In [33]:
```
df.head()
```

Out[33]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86
2	3	15619304	Onio	502	France	Female	42	8	159660.80
3	4	15701354	Boni	699	France	Female	39	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82

In [35]:
```
df.Balance
```

Out[35]:
```
0           0.00
1       83807.86
2      159660.80
3           0.00
4      125510.82
         ...
9995        0.00
9996    57369.61
9997        0.00
9998    75075.31
9999   130142.79
Name: Balance, Length: 10000, dtype: float64
```

In [37]:
```
df[["Balance", "Exited"]]
```

Out[37]:

	Balance	Exited
0	0.00	1
1	83807.86	0
2	159660.80	1
3	0.00	0
4	125510.82	0
...	...	...
9995	0.00	0
9996	57369.61	0
9997	0.00	1
9998	75075.31	1
9999	130142.79	0

In [38]:
```
df[["Balance", "Exited"]].describe()
```

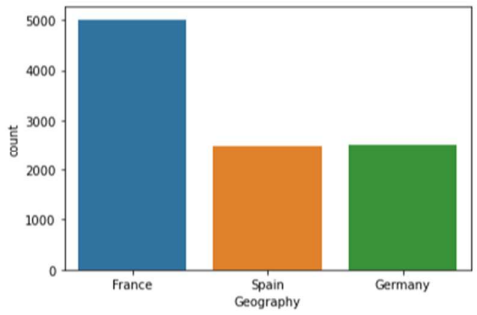
Out[38]:

	Balance	Exited
count	10000.000000	10000.000000
mean	76485.889288	0.203700
std	62397.405202	0.402769
min	0.000000	0.000000
25%	0.000000	0.000000
50%	97198.540000	0.000000
75%	127644.240000	0.000000
max	250898.090000	1.000000

Visualization

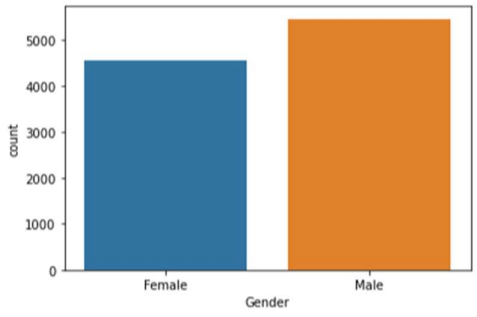
In [41]:
```
import seaborn as sns
sns.countplot(x= 'Geography', data = df)
```

Out[41]:
```
<AxesSubplot:xlabel='Geography', ylabel='count'>
```



In [42]:
```
sns.countplot(x= 'Gender', data = df)
```

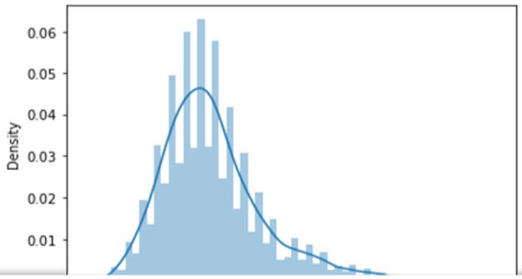
Out[42]:
```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



In [44]:
```
sns.distplot(df['Age'])
```

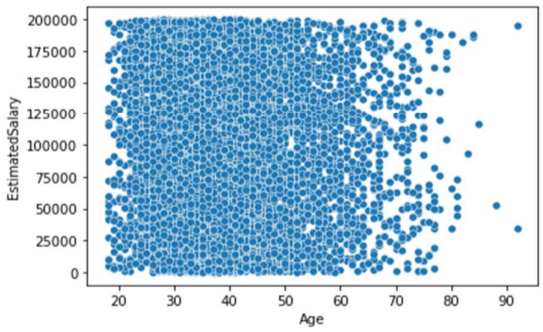
```
C:\Users\tanis\anaconda3\lib\site-packages\seaborn\distribution
and will be removed in a future version. Please adapt your code
flexibility) or `histplot` (an axes-level function for histogra
  warnings.warn(msg, FutureWarning)
```

Out[44]:
```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



