```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

df = pd.read_csv("/content/Salary_Data.csv")

df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 30,\n  \"fields\": [\n
{\n      \"column\": \"YearsExperience\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.8378881576627184,\n
\"min\": 1.1,\n        \"max\": 10.5,\n        \"num_unique_values\":
28,\n        \"samples\": [\n          3.9,\n          9.6,\n
3.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Salary\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 27414.4297845823,\n        \"min\": 37731.0,\n
\"max\": 122391.0,\n        \"num_unique_values\": 30,\n
\"samples\": [\n          112635.0,\n          67938.0,\n
113812.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
df.tail()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"YearsExperience\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.6140032573203502,\n
\"min\": 9.0,\n        \"max\": 10.5,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          9.5,\n          10.5,\n
9.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Salary\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 7001.097321134738,\n        \"min\": 105582.0,\n
\"max\": 122391.0,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          116969.0,\n          121872.0,\n
112635.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
df.isnull().sum()
```

```
YearsExperience    0
Salary             0
dtype: int64
```

```
df.info()
```

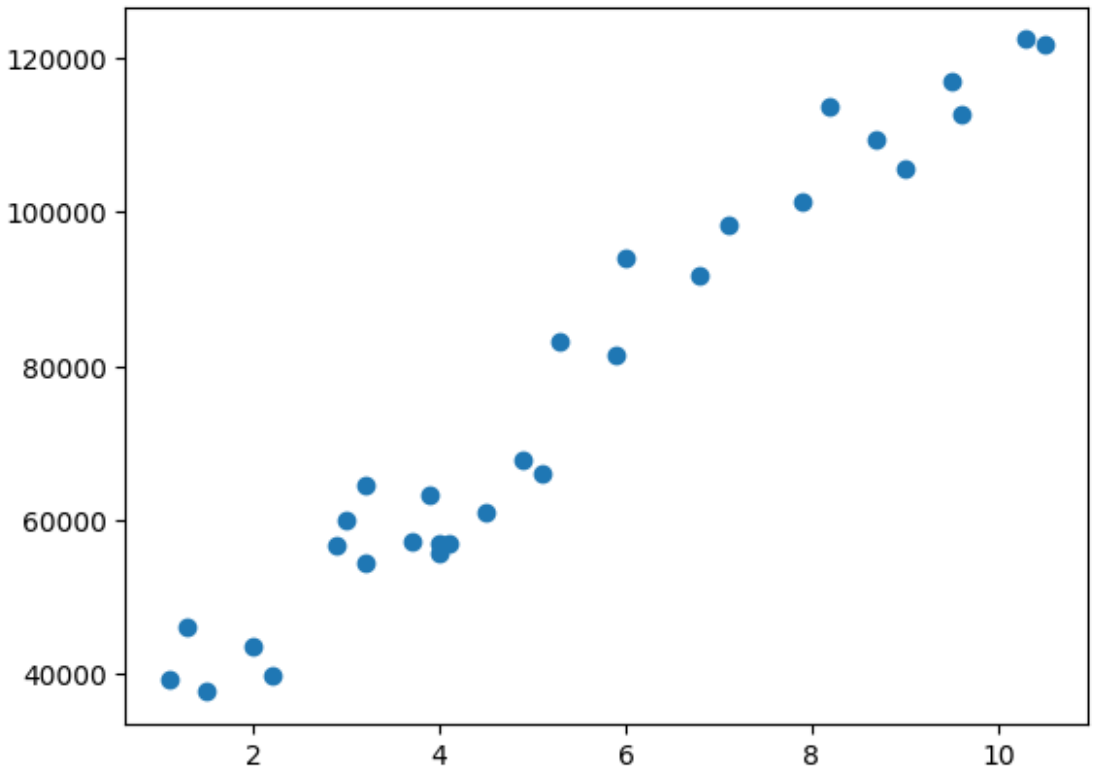
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
```

