DataVisualization

October 9, 2020

[6]:

```
#Import matpolot lib
import matplotlib.pyplot as plt
from matplotlib import style

%matplotlib inline
import numpy as np
import pandas as pd
```

[36]:

```
df = pd.read_csv('Datasetcln.csv')
```

[37]:

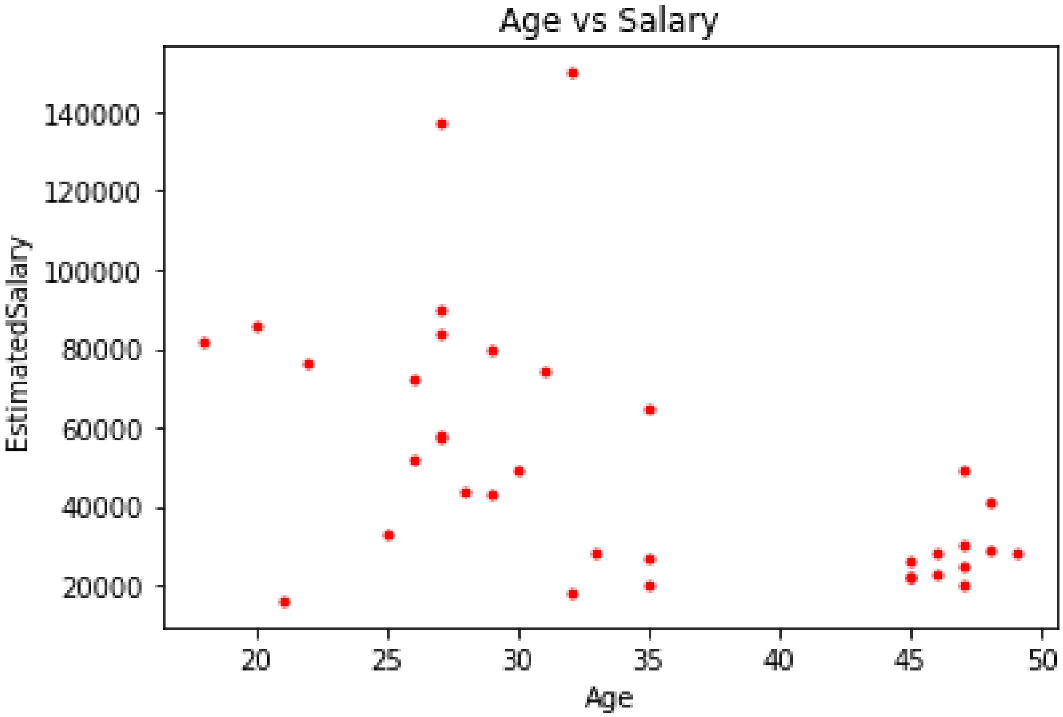
```
df.head(1)
```

[37]:

```
    User ID Gender  Age  EstimatedSalary  Purchased
0  15810944   Male   35            20000          0
```

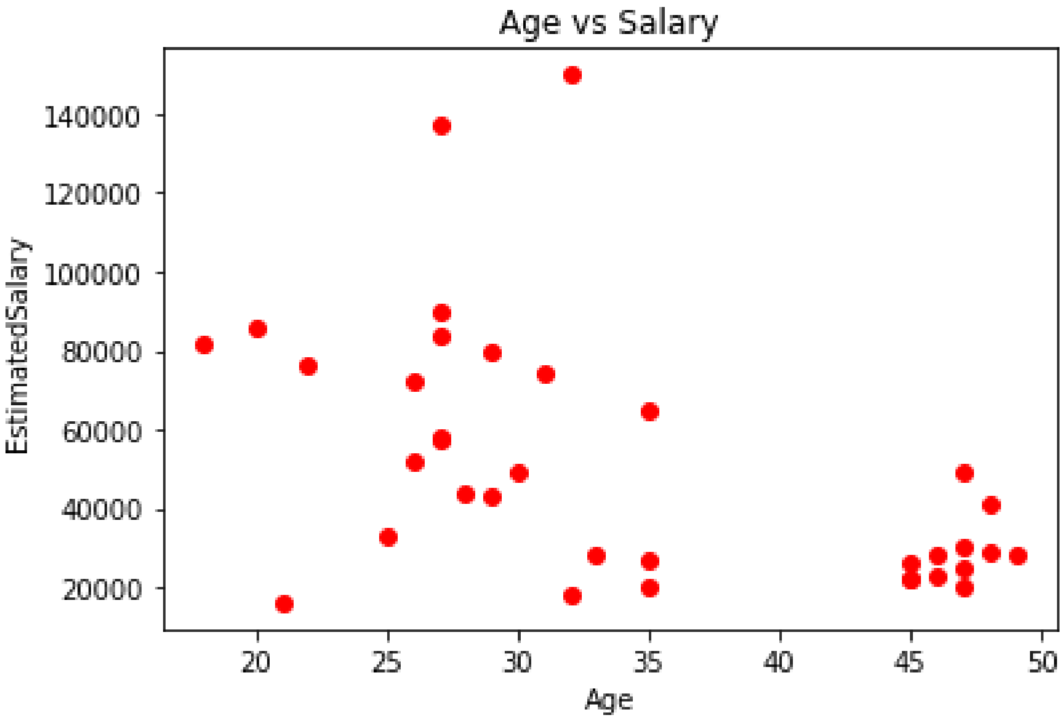
[61]:

```
# Simple plotting with matplotlib
plt.plot(df['Age'], df['EstimatedSalary'], 'r.')
plt.xlabel('Age')
plt.ylabel('EstimatedSalary')
plt.title('Age vs Salary')
plt.show()
```

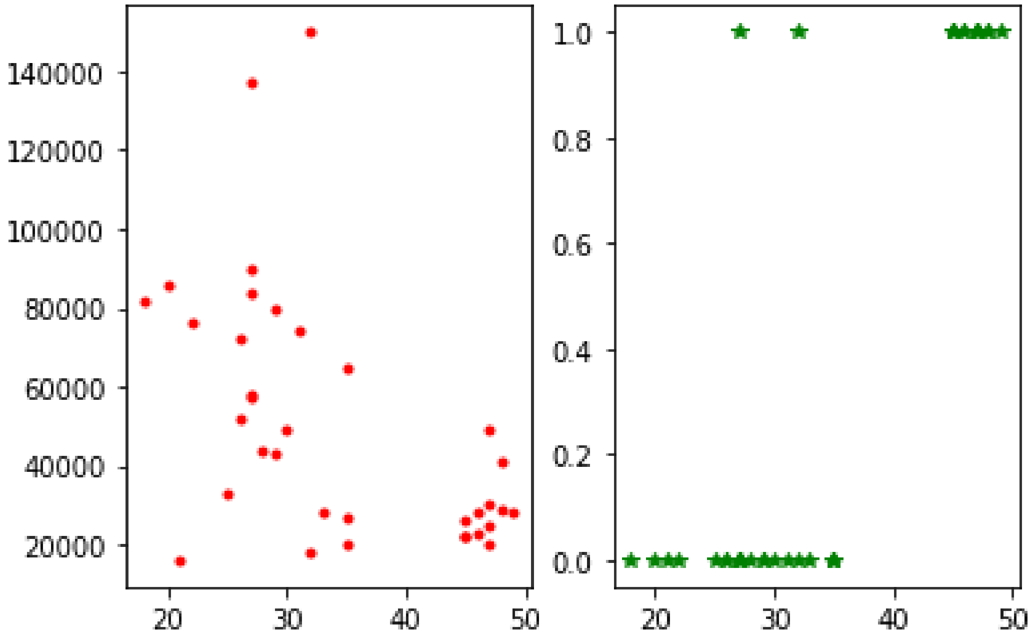


[58]:

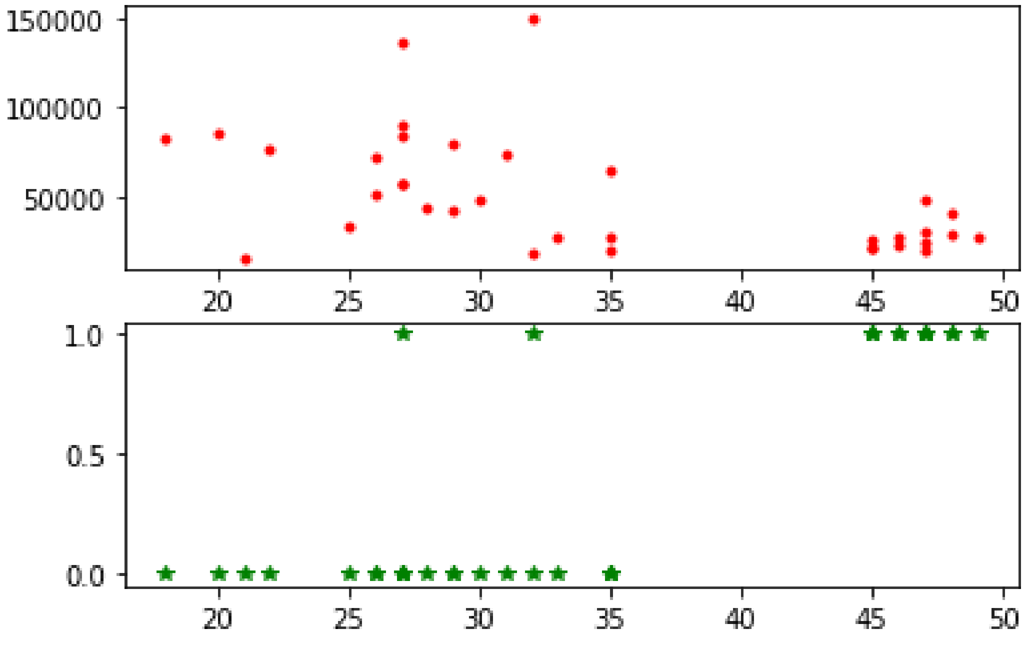
```
# Simple plotting with matplotlib
plt.scatter(df['Age'], df['EstimatedSalary'], color ='r')
plt.xlabel('Age')
plt.ylabel('EstimatedSalary')
plt.title('Age vs Salary')
plt.show()
```



