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

In [2]:
```
#read csv files
```

In [3]:
```
df = pd.read_csv('50_Startups.csv')
df.head()
```

Out[3]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

In [4]:
```
df.info()
```

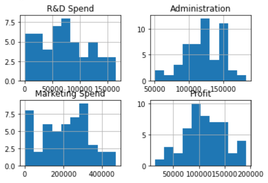
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   R&D Spend        50 non-null     float64
 1   Administration   50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

In [5]:
```
df.isna().sum()
```

Out[5]:
```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

In [6]:
```
# checking data distrubution of each columns
plt.figure(figsize=(10,10))
df.hist()
plt.show()
```

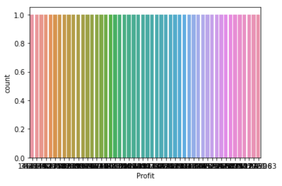
```
<Figure size 720x720 with 0 Axes>
```



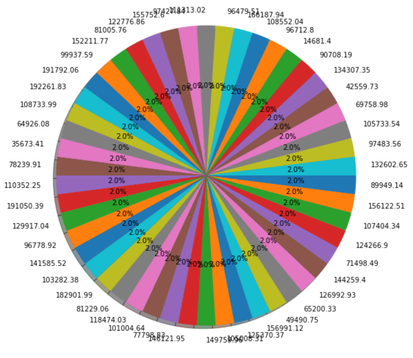
In [7]:
```
# checking target variable
sns.countplot(df['Profit'],data=df)
```

```
C:\Users\Nilesh koli\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[7]:
```
<AxesSubplot:xlabel='Profit', ylabel='count'>
```



In [8]:
```
plt.figure(figsize=(10,10))
plt.pie(df['Profit'].value_counts(),
labels=df['Profit'].value_counts().index, counterclock=False,
shadow=True, autopct='%1.1f%%', radius=1, startangle=0)
plt.show()
```



In [9]:
```
X = df.iloc[:,:-1]
```

In [10]:
```
y = df.iloc[:,4]
```

In [11]:
```
#convert the column into categorical columns
states=pd.get_dummies(X['State'],drop_first=True)
```

In [12]:
```
#Drop the columns
X = X.drop('State',axis=1)
```

In [13]:
```
#Concate the dummy variable
X = pd.concat([X,states],axis=1)
```

In [14]:
```
X.head()
```

Out[14]:

	R&D Spend	Administration	Marketing Spend	Florida	New York
0	165349.20	136897.80	471784.10	0	1
1	162597.70	151377.59	443898.53	0	0
2	153441.51	101145.55	407934.54	1	0
3	144372.41	118671.85	383199.62	0	1
4	142107.34	91391.77	366168.42	1	0

In [15]:
```
#splitting the data into the training set and test set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)
```

In [16]:
```
# fitting multiple linear regression on training dataset
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model
```

Out[16]:
```
LinearRegression()
```

In [17]:
```
model.fit(X_train,y_train)
```

Out[17]:
```
LinearRegression()
```

In [18]:
```
#predict the test set result
y_pred = model.predict(X_test)
y_pred
```

Out[18]:
```
array([104282.76472172, 132536.88499212, 133910.85007766,  72584.77489417,
       179920.9276189 , 114549.31079234,  66444.43261346,  98404.96840122,
       114499.82808602, 169367.50639895,  96522.6253998 ,  88040.6718287 ,
       110949.99405525,  90419.1897851 , 128020.46250064])
```

In [19]:
```
#checking r2_score
from sklearn.metrics import r2_score
print(r2_score(y_test,y_pred))
```

```
0.9358680970046243
```