In [45]:
```
sns.scatterplot(x= df["Age"], y= 'EstimatedSalary', data= df)
```

Out[45]:
```
<AxesSubplot:xlabel='Age', ylabel='EstimatedSalary'>
```



Matplotlib

In [48]:
```
x = np.arange(0,2*math.pi, 0.05)
x
```

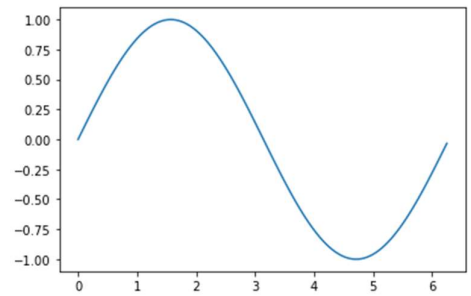
Out[48]:
```
array([0.  , 0.05, 0.1 , 0.15, 0.2 , 0.25, 0.3 , 0.35, 0.4 , 0.45, 0.5 ,
       0.55, 0.6 , 0.65, 0.7 , 0.75, 0.8 , 0.85, 0.9 , 0.95, 1.  , 1.05,
       1.1 , 1.15, 1.2 , 1.25, 1.3 , 1.35, 1.4 , 1.45, 1.5 , 1.55, 1.6 ,
       1.65, 1.7 , 1.75, 1.8 , 1.85, 1.9 , 1.95, 2.  , 2.05, 2.1 , 2.15,
       2.2 , 2.25, 2.3 , 2.35, 2.4 , 2.45, 2.5 , 2.55, 2.6 , 2.65, 2.7 ,
       2.75, 2.8 , 2.85, 2.9 , 2.95, 3.  , 3.05, 3.1 , 3.15, 3.2 , 3.25,
       3.3 , 3.35, 3.4 , 3.45, 3.5 , 3.55, 3.6 , 3.65, 3.7 , 3.75, 3.8 ,
       3.85, 3.9 , 3.95, 4.  , 4.05, 4.1 , 4.15, 4.2 , 4.25, 4.3 , 4.35,
       4.4 , 4.45, 4.5 , 4.55, 4.6 , 4.65, 4.7 , 4.75, 4.8 , 4.85, 4.9 ,
       4.95, 5.  , 5.05, 5.1 , 5.15, 5.2 , 5.25, 5.3 , 5.35, 5.4 , 5.45,
       5.5 , 5.55, 5.6 , 5.65, 5.7 , 5.75, 5.8 , 5.85, 5.9 , 5.95, 6.  ,
       6.05, 6.1 , 6.15, 6.2 , 6.25])
```

In [49]:
```
y = np.sin(x)
y
```

Out[49]:
```
array([ 0.        ,  0.04997917,  0.09983342,  0.14943813,  0.19866933,
        0.24740396,  0.29552021,  0.34289781,  0.38941834,  0.43496553,
        0.47942554,  0.52268723,  0.56464247,  0.60518641,  0.64421769,
        0.68163876,  0.71735609,  0.75128041,  0.78332691,  0.8134155 ,
        0.84147098,  0.86742323,  0.89120736,  0.91276394,  0.93203909,
        0.94898462,  0.96355819,  0.97572336,  0.98544973,  0.99271299,
        0.99749499,  0.99978376,  0.9995736 ,  0.99686503,  0.99166481,
        0.98398595,  0.97384763,  0.9612752 ,  0.94630009,  0.92895972,
        0.90929743,  0.88736237,  0.86320937,  0.83689879,  0.8084964 ,
        0.7780732 ,  0.74570521,  0.71147335,  0.67546318,  0.6377647 ,
        0.59847214,  0.55768372,  0.51550137,  0.47203054,  0.42737988,
        0.38166099,  0.33498815,  0.28747801,  0.23924933,  0.19042265,
        0.14112001,  0.09146464,  0.04158066, -0.00840725, -0.05837414,
       -0.10819513, -0.15774569, -0.20690197, -0.2555411 , -0.30354151,
       -0.35078323, -0.39714817, -0.44252044, -0.48678665, -0.52983614,
       -0.57156132, -0.61185789, -0.65062514, -0.68776616, -0.72318812,
       -0.7568025 , -0.78852525, -0.81827711, -0.8459837 , -0.87157577,
       -0.89498936, -0.91616594, -0.93505258, -0.95160207, -0.96577306,
       -0.97753012, -0.98684386, -0.993691  , -0.99805444, -0.99992326,
       -0.99929279, -0.99616461, -0.99054654, -0.98245261, -0.97190307,
       -0.95892427, -0.94354867, -0.92581468, -0.90576664, -0.88345466,
       -0.85893449, -0.83226744, -0.80352016, -0.77276449, -0.74007731,
       -0.70554033, -0.66923986, -0.63126664, -0.59171558, -0.55068554,
```