[53]:
```
# Subplot in matplotlib plt.subplot(nrows, ncols, plot_number)
plt.subplot(1,2,1)
plt.plot(df['Age'], df['EstimatedSalary'], 'r.')

plt.subplot(1,2,2)
plt.plot(df['Age'], df['Purchased'], 'g*')
plt.show()
```

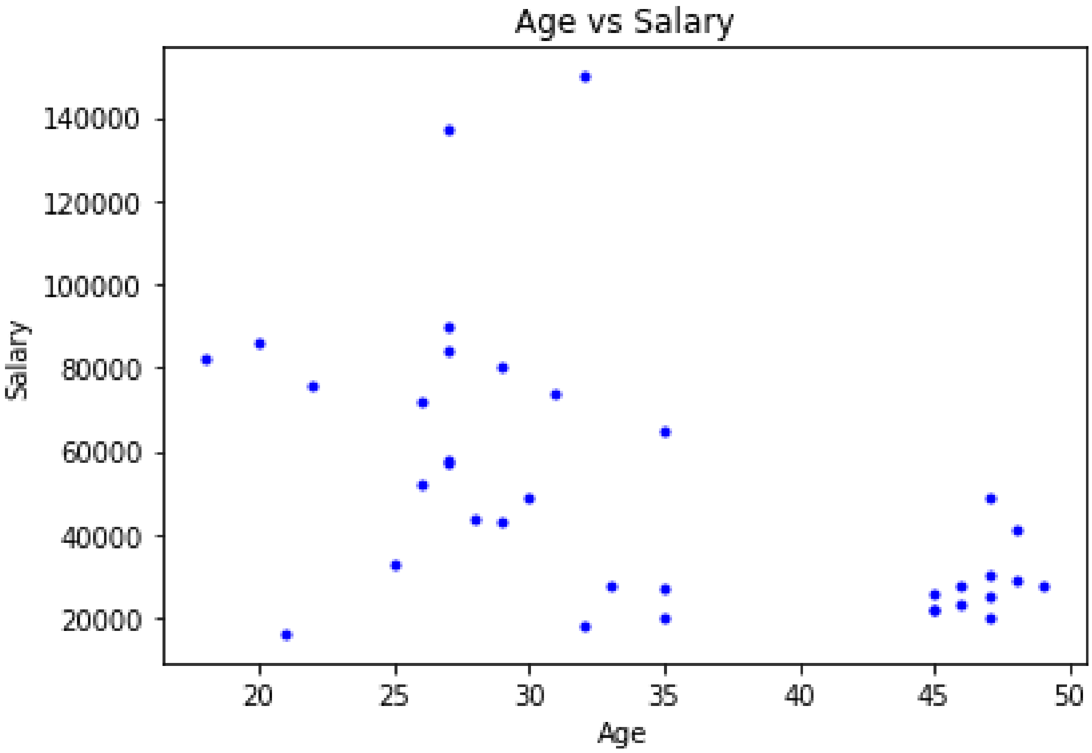


```
[52]: # Subplot in matplotlib plt.subplot(nrows, ncols, plot_number)
      plt.subplot(2,1,1)
      plt.plot(df['Age'], df['EstimatedSalary'], 'r.')

      plt.subplot(2,1,2)
      plt.plot(df['Age'], df['Purchased'], 'g*')
      plt.show()
```



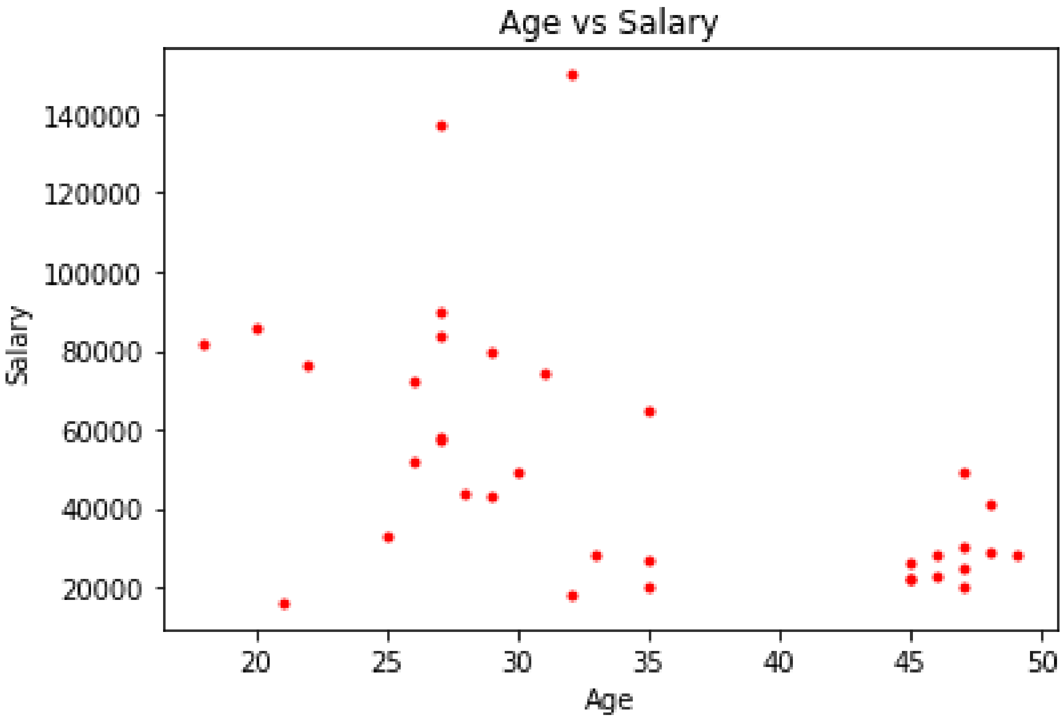
[49]:

```
# Object oriented programming style
fig = plt.figure()

#plot size add_axis(left, bottom, width, height)
axes = fig.add_axes([0.1,0.1, 0.8, 0.8])
axes.plot(df.Age, df.EstimatedSalary, 'b.')
axes.set_xlabel('Age')
axes.set_ylabel('Salary')
axes.set_title('Age vs Salary')
plt.show()
```

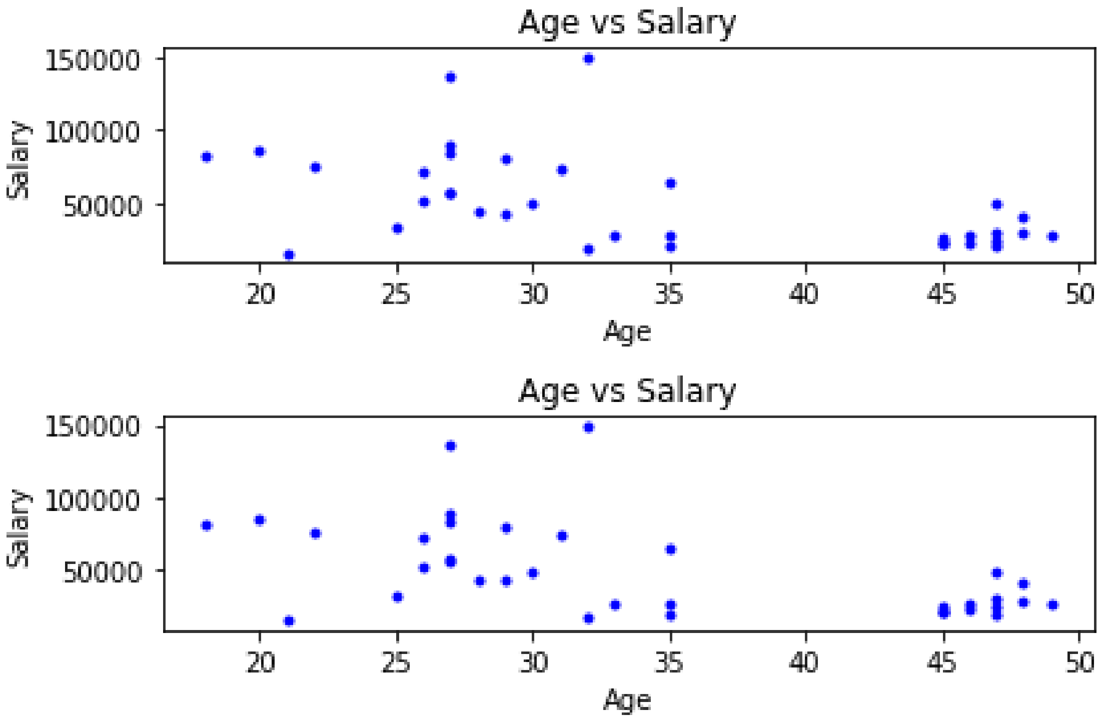


[48]:

```
# subplot() as OOO
fig, axes = plt.subplots()

axes.plot(df.Age, df.EstimatedSalary, 'r.')
axes.set_xlabel('Age')
axes.set_ylabel('Salary')
axes.set_title('Age vs Salary')
plt.show()
```