```
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

```
df.describe()
```

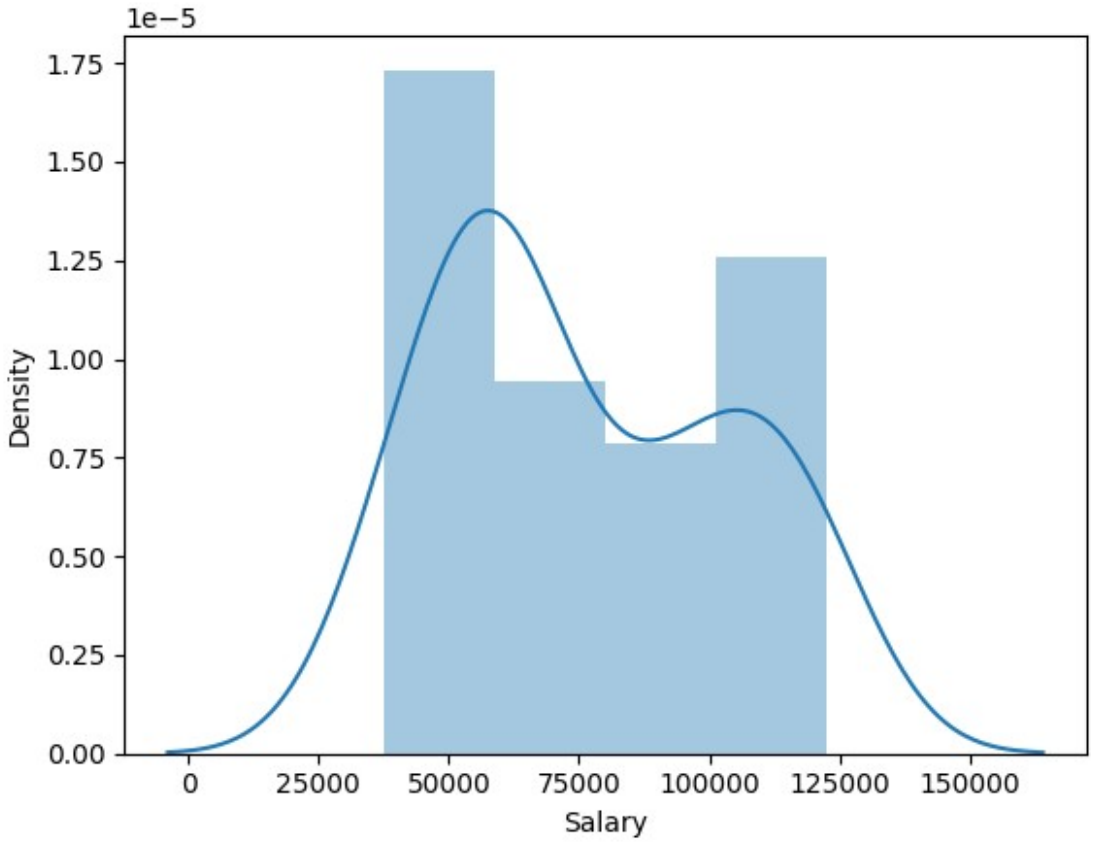

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n{\n      \"column\": \"YearsExperience\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 9.300670878343443,\n\"min\": 1.1,\n        \"max\": 30.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          5.3133333333333335,\n4.7,\n          30.0\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Salary\",\n      \"properties\": {\n        \"dtype\": \"number\",\n\"std\": 39605.7524645371,\n        \"min\": 30.0,\n        \"max\": 122391.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n76003.0,\n          65237.0,\n          30.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
plt.scatter(df.YearsExperience, df.Salary)
plt.show()
```



```
sns.distplot(df['Salary'])
```

```
<ipython-input-54-bc20e5e6d548>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Salary'])

<Axes: xlabel='Salary', ylabel='Density'>
```




```
X = df.drop('Salary', axis=1)
y = df['Salary']

X
```

{"summary":"{\n \"name\": \"X\",\n \"rows\": 30,\n \"fields\": [\n{\n   \"column\": \"YearsExperience\",\n   \"properties\": {\n\"dtype\": \"number\",\n   \"std\": 2.8378881576627184,\n\"min\": 1.1,\n   \"max\": 10.5,\n   \"num_unique_values\": 28,\n   \"samples\": [\n   3.9,\n   9.6,\n   3.7\n   ],\n   \"semantic_type\": \"\",\n\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"X"}

