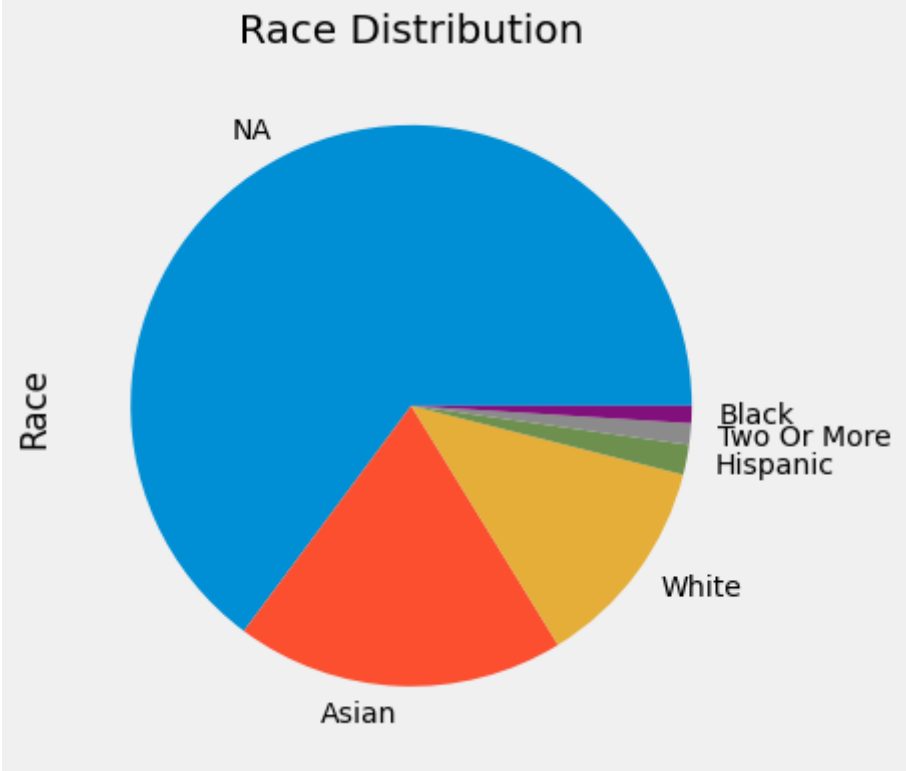
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
pay_by_race = data_salaries[['totalyearlycompensation','Race']].groupby(['Race'])\
                                                               .mean()\
                                                               .round(2)\
                                                               .sort_values('totalyearlycom
                                                               .plot.bar()
plt.title('Compensation By Race', size=20)
plt.xlabel('Compensation ($)', size = 15)
plt.ylabel('Race', size = 15)
plt.show();
```

<Figure size 1080x432 with 0 Axes>



In [39]:

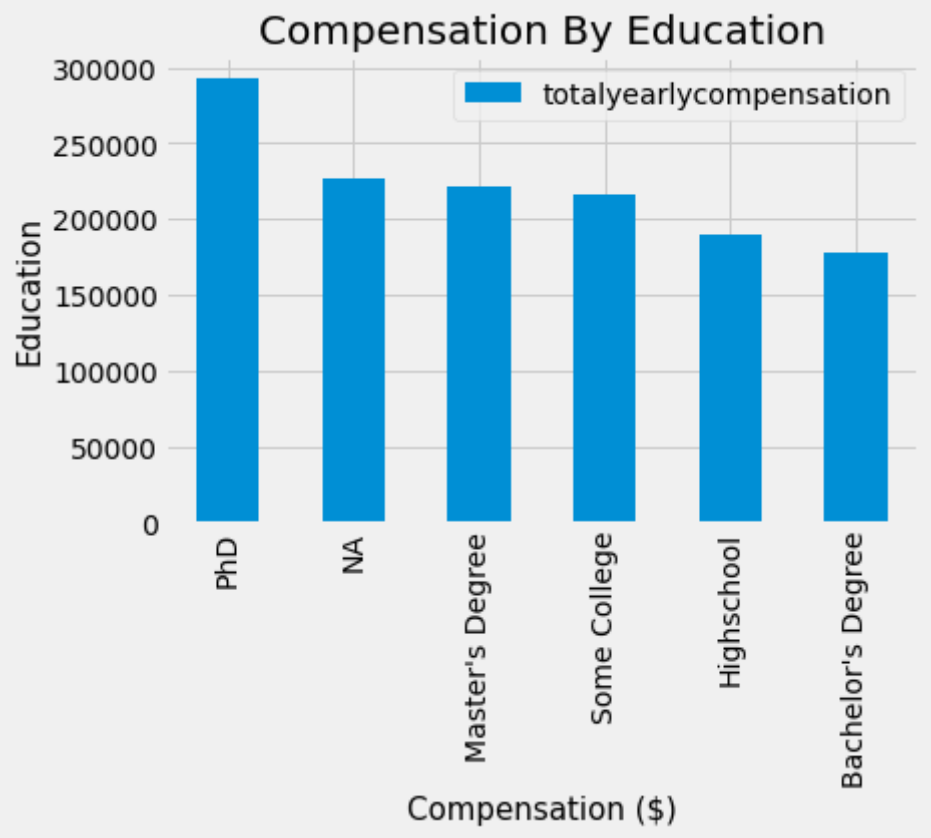
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
race_distribution = data_salaries['Race'].value_counts().plot.pie();
plt.title('Race Distribution', size=20)
plt.show();
```



In [40]:

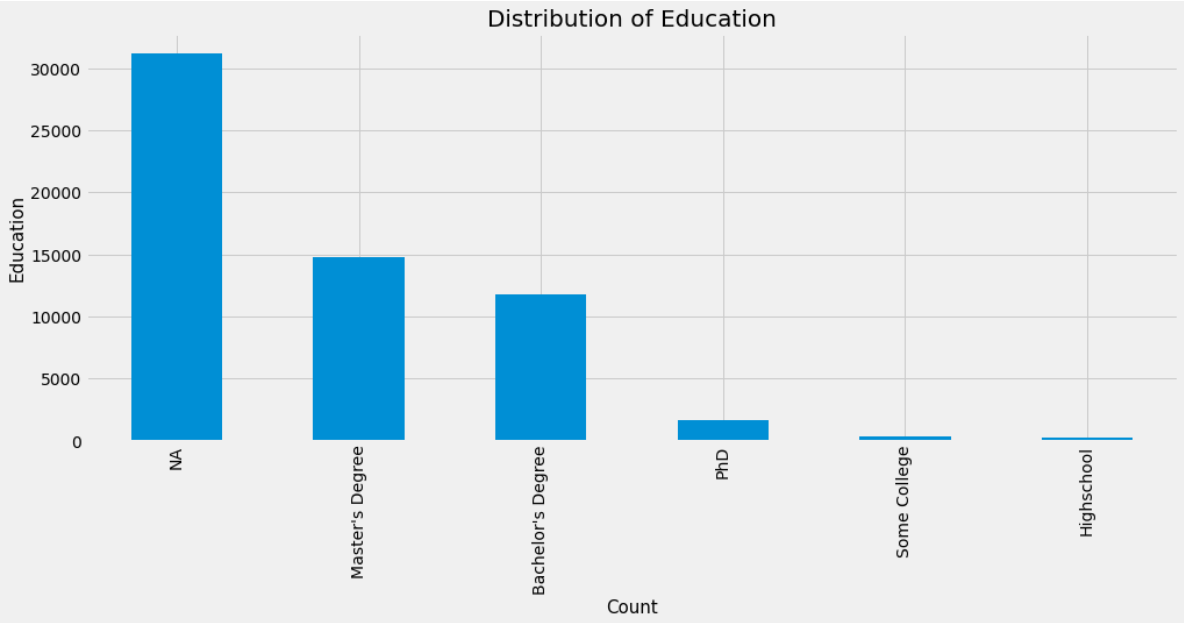
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
pay_by_education = data_salaries[['totalyearlycompensation','Education']].groupby(['Educati
plt.title('Compensation By Education', size=20)
plt.xlabel('Compensation ($)', size = 15)
plt.ylabel('Education', size = 15)
plt.show();
```

<Figure size 1080x432 with 0 Axes>



In [41]:

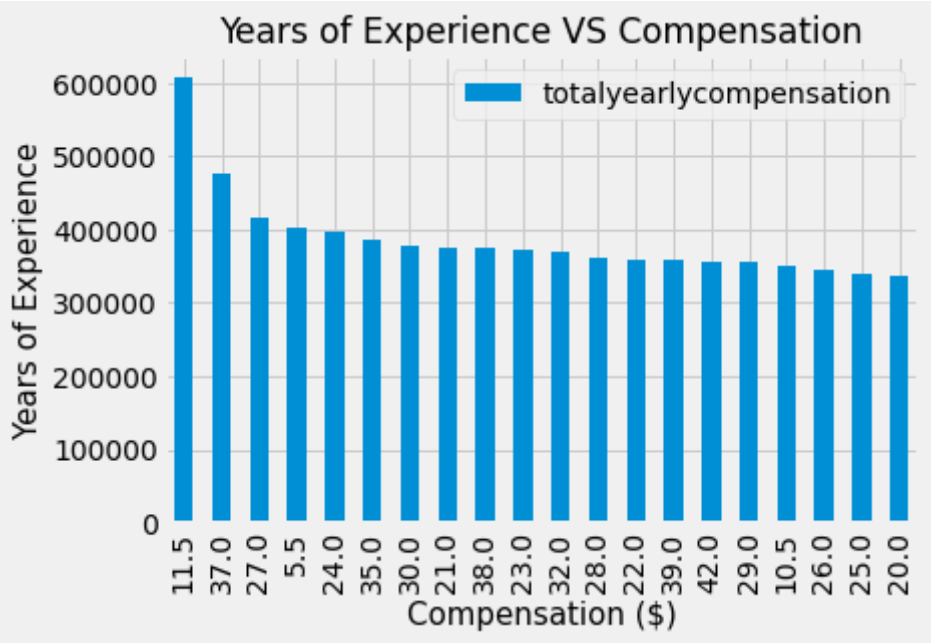
```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
education_distribution = data_salaries['Education'].value_counts().plot.bar()
plt.title('Distribution of Education', size=20)
plt.xlabel('Count', size = 15)
plt.ylabel('Education', size = 15)
plt.show();
```



In [42]:

```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
years_of_experience = data_salaries[['title', 'totalyearlycompensation','yearsofexperience'

plt.title('Years of Experience VS Compensation', size=17)
plt.xlabel('Compensation ($)', size = 15)
plt.ylabel('Years of Experience', size = 15)
plt.show();
```

```
<Figure size 1080x432 with 0 Axes>
```



In [43]:

```
plt.figure(figsize = (15,6))
plt.style.use('fivethirtyeight')
location = data_salaries['location'].value_counts().iloc[:20].plot.bar()
plt.title('Top 20 locations of tech jobs', size=17)
plt.xlabel('Number of workers', size = 15)
plt.ylabel('Company', size = 15)
plt.show();
```