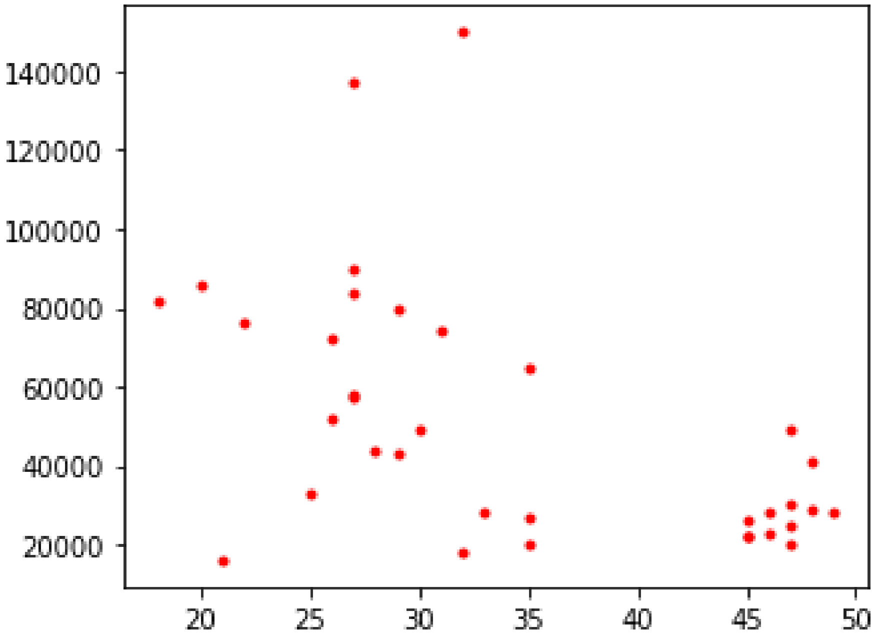


[84]:
```
# subplot() as OOO
fig, axes = plt.subplots(nrows = 2, ncols = 1)

for ax in axes:
    ax.plot(df.Age, df.EstimatedSalary, 'b.')
    ax.set_xlabel('Age')
    ax.set_ylabel('Salary')
    ax.set_title('Age vs Salary')

fig
plt.tight_layout()
```



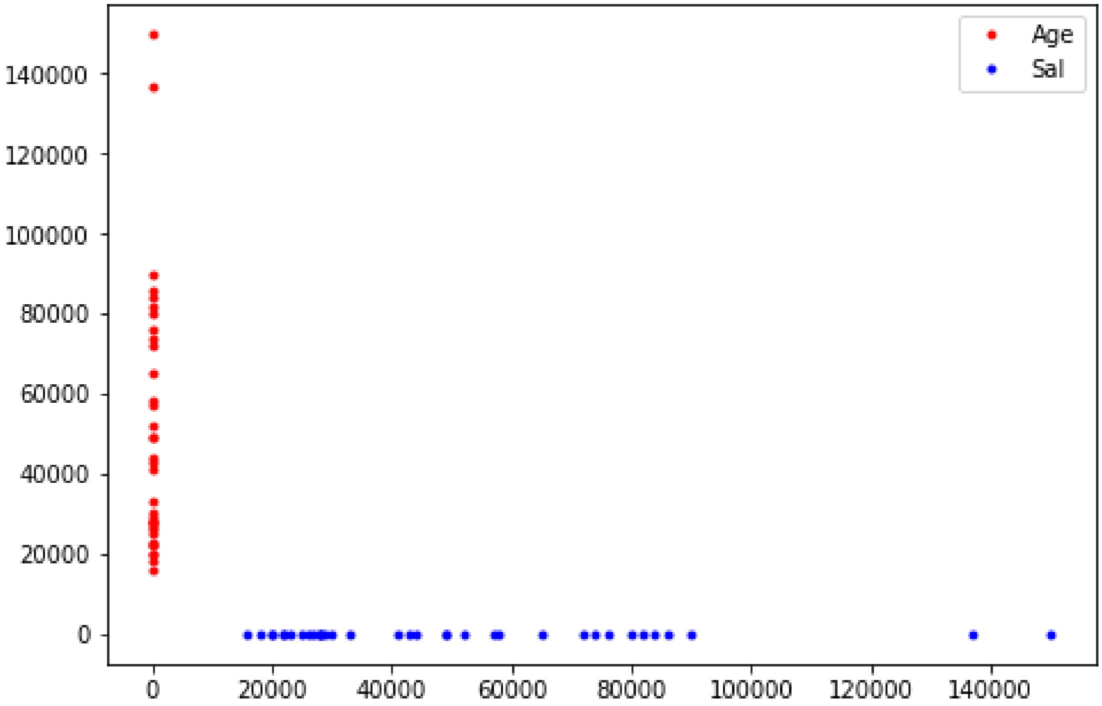
[118]:
```
# Initializing figsize and dpi
fig = plt.figure(figsize=(8,4), dpi = 100)

fig, axes = plt.subplots(figsize = (5,4))
axes.plot(df.Age, df.EstimatedSalary, 'r.')
plt.show()
```

```
<Figure size 800x400 with 0 Axes>
```



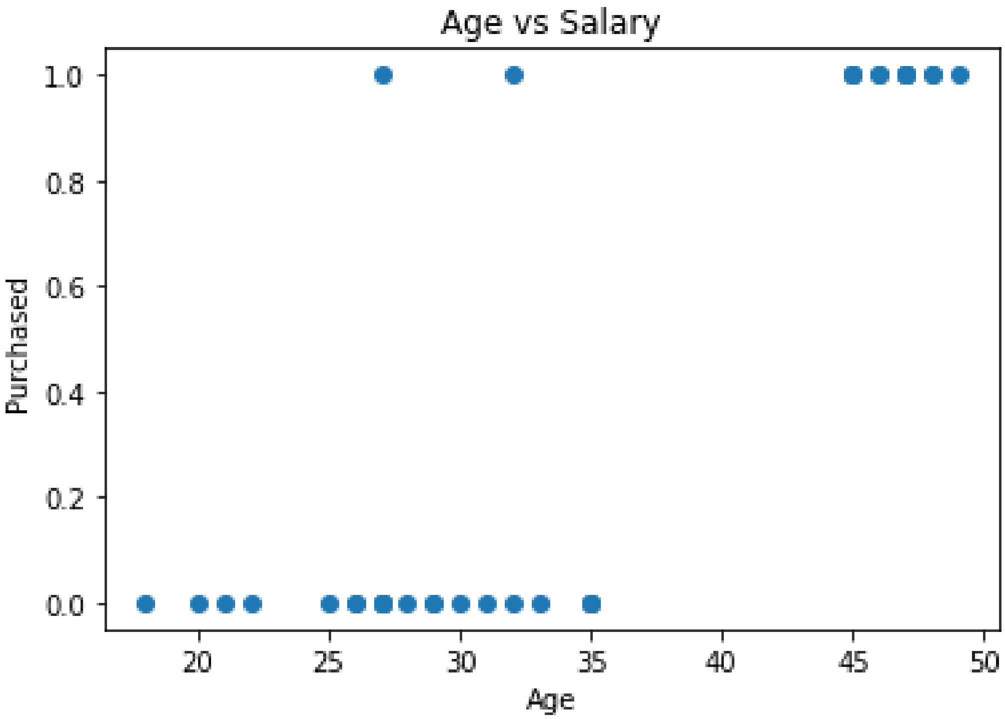
[94]:
```
fig.savefig('test.png', dpi = 200)
```

[113]:
```
#Legends, labels, and titles
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])

ax.plot(df.Age, df.EstimatedSalary, 'r.', label ='Age')
ax.plot(df.EstimatedSalary, df.Age, 'b.', label = 'Sal')
ax.legend()
plt.show()
```



1 Scatter Plot

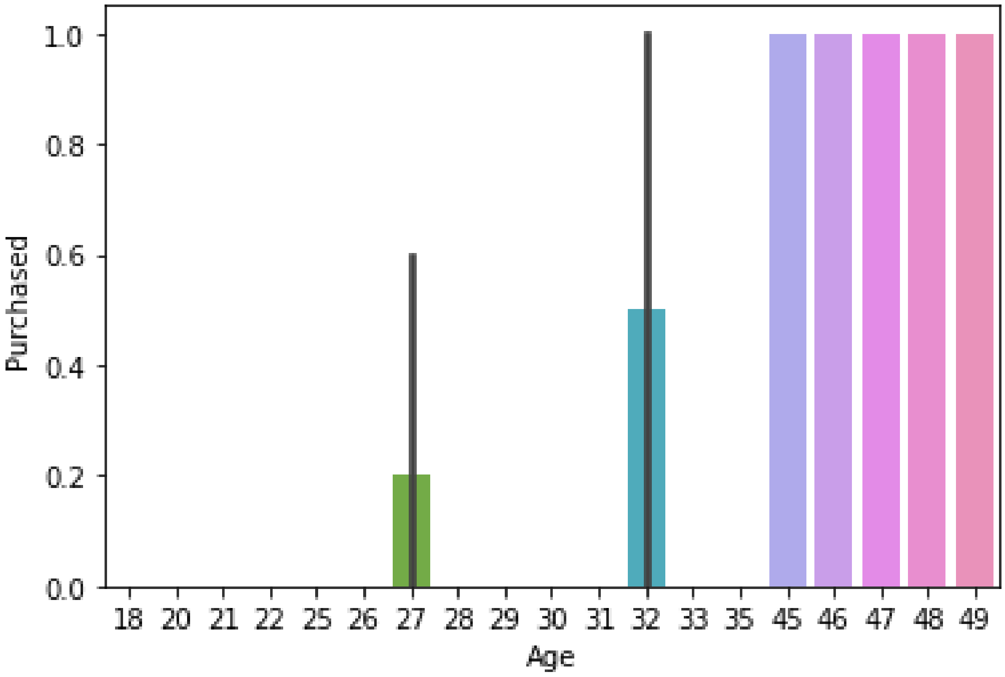
```
[122]: plt.scatter(df['Age'], df['Purchased'])
       plt.xlabel('Age')
       plt.ylabel('Purchased')
       plt.title('Age vs Salary')
       plt.show()
```