```
y
```

```
0      39343.0
1      46205.0
2      37731.0
3      43525.0
4      39891.0
5      56642.0
6      60150.0
7      54445.0
8      64445.0
9      57189.0
10     63218.0
11     55794.0
12     56957.0
13     57081.0
14     61111.0
15     67938.0
16     66029.0
17     83088.0
18     81363.0
19     93940.0
20     91738.0
21     98273.0
22    101302.0
23    113812.0
24    109431.0
25    105582.0
26    116969.0
27    112635.0
28    122391.0
29    121872.0
Name: Salary, dtype: float64
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

```
X_train.shape
```

```
(24, 1)
```

```
X_test.shape
```

```
(6, 1)
```

```
y_train.shape
```

```
(24,)
```

```
y_test.shape
```

```
(6,)
```

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

```
LinearRegression()
```

```
print(lm.intercept_)
```

```
27374.91926794422
```

```
lm.coef_
```

```
array([9325.05247783])
```

```
predictions = lm.predict(X_test)
```

```
plt.scatter(X_test, y_test, color = 'lightcoral')
plt.plot(X_test, predictions, color = 'firebrick')
plt.title('Salary vs Experience (Test Set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.box(False)
plt.show()
```


```
data = pd.read_csv("/content/Housing - area.csv")

data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 545,\n  \"fields\":
[\n    {\n      \"column\": \"price\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1870439,\n        \"min\":
1750000,\n        \"max\": 13300000,\n        \"num_unique_values\":
219,\n        \"samples\": [\n          3773000,\n          5285000,\n
1820000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"area\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 2170,\n        \"min\": 1650,\n        \"max\": 16200,\n
\"num_unique_values\": 284,\n        \"samples\": [\n          6000,\n
2684,\n          5360\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


```
data.tail()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 5,\n  \"fields\": [\
n    {\n      \"column\": \"price\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 30311,\n        \"min\":
1750000,\n        \"max\": 1820000,\n        \"num_unique_values\":
```

```
3,\n        \"samples\": [\n          1820000,\n          1767150,\n
1750000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"area\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 581,\n        \"min\": 2400,\n        \"max\": 3850,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          2400,\n
3850,\n          3620\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

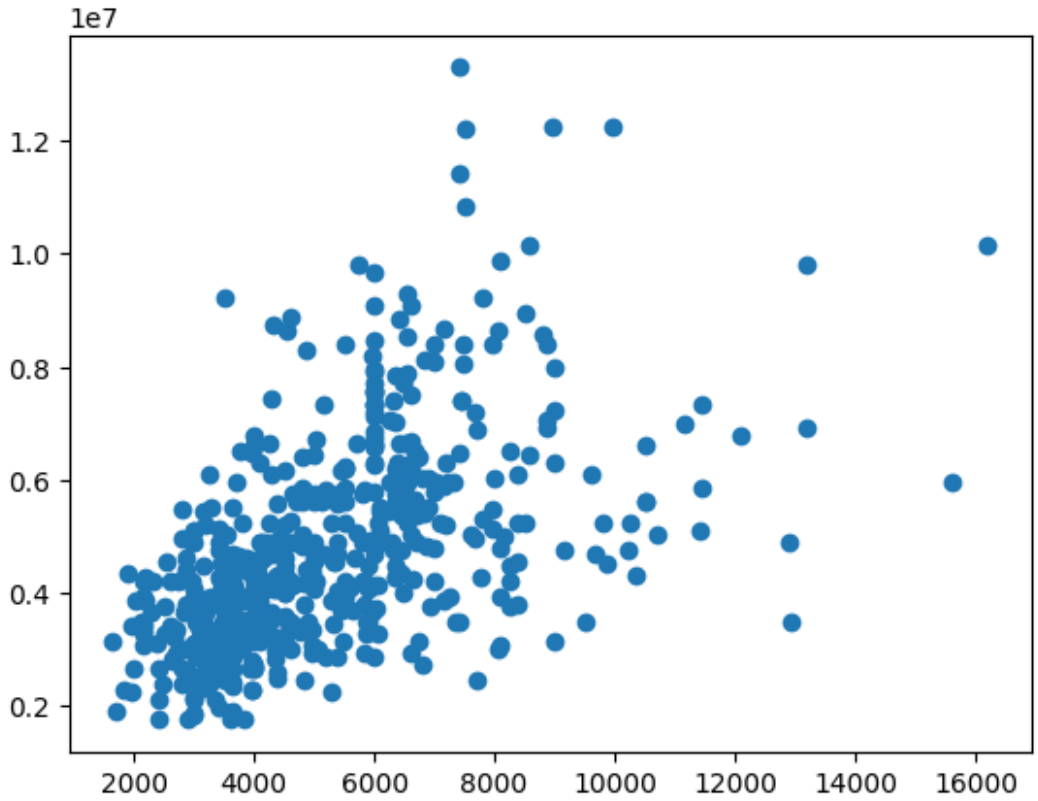

```
data.isnull().sum()
```

```
price    0
area     0
dtype: int64
```

```
data.info()
```

