```
#Assignment 1
#Simple linear regress
# Amit pawar
# PRN 25070126501
#Medical Insurance Cost Prediction data set.
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_fwf(
    '/content/medical-charges.csv',
    colspecs='infer',
    header=None
)
df.head()
```

	0
0	age,sex,bmi,children,smoker,region,charges
1	19,female,27.9,0,yes,southwest,16884.924
2	18,male,33.77,1,no,southeast,1725.5523
3	28,male,33,3,no,southeast,4449.462
4	33,male,22.705,0,no,northwest,21984.47061

Next steps: Generate code with df | New interactive sheet

```
df = pd.read_csv('medical-charges.csv')
print(df.columns)
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

```
print(df.shape)
```

```
(1338, 7)
```

```
df = df[['sex', 'children']]
```

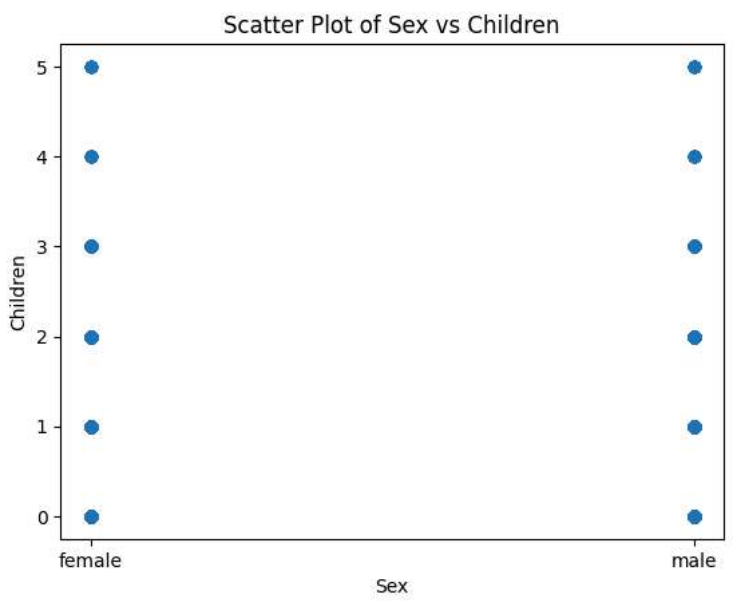
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   sex       1338 non-null   object
 1   children  1338 non-null   int64
dtypes: int64(1), object(1)
memory usage: 21.0+ KB
```

```
df.describe()
```

	children
count	1338.000000
mean	1.094918
std	1.205493
min	0.000000
25%	0.000000
50%	1.000000
75%	2.000000
max	5.000000

```
plt.scatter(df['smoker'], df['children'])
plt.xlabel('smoker')
plt.ylabel('Children')
plt.title('Scatter Plot of smoker vs Children')

plt.show()
```




```
x = df_full['age']
y = df_full['charges']
```

```
print(x.shape)
print(y.shape)
```

```
(1338,)
(1338,)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test = train_test_split(x, y, test_size=0.20,random_state=42)
```

```
print(X_train.shape)
print(X_train.shape)
print(y_train.shape)
print(y_train.shape)
```

```
(1070,)
(1070,)
(1070,)
(1070,)
```

```
from sklearn.linear_model import LinearRegression
ln = LinearRegression()
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
df = pd.read_csv('medical-charges.csv')

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

cat_cols = ['sex', 'smoker', 'region']

ct = ColumnTransformer(
    [('encode', OneHotEncoder(drop='first'), cat_cols)],
    remainder='passthrough'
)

X = ct.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

y_pred = model.predict(X_test)
print("R2 Score:", r2_score(y_test, y_pred))
```

```
R2 Score: 0.7835929767120722
```