2 Seaborn Library

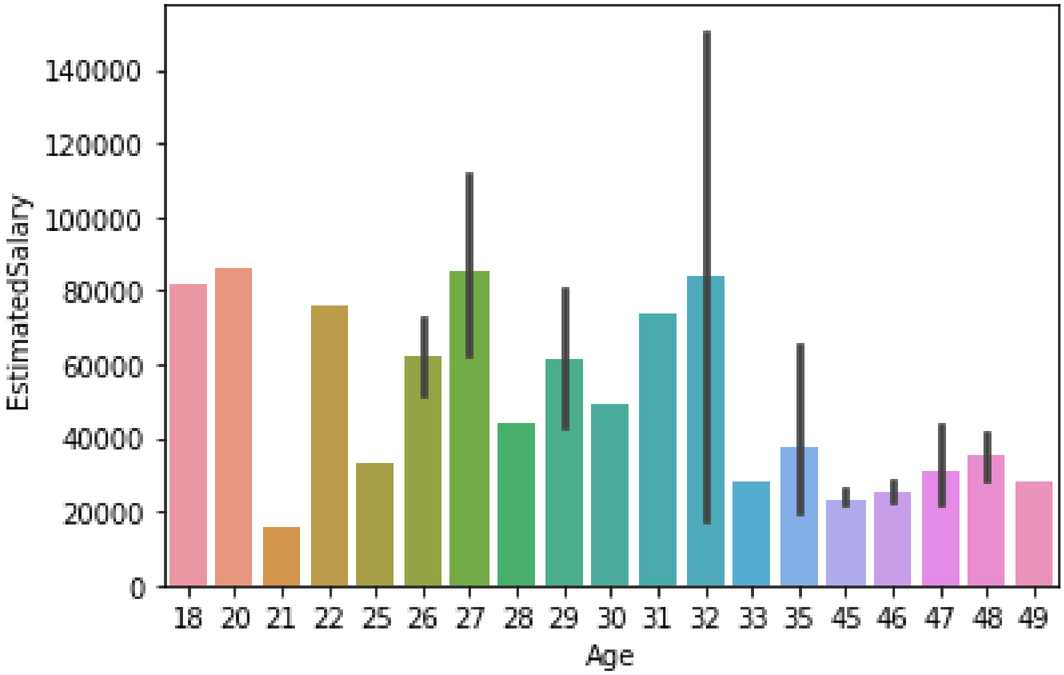
[144]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('Datasetcln.csv')
```

[145]:
```
# Bar plot
sns.barplot(x='Age',y='Purchased',data=df)
plt.show()
```

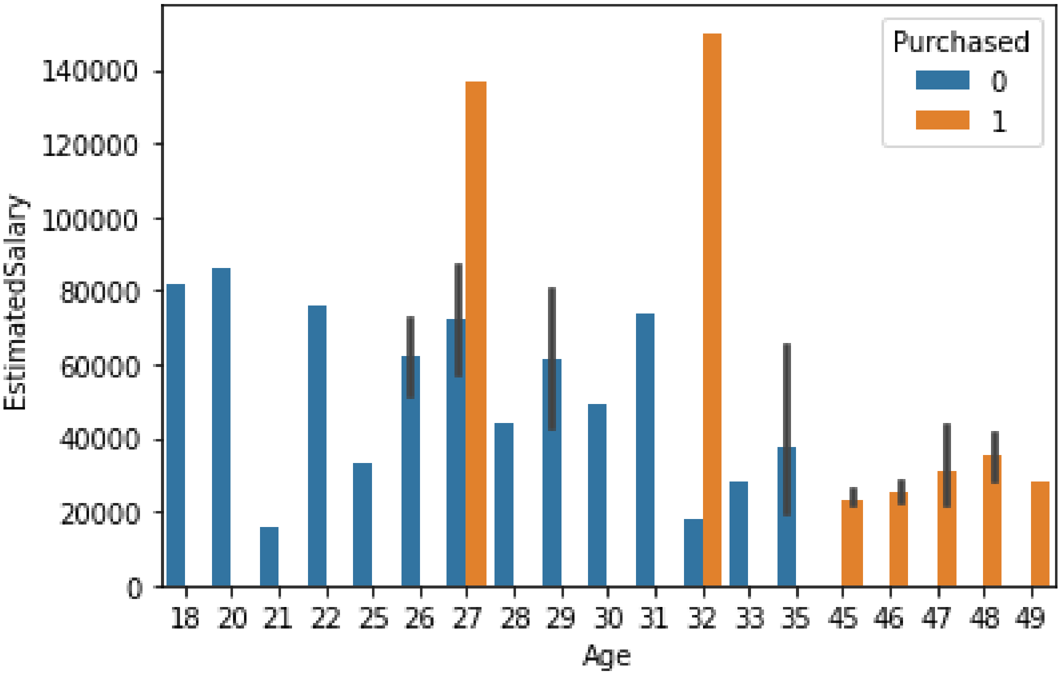


[146]:

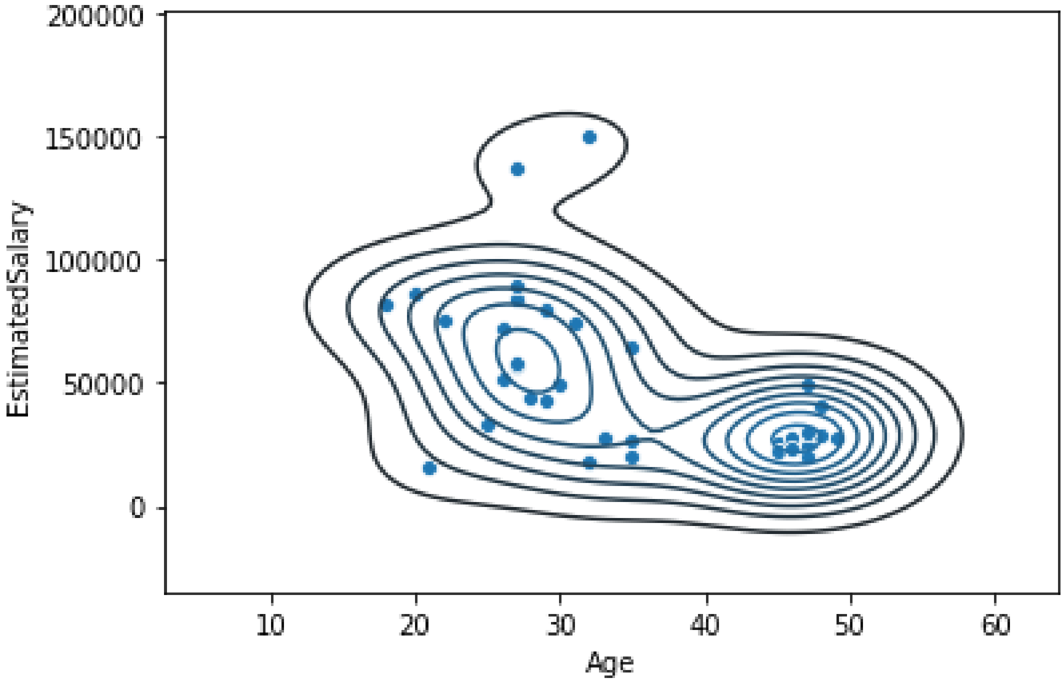
```
# Bar plot
sns.barplot(x='Age',y='EstimatedSalary',data=df)
plt.show()
```



[147]:
```
# Bar plot
sns.barplot(x='Age',y='EstimatedSalary',data=df, hue = 'Purchased')
plt.show()
```



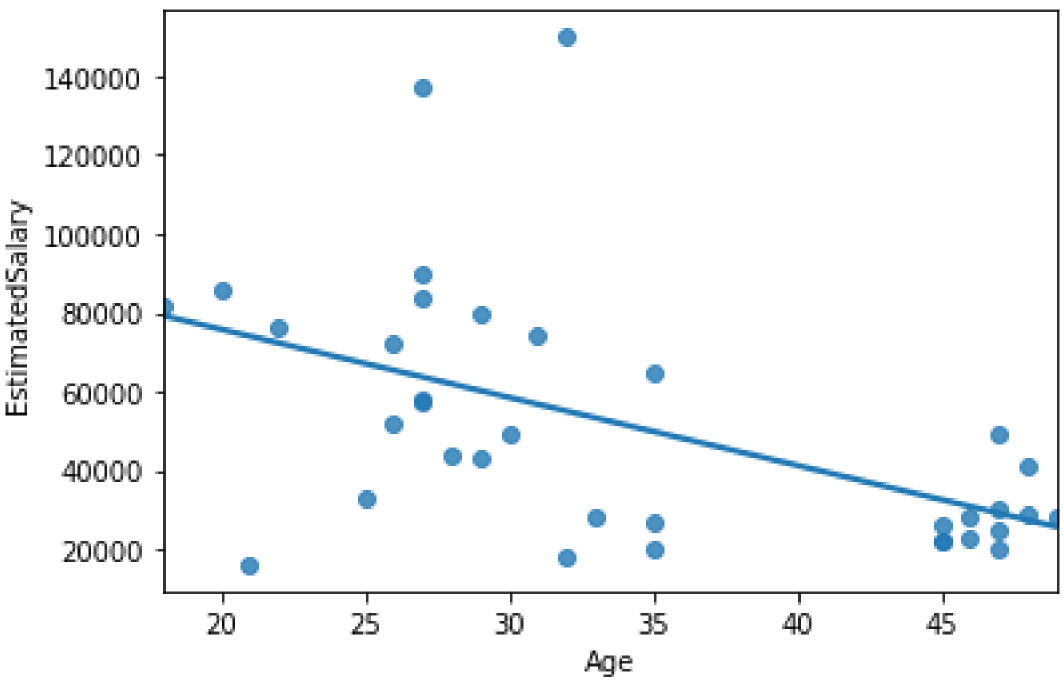
[154]:
```
sns.scatterplot(x='Age', y='EstimatedSalary', data=df)
sns.kdeplot(df.Age, df.EstimatedSalary)
plt.show()
```



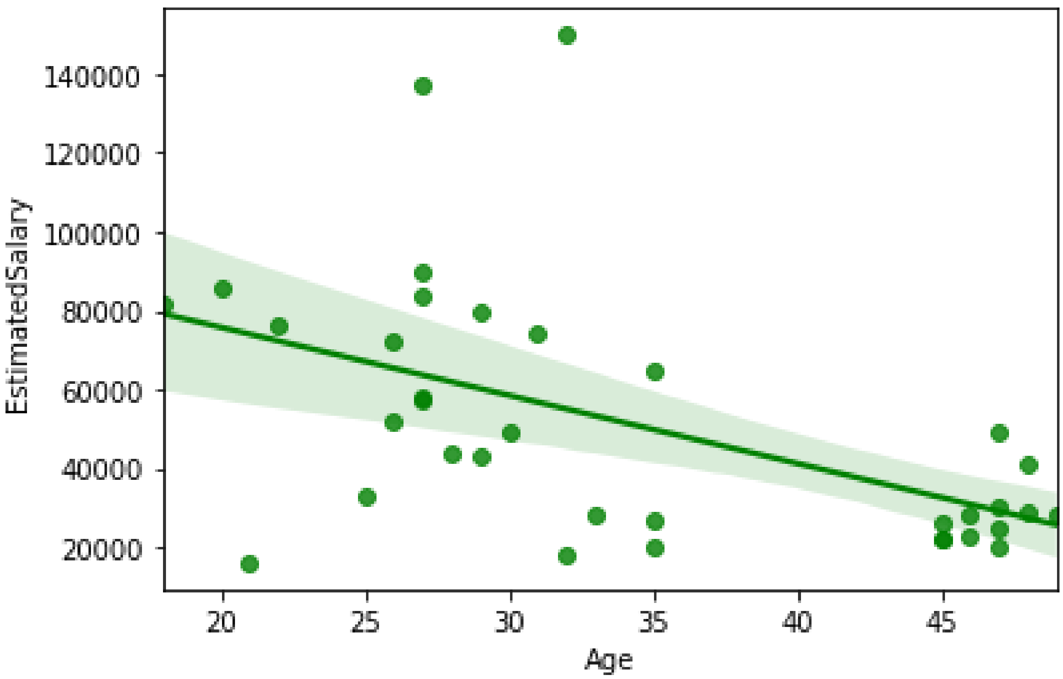
```
[149]: df.head(1)
```

