In [105]:
```
#importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy import math



from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error

import matplotlib.pyplot as plt
```

In [106]:
```
#loading our data
data_frame = pd.read_csv('50_Startups.csv')
data_frame.head(10)
```

Out[106]:

	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
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96

In [107]:
```
len(data_frame)
```

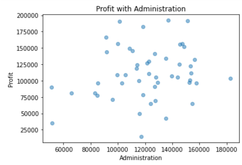
Out[107]:
```
50
```

In [108]:
```
#Data visualization visualizing  profit with  Marketing spend
plt.scatter(data_frame['Marketing Spend'],data_frame['Profit'],alpha=0.5)
plt.title('Profit with Marketing Spend')
plt.xlabel('Marketing Spend')
plt.ylabel('Profit')
plt.show()
```

From above plot we can see that the more the marketing spend the more the profit

In [109]:
```
plt.scatter(data_frame['R&D Spend'],data_frame['Profit'],alpha=0.5)
plt.title('Profit with R&D Spend')
plt.xlabel('R&D spend')
plt.ylabel('Profit')
plt.show()
```

In [110]:
```
plt.scatter(data_frame['Administration'],data_frame['Profit'],alpha=0.5)
plt.title('Profit with Administration')
plt.xlabel('Administration')
plt.ylabel('Profit')
plt.show()
```



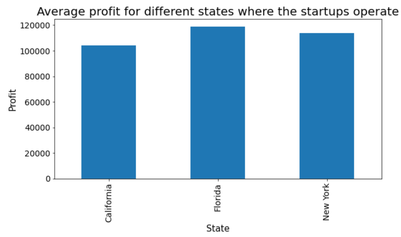
In [111]:
```
# creating the figure object

ax= data_frame.groupby(['State'])['Profit'].mean().plot.bar(figsize=(10,5),fontsize=14)

#setting the title of bargraph

ax.set_title('Average profit for different states where the startups operate',fontsize=20)
ax.set_xlabel('State',fontsize=15)
ax.set_ylabel('Profit',fontsize=15)
```

Out[111]:
```
Text(0, 0.5, 'Profit')
```



From above bargraph we can se ethat the average profit is higher for florida startups

In [112]:
```
data_frame.State.value_counts()
```

Out[112]:
```
New York      17
California    17
Florida       16
Name: State, dtype: int64
```

In [113]:
```
# creating dummy variables for the our categorical variable state
data_frame['NewYork_State']=np.where(data_frame['State']=='New York',1,0)
data_frame['California_State']=np.where(data_frame['State']=='California',1,0)
data_frame['Florida_State']=np.where(data_frame['State']=='Florida',1,0)

#Droppimg the original column state from the dataframe
data_frame.drop(columns=['State'],axis=1,inplace=True)
```

In [114]:
```
 data_frame.head()
```

Out[114]:

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

In [115]:
```
dependent_variable='Profit'
```

In [116]:
```
# create a list of independent variables
independent_variables = data_frame.columns.tolist()
```

In [117]:
```
 independent_variables.remove(dependent_variable)
```

In [118]:
```
#initializing X
X=data_frame[independent_variables].values

#intializing y

y=data_frame[dependent_variable].values
```

In [119]:
```
#splitting the dataset into training and test set
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

In [120]:
```
#Transforming data using min max scaler to get our X variable into same range 0 - 1
scaler= MinMaxScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

In [121]:
```
# Fitting MultiVariate linear regression to the training set
regressor=LinearRegression()
regressor.fit(X_train,y_train)
```

Out[121]:
```
LinearRegression()
```

In [122]:
```
#predicting the test set result
y_pred = regressor.predict(X_test)
```

In [123]:
```
#Evaluation metrics  it gives us how much error is there in an average
math.sqrt(mean_squared_error(y_test,y_pred))
```

Out[123]:
```
9137.990152794944
```

The above code calculate the root mean square error to give average error between our predicted and actual value

In [124]:
```
r2_score(y_test,y_pred)
```

Out[124]:
```
0.9347068473282425
```

r2 is a good way to predict the performance of regression model

from above r2 score we can see that our model has accuracy of 0.93 which is 93%

In [ ]:
```

```