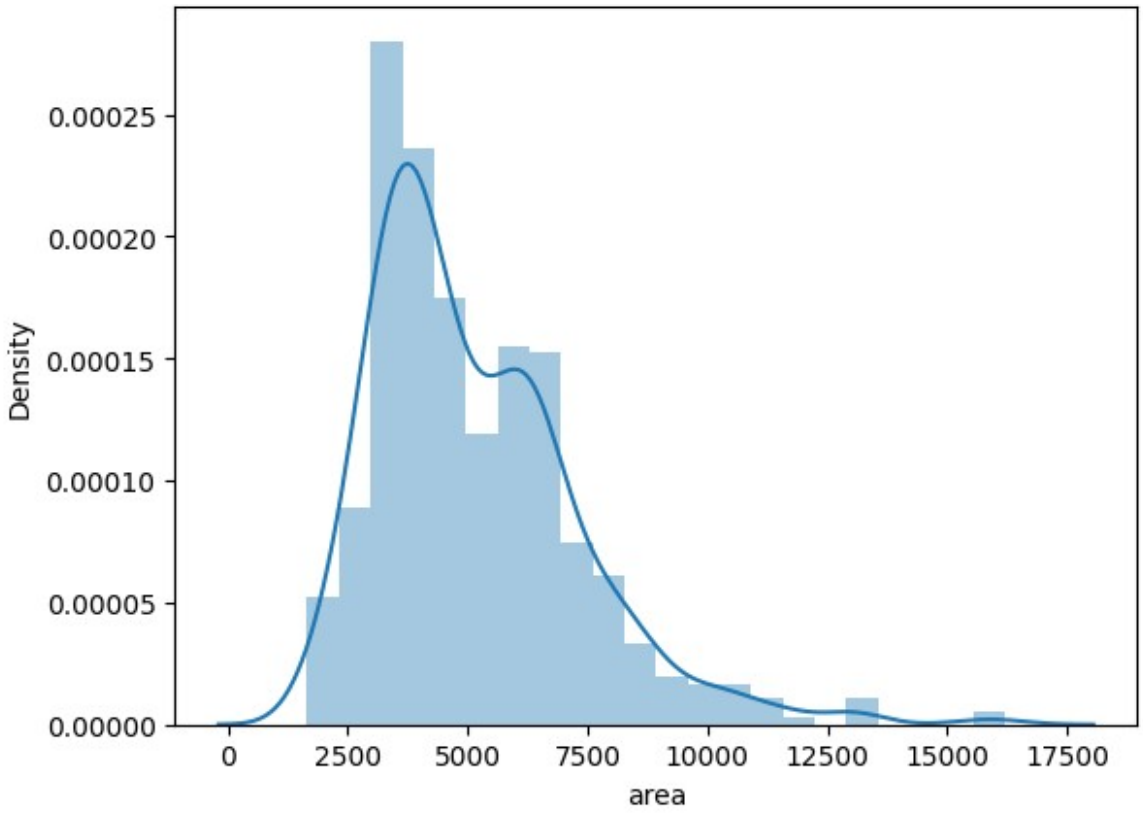
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   price   545 non-null    int64
 1   area    545 non-null    int64
dtypes: int64(2)
memory usage: 8.6 KB
```