```
[149]:     User ID Gender  Age  EstimatedSalary  Purchased
       0  15810944   Male   35            20000          0
```

```
[155]: sns.regplot(x='Age', y='EstimatedSalary', ci=None, data=df)
       plt.show()
```

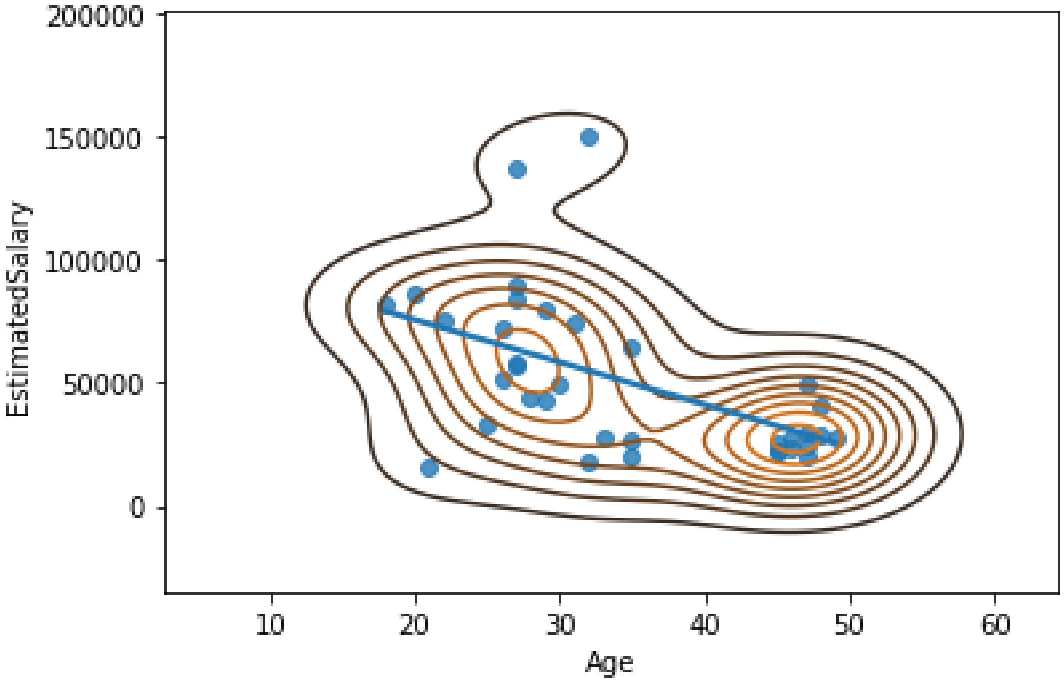


[156]:
```
sns.regplot(x='Age', y='EstimatedSalary', color='g', data=df)
plt.show()
```



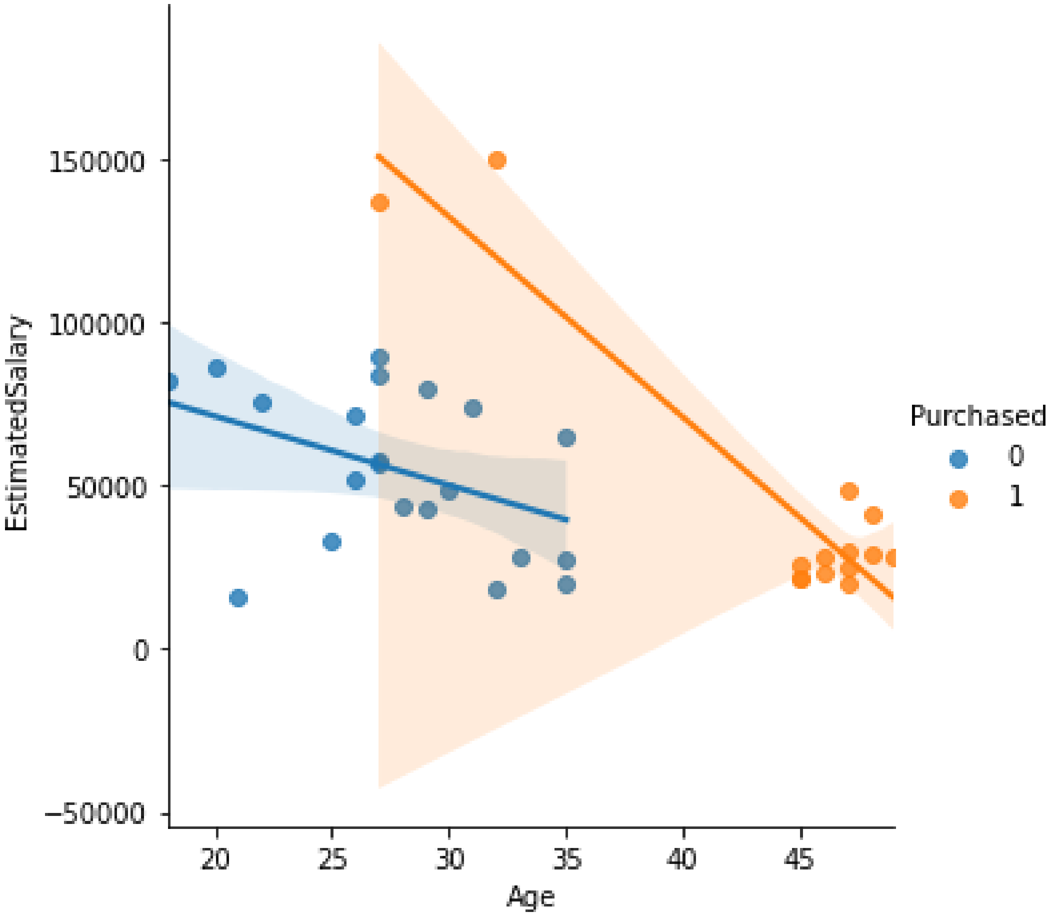
[158]:

```
sns.regplot(x='Age', y='EstimatedSalary', ci=None, data=df)
sns.kdeplot(df.Age, df.EstimatedSalary)
plt.show()
```



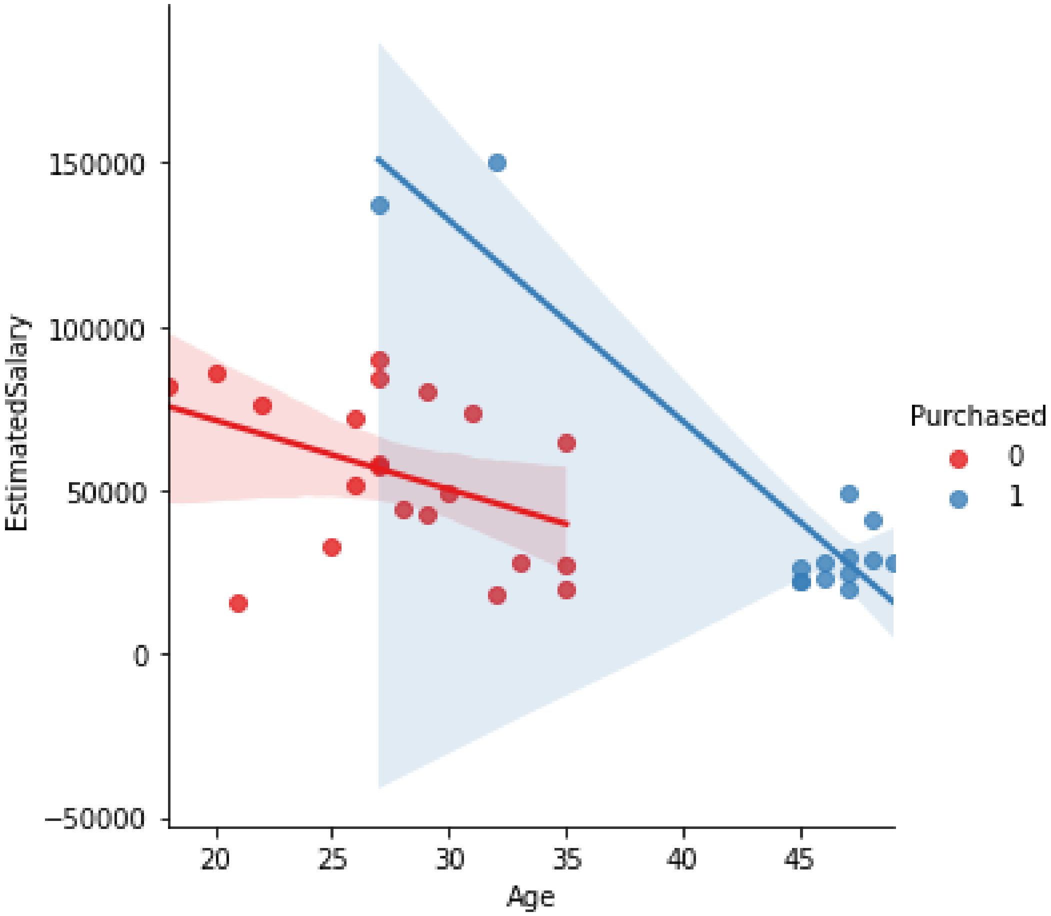
[159]:

```
sns.lmplot(x='Age', y='EstimatedSalary',hue='Purchased', data=df)
plt.show()
```



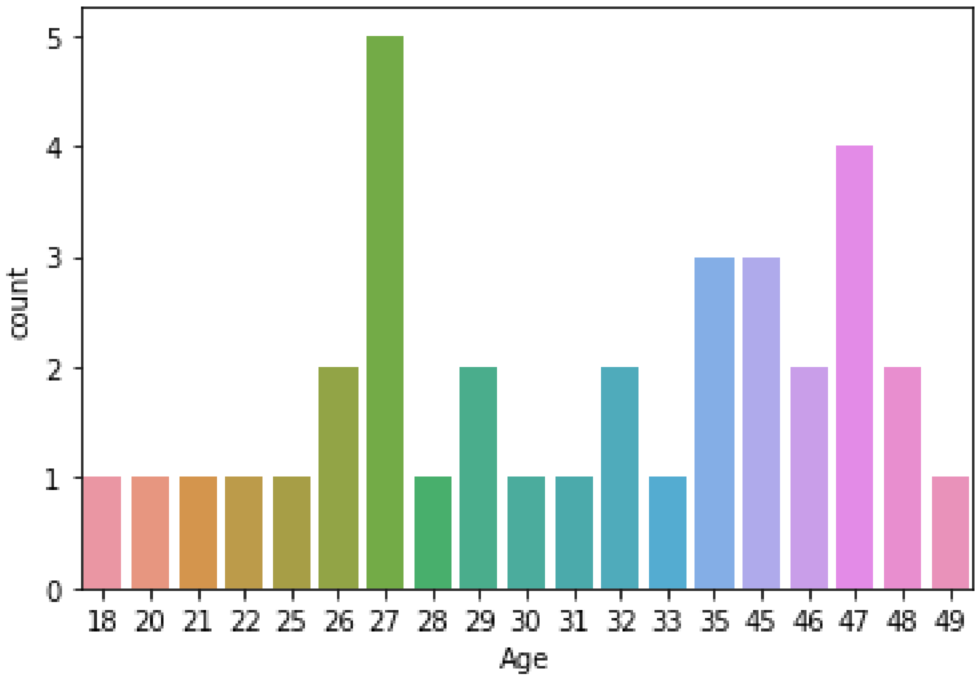
```
[160]: sns.lmplot(x='Age', y='EstimatedSalary', hue='Purchased', palette="Set1",␣
        ↪data=df)
```

```
[160]: <seaborn.axisgrid.FacetGrid at 0x7fd772b8cc40>
```

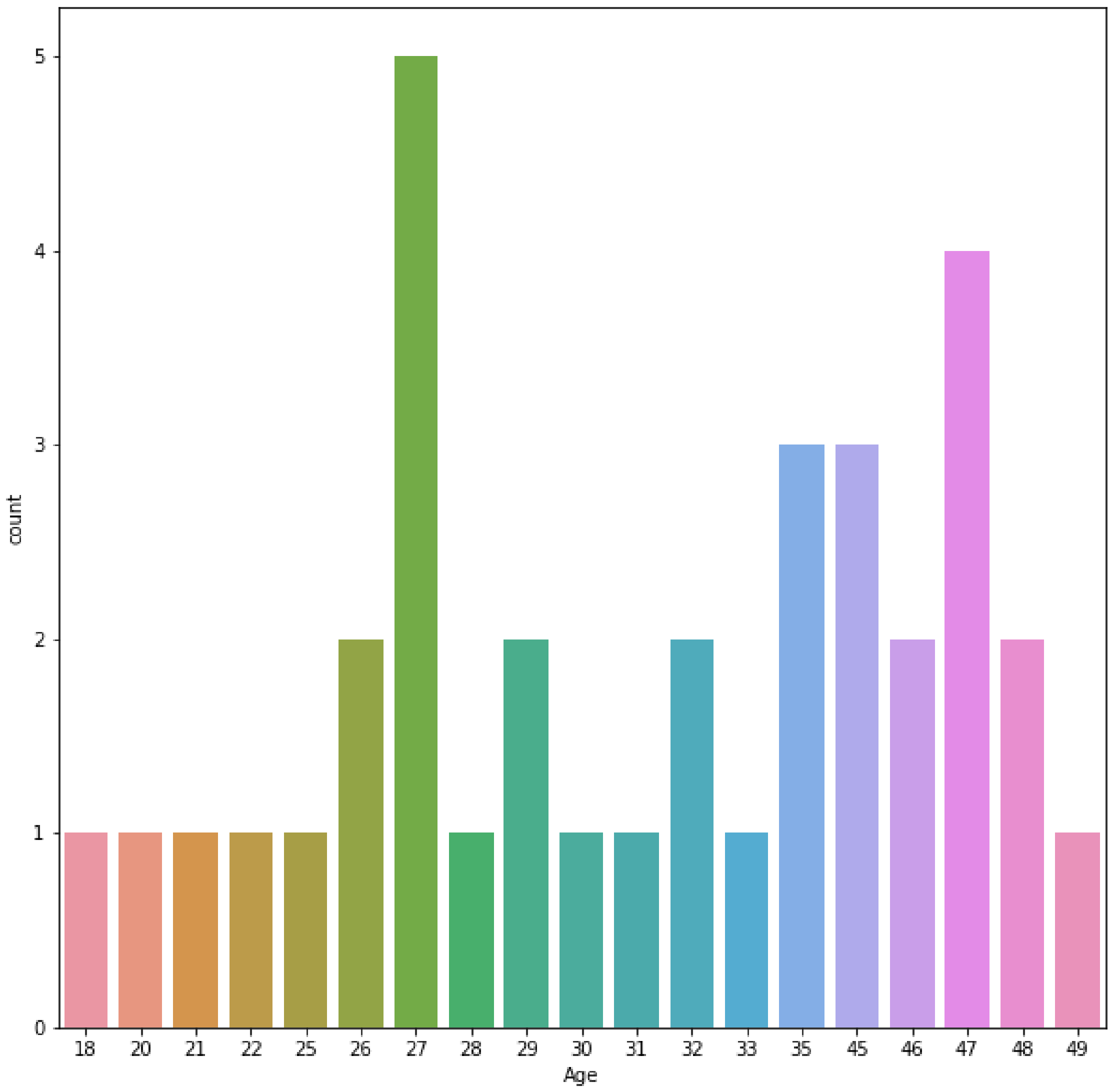


3 Count Plot

[161]:
```
sns.countplot(x='Age',data=df)
plt.show()
```



[167]:
```
fig_d = (10, 10)
fig, ax = plt.subplots(figsize=fig_d)

sns.countplot(x='Age',ax = ax, data=df)
plt.show()
```

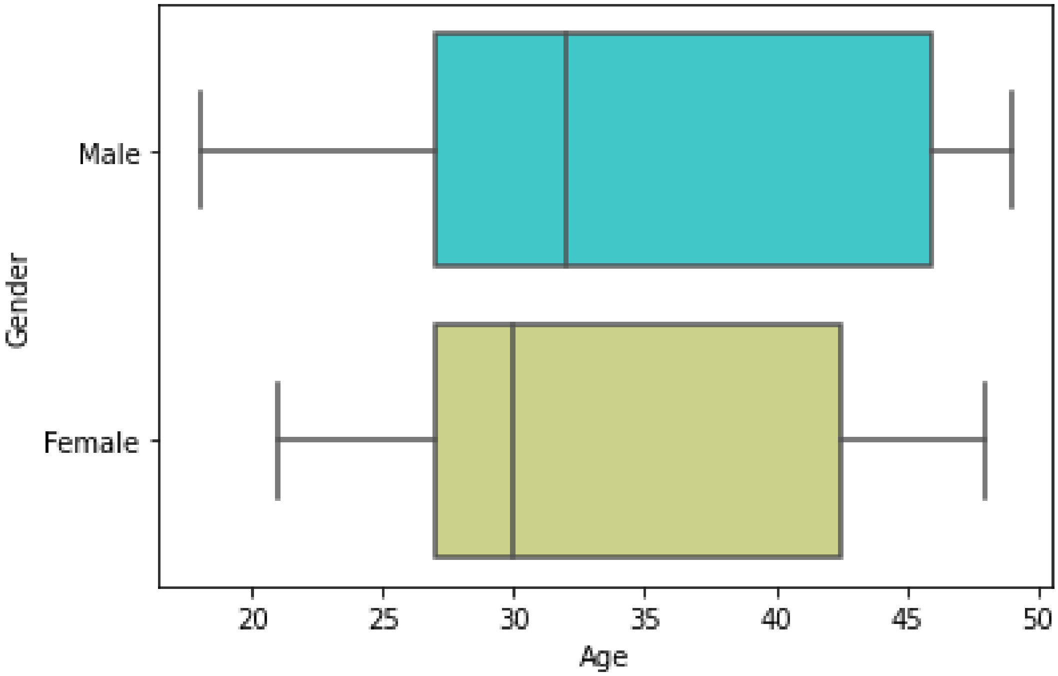


4 Box Plot

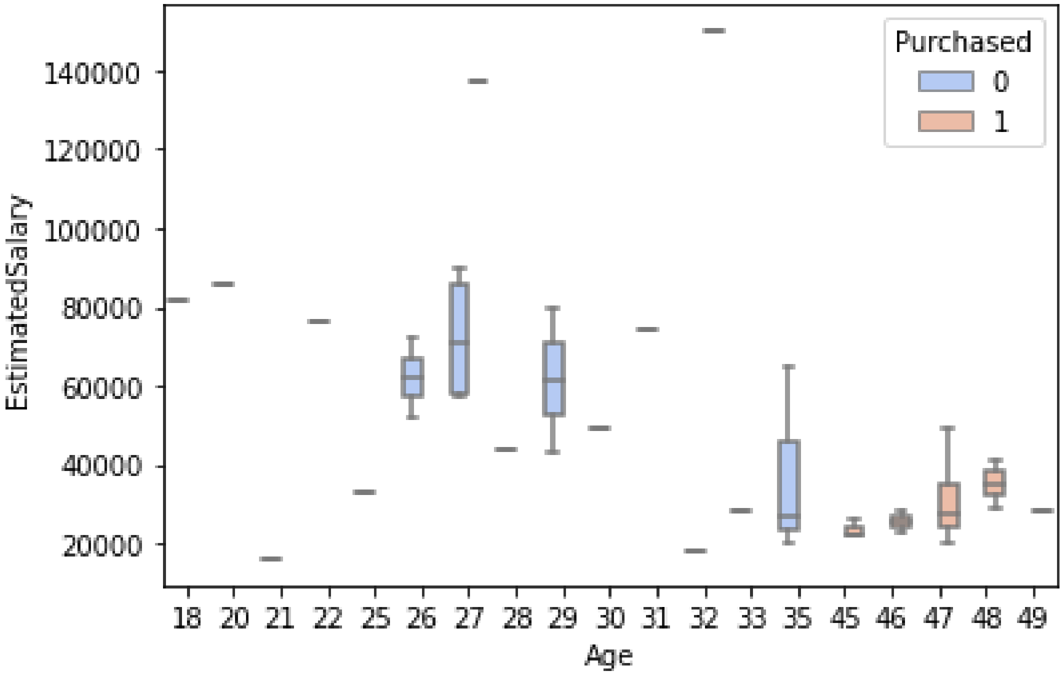
A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.

[168]:

```
# Box plot
sns.boxplot(x='Age',y='Gender',data=df,palette='rainbow')
plt.show()
```



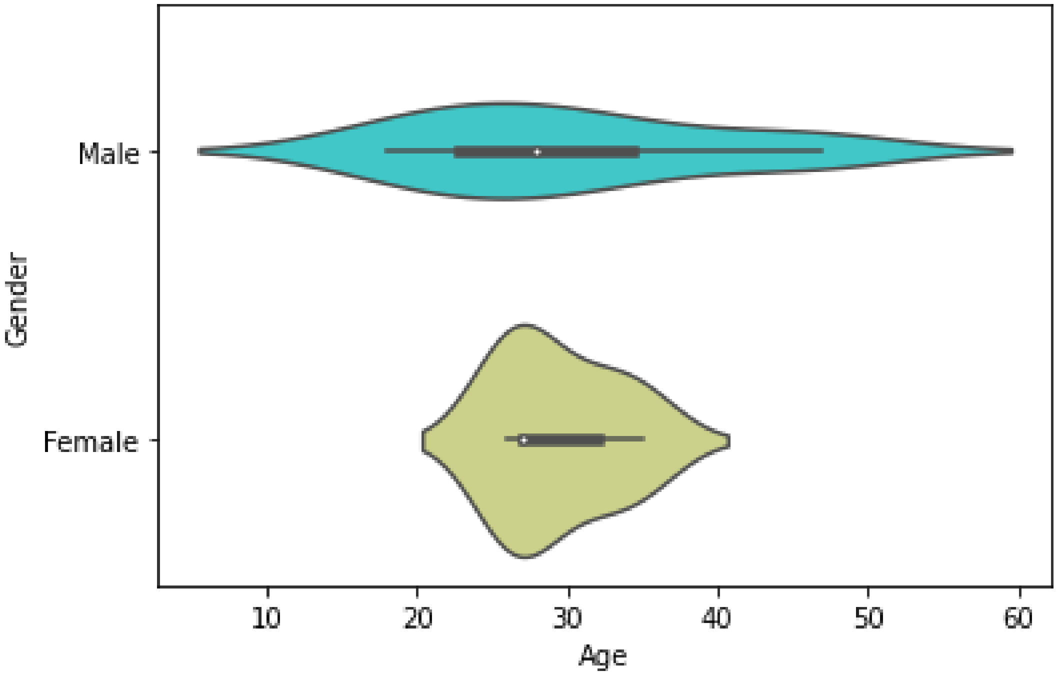
[170]:
```
sns.boxplot(x="Age",y="EstimatedSalary",hue="Purchased",data=df,␣
↪palette="coolwarm")
plt.show()
```



5 Violin plot

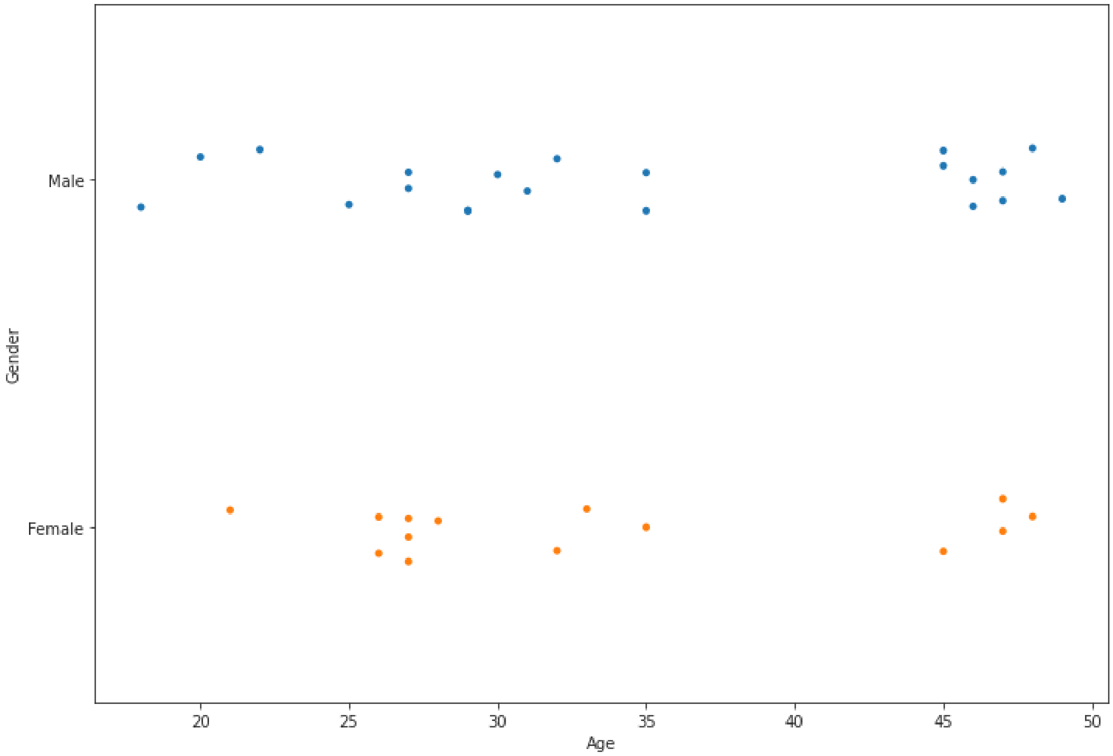
A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints, the violin plot features a kernel density estimation of the underlying distribution.

[180]:
```
sns.violinplot(x="Age", y="Gender", data=df.iloc[0:15,0:15], palette='rainbow')
plt.show()
```



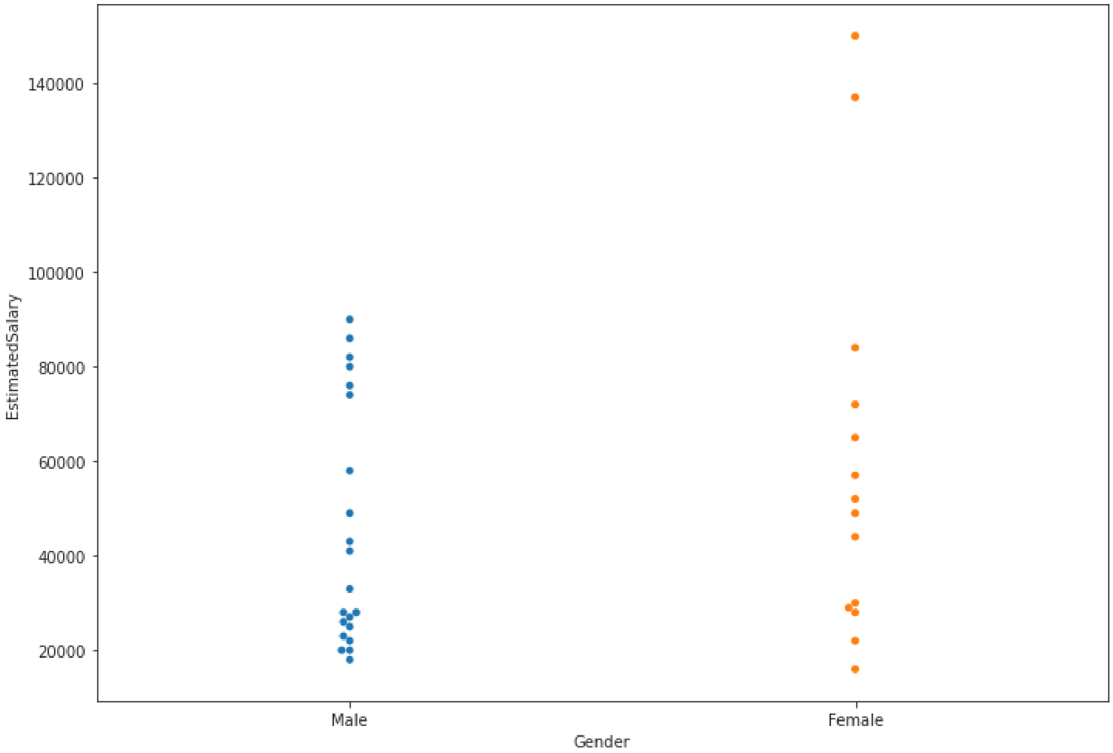
[182]:
```
sns.set_style('ticks')
fig, ax = plt.subplots()
# the size of A4 paper
fig.set_size_inches(11.7, 8.27)
sns.stripplot(x="Age", y="Gender", data=df, ax= ax)
```

[182]:
```
<AxesSubplot:xlabel='Age', ylabel='Gender'>
```



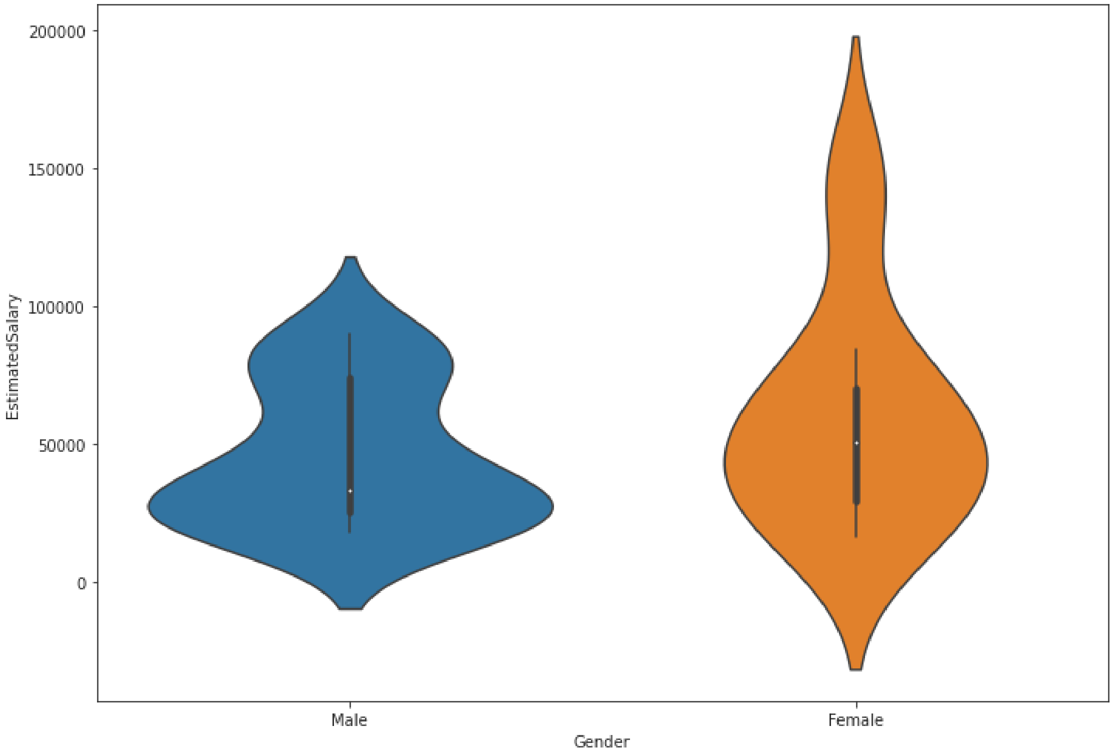
```
[185]: sns.set_style('ticks')
       fig, ax = plt.subplots()
       # the size of A4 paper
       fig.set_size_inches(11.7, 8.27)
       sns.swarmplot(x="Gender", y="EstimatedSalary", data=df, ax= ax)
```