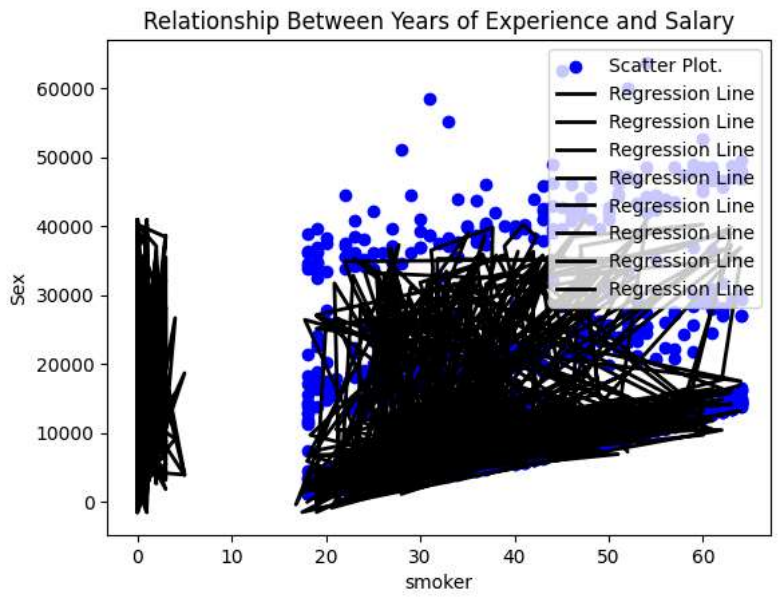
LinearRegression() In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

LinearRegression ?Documentation for LinearRegressioniFitted LinearRegression() **bold text**

```
ln = LinearRegression()
ln.fit(X_train, y_train)

print("Slope (a):", ln.coef_)
print("Intercept (b):", ln.intercept_)
```

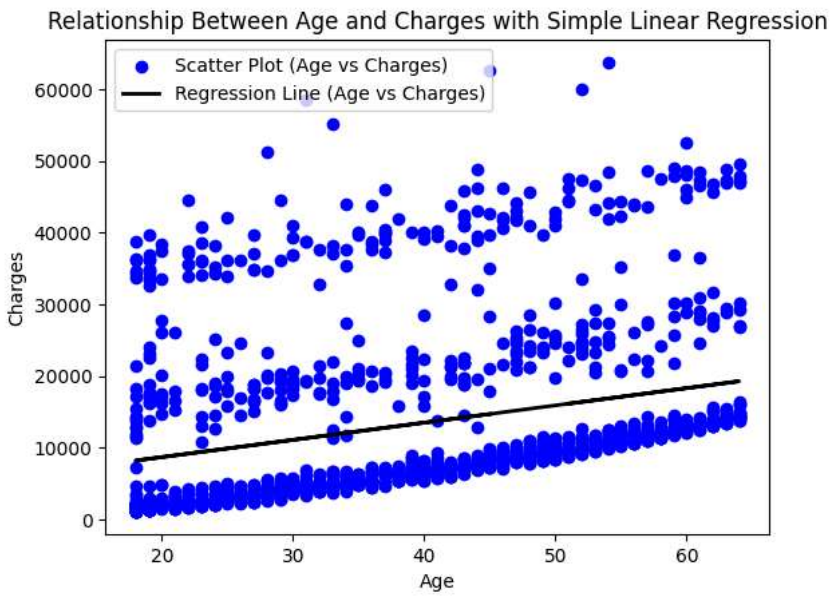
```
Slope (a): [-1.85916916e+01  2.36511289e+04 -3.70677326e+02 -6.57864297e+02
 -8.09799354e+02  2.56975706e+02  3.37092552e+02  4.25278784e+02]
Intercept (b): -11931.219050326445
```

```
plt.scatter(x, y, color='blue', label="Scatter Plot.")
plt.plot(X_test, y_pred, color='black', linewidth=2, label='Regression Line')
plt.title("Relationship Between Years of Experience and Salary")
plt.xlabel('smoker')
plt.ylabel('Sex')
plt.legend()
plt.show()
```




```
# The original x (age) and y (charges) Series are available from previous steps.

x_train_simple, x_test_simple, y_train_simple, y_test_simple = train_test_split(x, y, test_size=0.20, random_state=42)

x_train_simple_reshaped = x_train_simple.values.reshape(-1, 1)
x_test_simple_reshaped = x_test_simple.values.reshape(-1, 1)
simple_ln = LinearRegression()
simple_ln.fit(x_train_simple_reshaped, y_train_simple)
```

```
simple_ln.fit(x_train_simple_reshaped, y_train_simple)
y_pred_simple = simple_ln.predict(x_test_simple_reshaped)

plt.scatter(x, y, color='blue', label="Scatter Plot (Age vs Charges)")

plt.plot(x_test_simple_reshaped, y_pred_simple, color='black', linewidth=2, label='Regression Line (Age vs Charges)')

plt.title("Relationship Between Age and Charges with Simple Linear Regression")
plt.xlabel('Age')
plt.ylabel('Charges')
plt.legend()
plt.show()
```



Start coding or generate with AI.