```
plt.scatter(data.area, data.price)
plt.show()
```




```
sns.distplot(data['area'])
```

```
<ipython-input-74-b721d5f339fd>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['area'])

<Axes: xlabel='area', ylabel='Density'>
```




```
X = data.drop('area', axis=1)
y = data['area']
```

```
X
```

```
{"summary":"{\n  \"name\": \"X\",\n  \"rows\": 545,\n  \"fields\": [\n
{\n      \"column\": \"price\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1870439,\n        \"min\":
1750000,\n        \"max\": 13300000,\n        \"num_unique_values\":
219,\n        \"samples\": [\n          3773000,\n          5285000,\n
1820000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"X"}
```

```
y
```

```
0      7420
1      8960
2      9960
3      7500
4      7420
      ...
540    3000
541    2400
542    3620
```

```
543    2910
544    3850
Name: area, Length: 545, dtype: int64
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2)
```

```
X_train.shape
```

```
(436, 1)
```

```
X_test.shape
```

```
(109, 1)
```

```
y_train.shape
```

```
(436,)
```

```
y_test.shape
```

```
(109,)
```

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

```
LinearRegression()
```

```
print(lm.intercept_)
```

```
2064.600522705819
```

```
lm.coef_
```

```
array([0.0006556])
```

```
predictions = lm.predict(X_test)
```

```
plt.scatter(X_test, y_test, color = 'lightcoral')
plt.plot(X_test, predictions, color = 'firebrick')
plt.title('Price vs Area (Test Set)')
plt.xlabel('Area')
plt.ylabel('Price')
plt.box(False)
plt.show()
```


```
df1 = pd.read_csv("/content/50_Startups.csv")

df1.head()
```


```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 50,\n  \"fields\": [\n    {\n      \"column\": \"R&D Spend\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 45902.25648230753,\n\"min\": 0.0,\n        \"max\": 165349.2,\n\"num_unique_values\": 49,\n        \"samples\": [\n91992.39,\n          1000.23,\n          0.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Administration\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 28017.802755488683,\n        \"min\": 51283.14,\n        \"max\": 182645.56,\n        \"num_unique_values\": 50,\n        \"samples\": [\n          135495.07,\n          82982.09,\n          115641.28\n],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n}\n    },\n    {\n      \"column\": \"Marketing Spend\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 122290.31072584528,\n        \"min\": 0.0,\n        \"max\": 471784.1,\n        \"num_unique_values\": 48,\n        \"samples\": [\n          353183.81,\n          172795.67,\n          134050.07\n],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
```

```
}\n    },\n    {\n      \"column\": \"State\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"New York\",\n
\"California\",\n          \"Florida\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Profit\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
40306.18033765055,\n        \"min\": 14681.4,\n        \"max\":
192261.83,\n        \"num_unique_values\": 50,\n        \"samples\":
[\n          134307.35,\n          81005.76,\n          99937.59\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df1"}
```


```
df1.tail()
```


```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"R&D Spend\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 589.7833681700425,\n
\"min\": 0.0,\n        \"max\": 1315.46,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1315.46,\
n          542.05,\n          1000.23\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Administration\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
32849.625535718515,\n        \"min\": 51743.15,\n        \"max\":
135426.92,\n        \"num_unique_values\": 5,\n        \"samples\": [\
n          115816.21,\n          116983.8,\n          135426.92\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Marketing Spend\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
129061.69240878528,\n        \"min\": 0.0,\n        \"max\":
297114.46,\n        \"num_unique_values\": 4,\n        \"samples\": [\
n          297114.46,\n          45173.06,\n          1903.93\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"State\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"New York\",\n          \"Florida\",\n
\"California\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Profit\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 18486.05628017047,\n        \"min\": 14681.4,\n
\"max\": 64926.08,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          49490.75,\n          14681.4,\n
42559.73\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
df1.isnull().sum()
```

```
R&D Spend          0
Administration     0
```

```
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

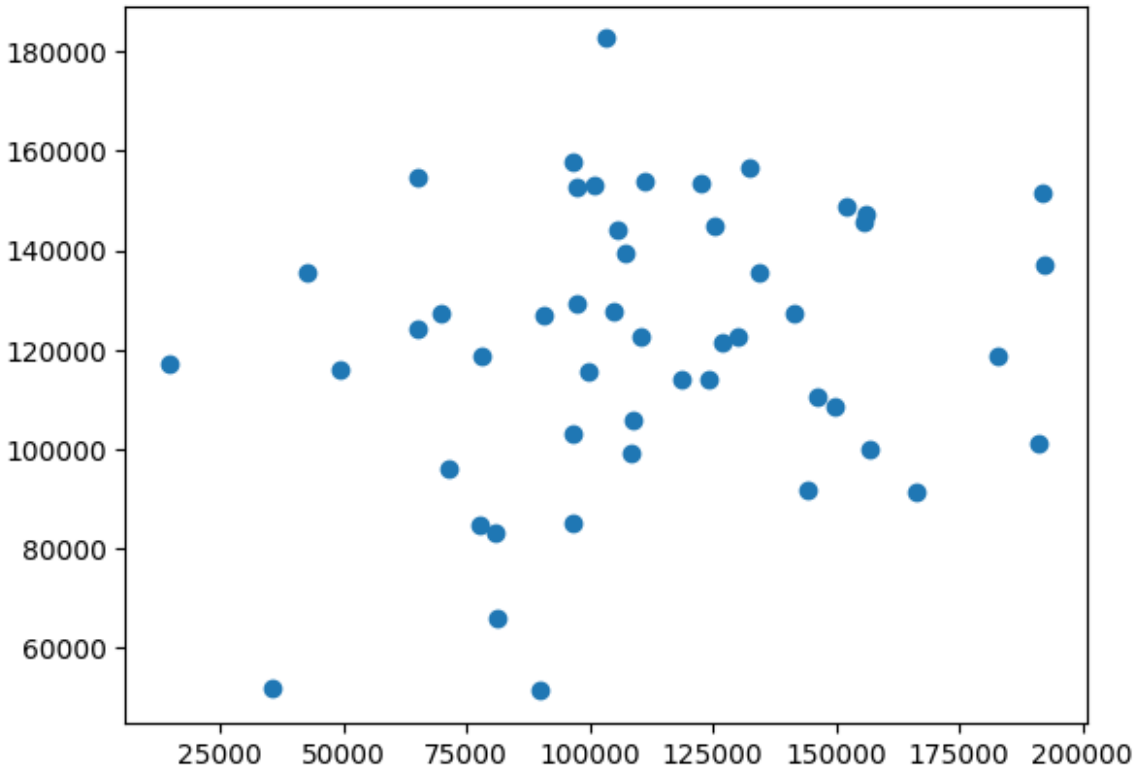
```
df1.info()
```

```
<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
```

```
df1.describe()
```

