```python
# Multi-Linear Regression on insurance Data

import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

df =
pd.read_csv("/Users/babarhussain/MachineLearning/Multiple_Linear_Regression/MLRcompanies/50_Startups.csv")

df.head()
# check null values
df.isnull().sum()

sns.heatmap(df.isnull())
plt.show()
df.dtypes
# length of dataset
len(df)
df.shape

# lets check the relationship between with other variables

plt.scatter(df['Marketing Spend'], df['Profit'])
plt.title("Profit with marketing Spend")
plt.xlabel("Marketing Spend")
plt.ylabel("profit")
plt.show()

#as per the plot we have linear corelation is good

plt.scatter(df['R&D Spend'], df['Profit'])
plt.title("Profit with R&D Spend")
plt.xlabel("R&D Spend")
plt.ylabel("profit")
plt.show()

# same correlation is good
plt.scatter(df['Administration'], df['Profit'])
plt.title("Profit with Administration")
plt.xlabel("Administration")
plt.ylabel("profit")
plt.show()
# seems to be we don't have any correlation and very little bit have a linear

# Create bar plot of mean values
mean_values = df.groupby('State')['Profit'].mean()
plt.bar(mean_values.index, mean_values)
plt.xlabel('State')
plt.ylabel('Mean Value')
plt.title('Mean Value by State')
plt.show()

# lets see unique values in the state
df["State"].value_counts()
# so we see we have one text value but in MLR we need only continuous or number values
#lets convert this State to numbers

#Create dummy variables for the categorical variable State
df['New York'] = np.where(df["State"]=='New York', 1,0)
df['California'] = np.where(df["State"]=='California', 1,0)
df['Florida'] = np.where(df["State"]=='Florida', 1,0)
print(df)

# lets drop the state column
df.drop(columns=['State'], axis=1, inplace=True)
print(df)

df1 = df.head()
```

```
# Create Dependent variable & Independent varibale

Dependent_Variable = 'Profit'
Independent_Variable = df.columns.tolist()
print(Independent_Variable)
Independent_Variable.remove(Dependent_Variable)
print(Independent_Variable)

X = df[Independent_Variable].values
y = df[Dependent_Variable].values

#lets create & trian the data

x_train,x_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)

# ============================================================================
# # transforming the Data
# # if we see the dataset the values or in 0's & 1's so
# # we do all the values to within the range of 0 t0 1 by using MinMaxScaler
# ============================================================================

scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

# Fitting the MLR to the Training set

model = LinearRegression()
model.fit(x_train,y_train)

# Predict the test set results

y_pred = model.predict(x_test)

# find the meansqured error

math.sqrt(mean_squared_error(y_test, y_pred))


# Find the r2 score

r2_score(y_test,y_pred)

# ============================================================================
# so the r2 score is 93% and mean_squared_error is 91% so this is good model
#
# ============================================================================
```