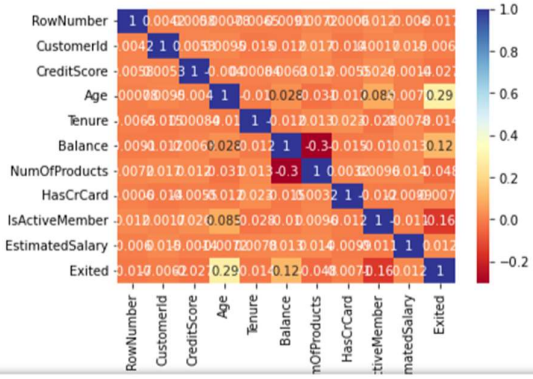
In [50]:
```
plt.plot(x,y)
```

Out[50]:
```
[<matplotlib.lines.Line2D at 0x147565bffd0>]
```



In [46]:
```
sns.heatmap(df.corr(), annot=True, cmap = "RdYlBu")
```

Out[46]:
```
<AxesSubplot:>
```



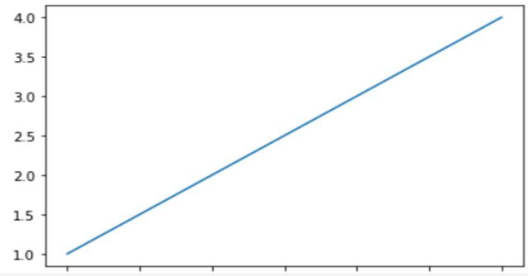
```
In [11]: # Transpose of a martrix
         c = b.T
         c

Out[11]: array([[0, 2, 4, 6, 8],
                [1, 3, 5, 7, 9]])
```

```
In [15]: # creates one-d array and reshapes into 3*3 matrix
         A = np.arange(9).reshape(3,3)
         print(A)
         B = np.arange(9).reshape(3,3)
         print(B)

         [[0 1 2]
          [3 4 5]
          [6 7 8]]
         [[0 1 2]
          [3 4 5]
          [6 7 8]]
```

```
In [16]: D = A+B
         D

Out[16]: array([[ 0,  2,  4],
                [ 6,  8, 10],
                [12, 14, 16]])
```

Matplotlib

```
In [17]: import matplotlib.pyplot as plt
         A = [1,2,3,4]
         plt.plot(A)

Out[17]: [<matplotlib.lines.Line2D at 0x1474dc59250>]
```




```
import pandas as pd
import io

df = pd.read_csv('Bank_churn_modelling.csv')
# df = data frame
# read_csv is a function in pandas
# reads csv file and puts in data frame

df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86
2	3	15619304	Onio	502	France	Female	42	8	159660.80
3	4	15701354	Boni	699	France	Female	39	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82
...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61
9997	9998	15584532	Liu	709	France	Female	36	7	0.00
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79

Numpy

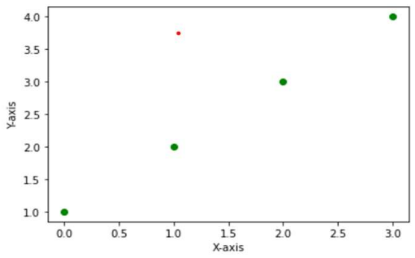
```
In [6]: import numpy as np
        x = np.arange(10)
        print(x)

        type(x)

        [0 1 2 3 4 5 6 7 8 9]

Out[6]: numpy.ndarray
```

```
In [7]: # one-d array
        print(x.shape)

        (10,)
```

```
In [8]: y = x.reshape(1,10)
        y

Out[8]: array([[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]])
```

```
In [9]: # generates 5 rows and 2 columns
        b = x.reshape(5,2)
        b

Out[9]: array([[0, 1],
               [2, 3],
               [4, 5],
               [6, 7],
               [8, 9]])
```

```
In [18]: A = [1,2,3,4]
         # g= green colour, o=plots on line
         plt.plot(A, 'go')
         plt.ylabel('Y-axis')
         plt.xlabel('X-axis')
         plt.show

Out[18]: <function matplotlib.pyplot.show(close=None, block=None)>
```




```
In [2]: import math

        print(math.ceil(4.01))
        print(math.floor(4.09))

        5
        4
```

```
In [3]: a = [1,2,3,4,5]

        print(math.fsum(a))

        15.0
```

```
In [4]: print(math.pow(5, 2))

        25.0
```