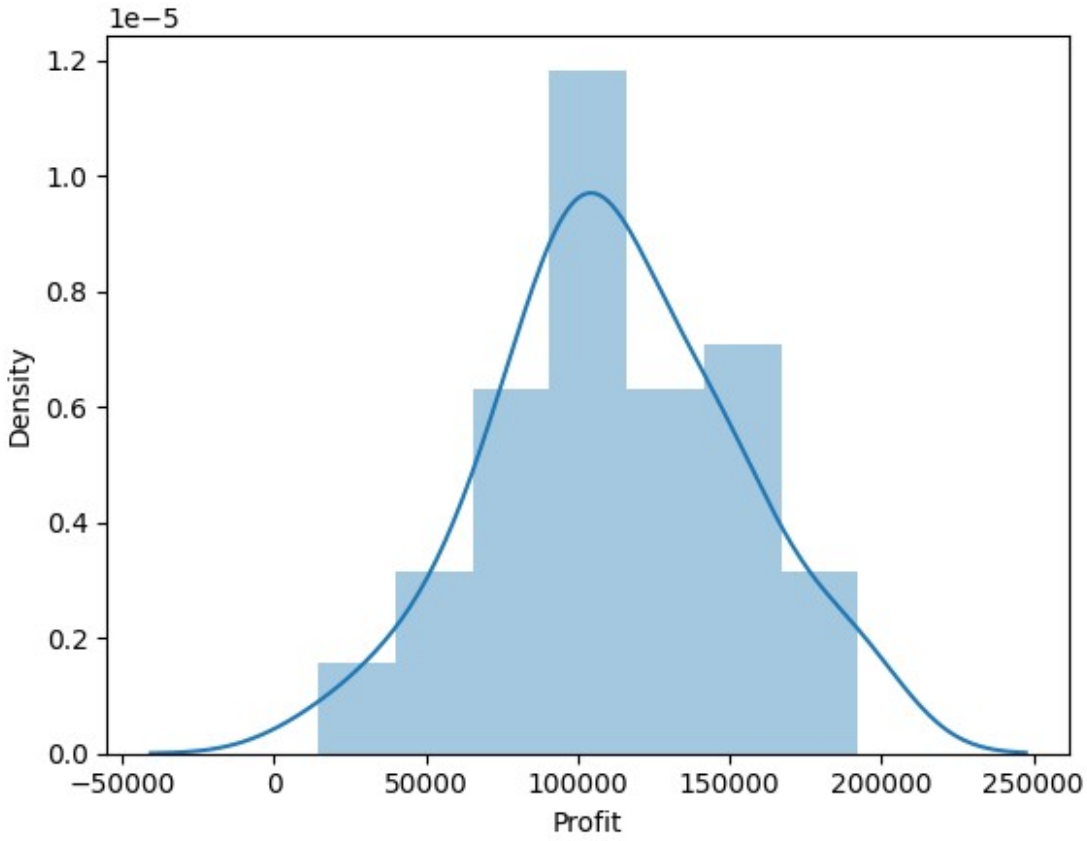

```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 8,\n  \"fields\": [\n{\n      \"column\": \"R&D Spend\",\n      \"properties\": {\n\"dtype\": \"number\",\n        \"std\": 54687.51901616005,\n\"min\": 0.0,\n        \"max\": 165349.2,\n\"num_unique_values\": 8,\n        \"samples\": [\n73721.6156,\n          73051.08,\n          50.0\n        ],\n\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Administration\",\n\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 62235.943809479024,\n        \"min\": 50.0,\n        \"max\": 182645.56,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          121344.63960000001,\n          122699.795,\n          50.0\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Marketing Spend\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 156807.9429432482,\n        \"min\": 0.0,\n        \"max\": 471784.1,\n        \"num_unique_values\": 8,\n\"samples\": [\n          211025.09780000002,\n          212716.24,\n50.0\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Profit\",\n      \"properties\": {\n        \"dtype\": \"number\",\n\"std\": 65367.40907318825,\n        \"min\": 50.0,\n        \"max\": 192261.83,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          112012.63920000002,\n          107978.19,\n          50.0\n],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n}\n    }\n  ]\n}","type":"dataframe"}
```

```
plt.scatter(df1.Profit, df1.Administration)
plt.show()
```




```
sns.distplot(df1['Profit'])
```

```
<ipython-input-95-34c9eb850367>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df1['Profit'])

<Axes: xlabel='Profit', ylabel='Density'>
```




```
X = df1[['Administration','R&D Spend', 'Marketing Spend', 'State']]
y = df1['Profit']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2)

X_train.shape
```

```
(40, 4)
```

```
X_test.shape
```

```
(10, 4)
```

```
y_train.shape
```

```
(40,)
```

```
y_test.shape
```

```
(10,)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X_train['State'] = le.fit_transform(X_train['State'])
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X_test['State'] = le.fit_transform(X_test['State'])
```

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
```

```
LinearRegression()
```

```
print(lm.intercept_)
```

```
50164.9102758657
```

```
lm.coef_
```

```
array([-3.59471995e-02,  8.01864717e-01,  3.02185516e-02,
4.20525816e+02])
```

```
y_pred = lm.predict(X_test)
```

```
from sklearn.metrics import r2_score
score = r2_score(y_test, y_pred)
score
```

```
0.9411683549248834
```

```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 7239.607924307934
MSE: 78184883.48231527
RMSE: 8842.221637253573
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```