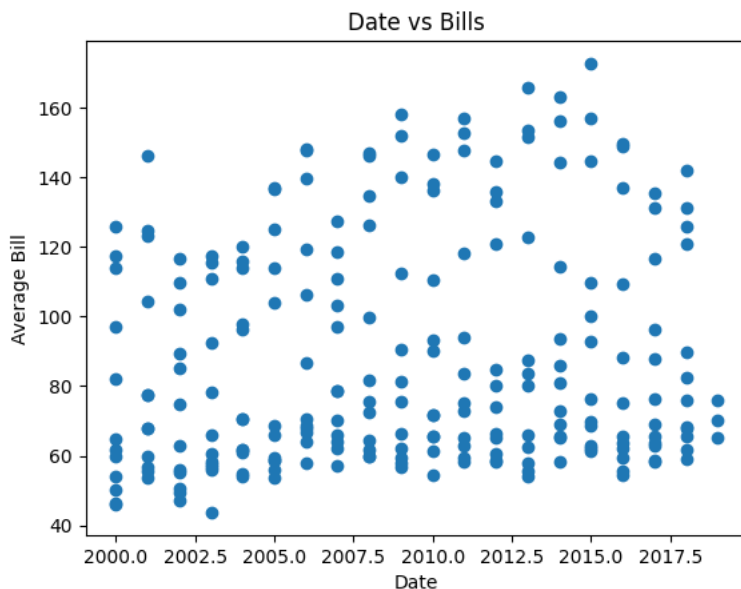


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
df=pd.read_csv('/content/Residential_Average_Monthly_kWh_and_Bills.csv')
plt.scatter(df['Date'],df['Average Bill'])
plt.title("Date vs Bills")
plt.xlabel('Date')
plt.ylabel('Average Bill')
plt.show()
```



```
X = df.iloc[:, :1] # independent
print(X)
y = df.iloc[:, 3:] # dependent
print(y)
```

```

      Date
0    2000
1    2000
2    2000
3    2000
4    2000
..     ...
226  2018
227  2018
228  2019
229  2019
230  2019
```

```

[231 rows x 1 columns]
Average Bill
0         54.26
1         50.27
2         45.91
3         46.50
4         61.68
..         ...
226        68.01
227        65.67
228        76.13
229        70.38
230        65.13
```

```
[231 rows x 1 columns]
```

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

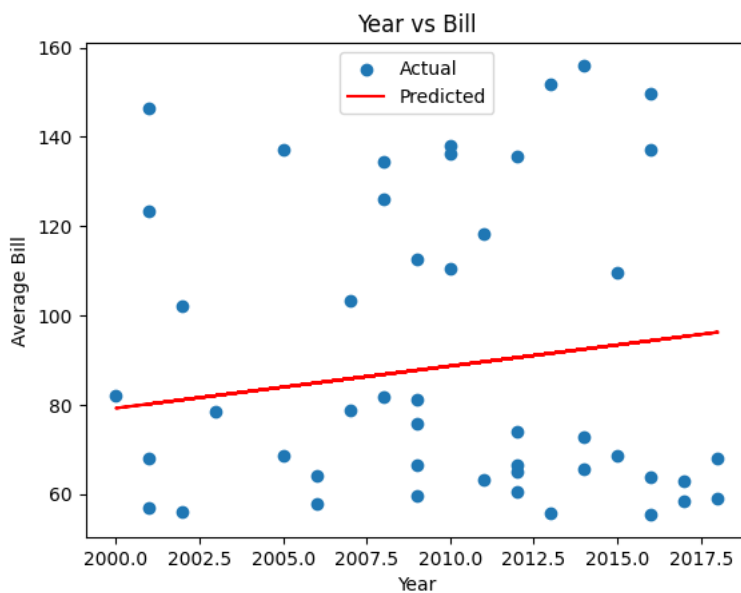
```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▼ LinearRegression  
LinearRegression()
```

```
predictions = model.predict(X_test)  
mse=mean_squared_error(y_test,predictions)  
r2=r2_score(y_test,predictions)  
print(mse)  
print(r2)
```

```
1116.3523203122834  
-0.04284539303833679
```

```
plt.scatter(X_test,y_test,label='Actual')  
plt.plot(X_test,predictions,color='red',label='Predicted')  
plt.title('Year vs Bill')  
plt.xlabel('Year')  
plt.ylabel('Average Bill')  
plt.legend()  
plt.show()
```



```
predictions = model.predict([[2025]])  
predictions
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression  
warnings.warn(  
array([[102.85302122]])
```