```
[185]: <AxesSubplot:xlabel='Gender', ylabel='EstimatedSalary'>
```

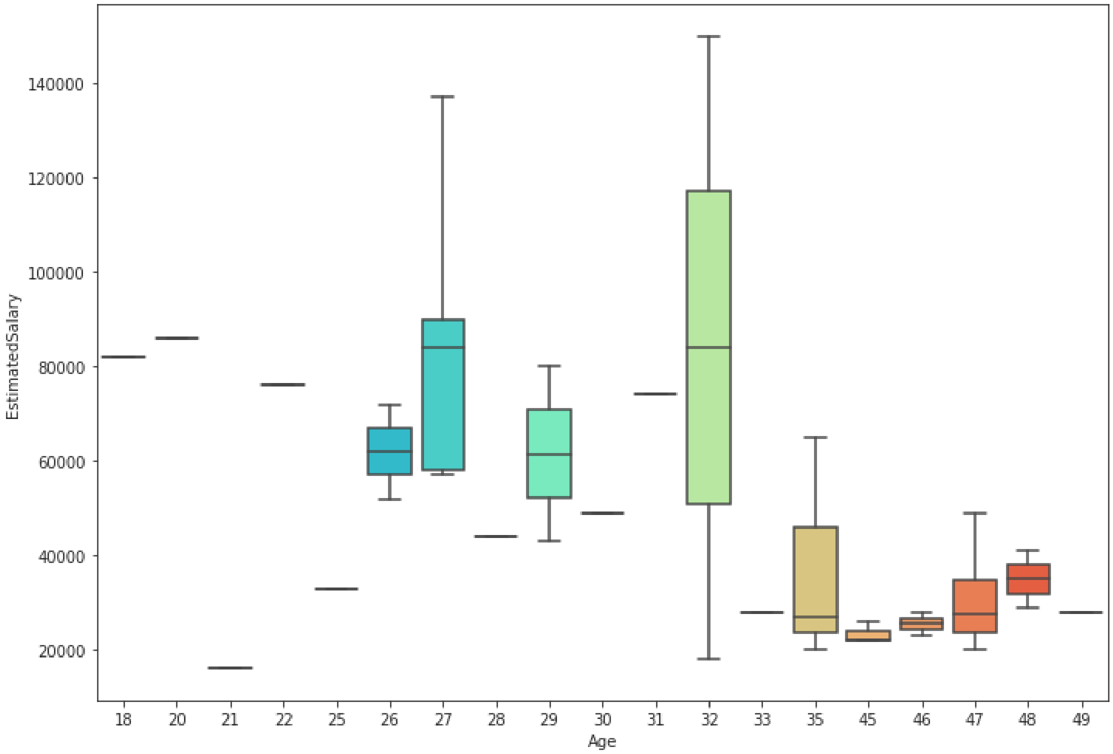


[186]:
```
sns.set_style('ticks')
fig, ax = plt.subplots()
# the size of A4 paper
fig.set_size_inches(11.7, 8.27)
sns.violinplot(x="Gender", y="EstimatedSalary", data=df, ax= ax)
```

[186]:
```
<AxesSubplot:xlabel='Gender', ylabel='EstimatedSalary'>
```

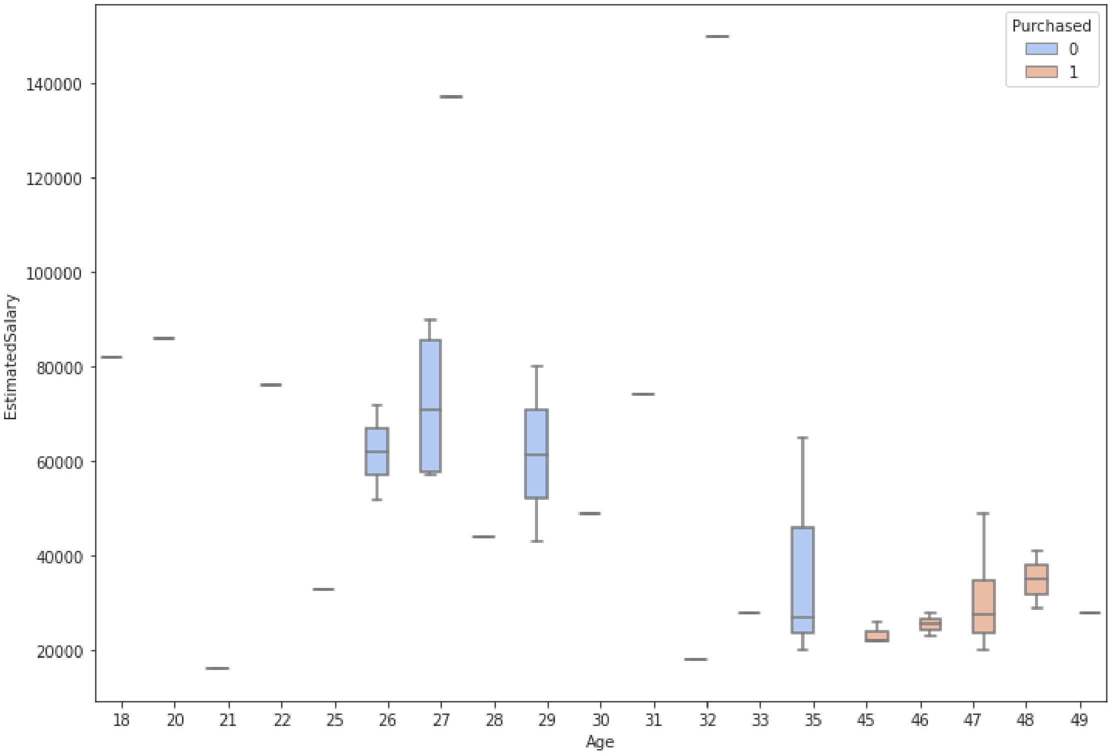


[190]:
```
sns.set_style('ticks')
fig, ax = plt.subplots()
# the size of A4 paper
fig.set_size_inches(11.7, 8.27)
sns.boxplot(x="Age", y="EstimatedSalary", data=df, palette='rainbow', ax= ax)
plt.show()
```



[191]:
```
sns.set_style('ticks')
fig, ax = plt.subplots()
# the size of A4 paper
fig.set_size_inches(11.7, 8.27)
sns.boxplot(x="Age",y="EstimatedSalary",hue="Purchased",data=df,␣
 ↪palette="coolwarm", ax= ax)
```

[191]:
```
<AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>
```


Purchased
0
1
140000
120000
100000
80000
60000
40000
20000
EstimatedSalary
18
20
21
22
25
26
27
28
29
30
31
32
33
35
45
46
47
48
49
Age

[ ]: