In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv("insurance.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [4]:

```
df.shape
```

Out[4]:

```
(1338, 7)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [6]:

```
df.describe()
```

Out[6]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [7]:

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

Out[7]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [8]:

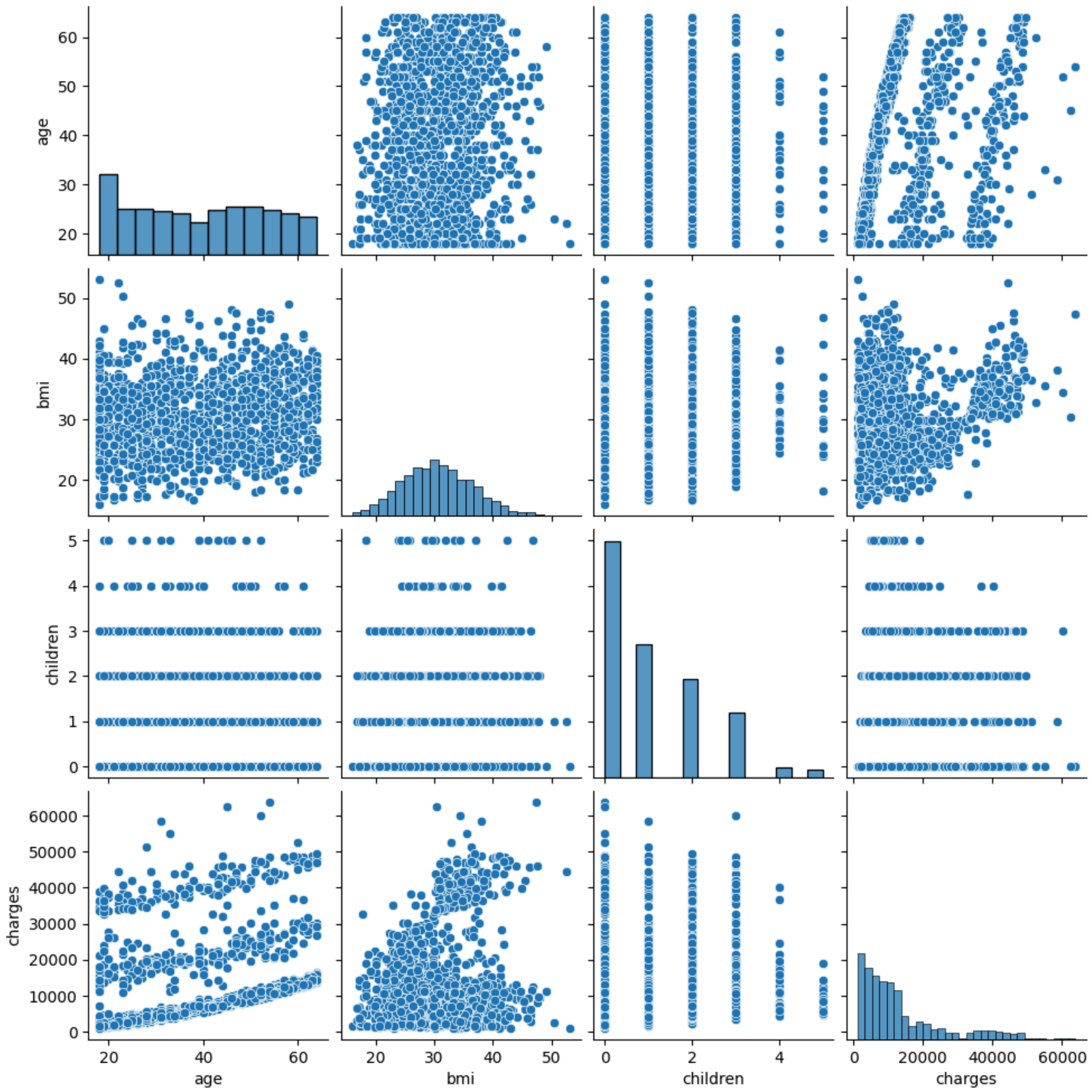
```
import seaborn as sns
```

In [9]:

```
sns.pairplot(df)
```

Out[9]:

```
<seaborn.axisgrid.PairGrid at 0x2439f32aba0>
```



In [14]:
```
from sklearn.preprocessing import LabelEncoder
```

In [17]:
```
le= LabelEncoder()
```

In [18]:
```
df["sex"] = le.fit_transform(df["sex"])
df["smoker"]=le.fit_transform(df["smoker"])
df["region"]=le.fit_transform(df["region"])
```

In [19]:
```
df.head()
```

Out[19]:

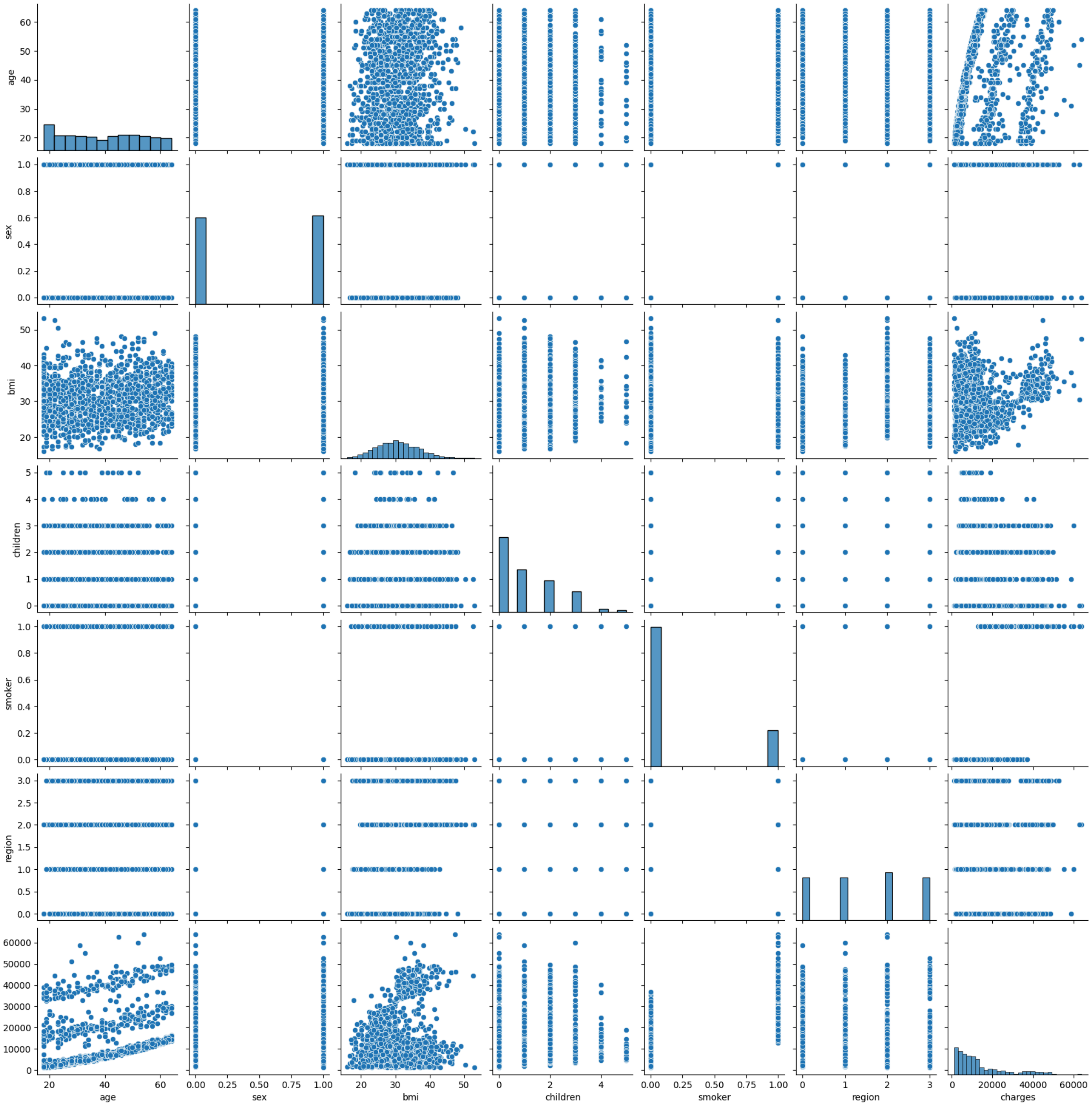
	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

In [20]:

```
sns.pairplot(df)
```

Out[20]:

```
<seaborn.axisgrid.PairGrid at 0x243a5f9da90>
```

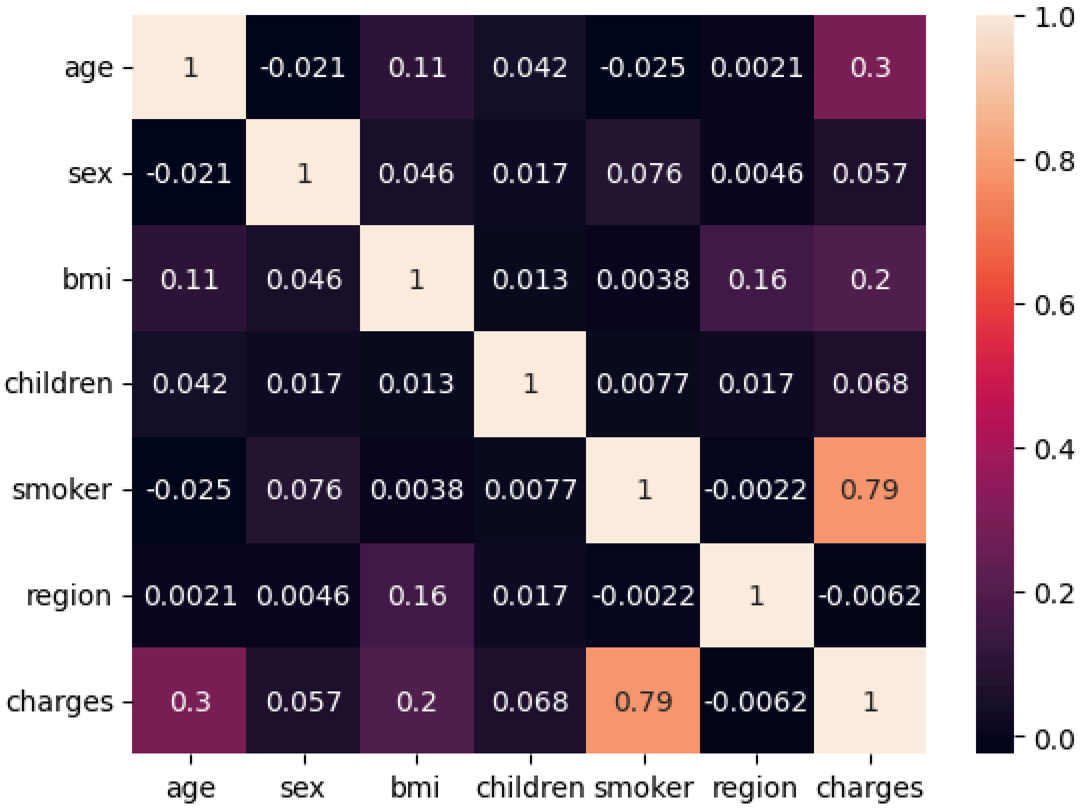


In [21]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[21]:

```
<Axes: >
```



In [22]:
```
x=df[['age']]
y=df[['charges']]
```

In [23]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_st
```

In [24]:
```
x_train
```

Out[24]:

	age
693	24
1297	28
634	51
1022	47
178	46
...	...
1095	18
1130	39
1294	58
860	37
1126	55

1003 rows × 1 columns

In [26]:

```
from sklearn.linear_model import LinearRegression
```

In [27]:

```
lr=LinearRegression()
```

In [29]:

```
model=lr.fit(x_train,y_train)
```

In [40]:

```
y_predict1=model.predict(x_test)
```

In [45]:

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data
y_predict_train=model.predict(x_train)
mse_train1=mean_squared_error(y_train, y_predict_train)

rmse_train1=np.sqrt(mse_train1)
r2_train1=r2_score(y_train,y_predict_train)
print(f"the mse for traning1:{mse_train1}\nthe rmse for traning1 : {rmse_train1}\nt
```

```
the mse for traning1:132597611.08057606
the rmse for traning1 : 11515.103607027428
the r2score for traning1: 0.08610344496017153
```

In [ ]:

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data

mse_test1=mean_squared_error(y_test, y_predict1)
rmse_test1=np.sqrt(mse_test1)
```

```
r2_test1=r2_score(y_test,y_predict1)
print(f"the mse for testing1:{mse_test1}\nthe rmse for testing1 : {rmse_test1}\nthe
```

```
the mse for testing1:135993724.9234396
 the rmse for testing1 : 11661.634744899173
 the r2score for testing1: 0.09872955304263209
```

In [48]:

```
x1=df[['smoker']]
y=df[['charges']]
```

In [49]:

```
from sklearn.model_selection import train_test_split
x_train1, x_test1, y_train1, y_test1 = train_test_split(x1, y, test_size=0.25, rand
```

In [50]:

```
model1=lr.fit(x_train1,y_train1)
```

In [55]:

```
y_predict2=model1.predict(x_test1)
```

In [56]:

```
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data
y_predict_train2=model1.predict(x_train1)
mse_train2=mean_squared_error(y_train1, y_predict_train2)

rmse_train2=np.sqrt(mse_train2)
r2_train2=r2_score(y_train1,y_predict_train2)
print(f"the mse for traning2:{mse_train2}\nthe rmse for traning2 : {rmse_train2}\nt
```

```
the mse for traning2:56368544.2961563
the rmse for traning2 : 7507.898793680979
the r2score for traning2: 0.6114936157216068
```

In [ ]:

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data

mse_test2=mean_squared_error(y_test1, y_predict2)
rmse_test2=np.sqrt(mse_test2)
r2_test2=r2_score(y_test1,y_predict2)
print(f"the mse for testing2:{mse_test2}\nthe rmse for testing2 : {rmse_test2}\nthe
```

```
the mse for testing2:53840720.19066271
the rmse for testing2 : 7337.623606499771
the r2score for testing1: 0.6431816984345173
```

In [59]:

```
x2=df.drop('charges',axis=1)
y=df[['charges']]
x2
```

Out[59]:

	age	sex	bmi	children	smoker	region
0	19	0	27.900	0	1	3
1	18	1	33.770	1	0	2
2	28	1	33.000	3	0	2
3	33	1	22.705	0	0	1
4	32	1	28.880	0	0	1
...	...	...	...	...	...	...
1333	50	1	30.970	3	0	1
1334	18	0	31.920	0	0	0
1335	18	0	36.850	0	0	2
1336	21	0	25.800	0	0	3
1337	61	0	29.070	0	1	1

1338 rows × 6 columns

In [60]:

```
x_train2, x_test2, y_train2, y_test2 = train_test_split(x2, y, test_size=0.25, rand
```

In [61]:

```
model2=lr.fit(x_train2,y_train2)
```

In [64]:

```
y_predict3=model2.predict(x_test2)
```

In [65]:

```
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data
y_predict_train3=model2.predict(x_train2)
mse_train3=mean_squared_error(y_train2, y_predict_train3)

rmse_train3=np.sqrt(mse_train3)
r2_train3=r2_score(y_train2,y_predict_train3)
print(f"the mse for traning3:{mse_train3}\nthe rmse for traning3 : {rmse_train3}\nt
```

```
the mse for traning3:37011292.58315399
the rmse for traning3 : 6083.690704100102
the r2score for traning3: 0.7449087316606229
```

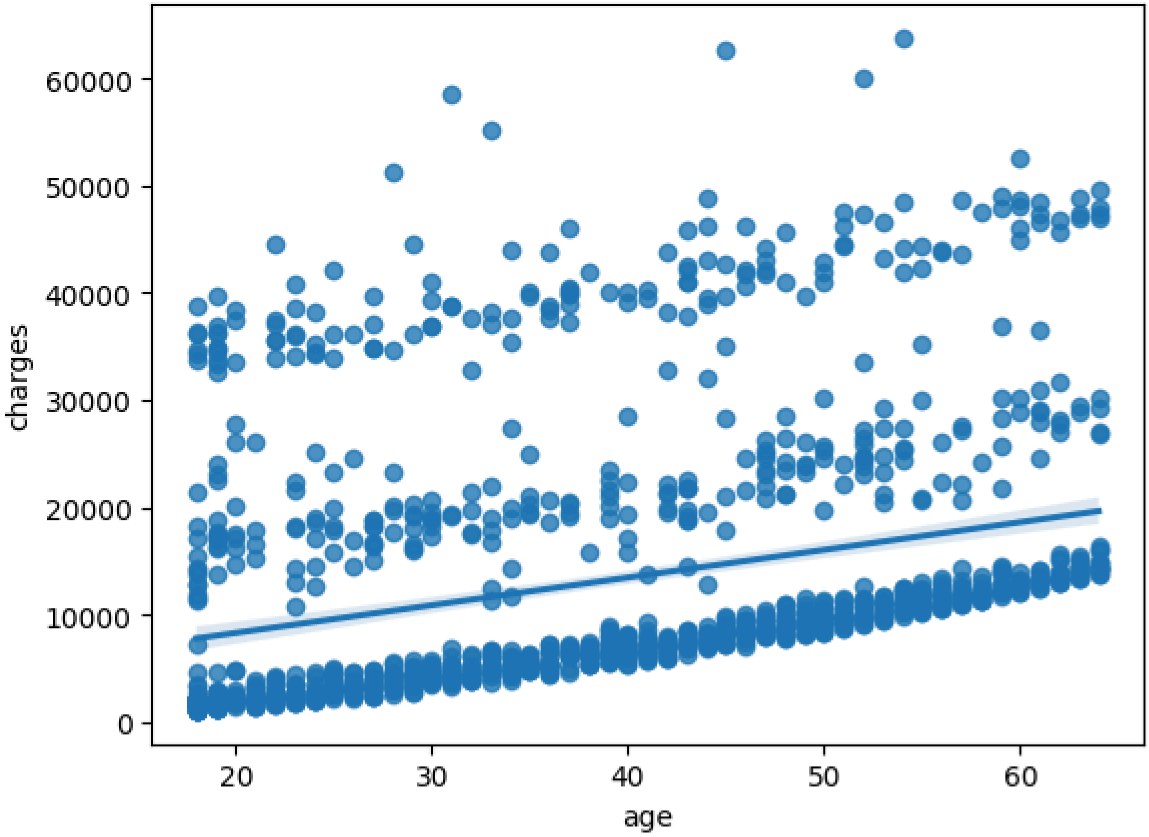
In [66]:

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import numpy as np
# prediction on traning data

mse_test3=mean_squared_error(y_test2, y_predict3)
rmse_test3=np.sqrt(mse_test3)
r2_test3=r2_score(y_test2,y_predict3)
print(f"the mse for testing3:{mse_test3}\nthe rmse for testing3 : {rmse_test3}\nthe
the mse for testing3:35174149.32705306
the rmse for testing3 : 5930.779824530081
the r2score for testing3: 0.7668905583460908
```

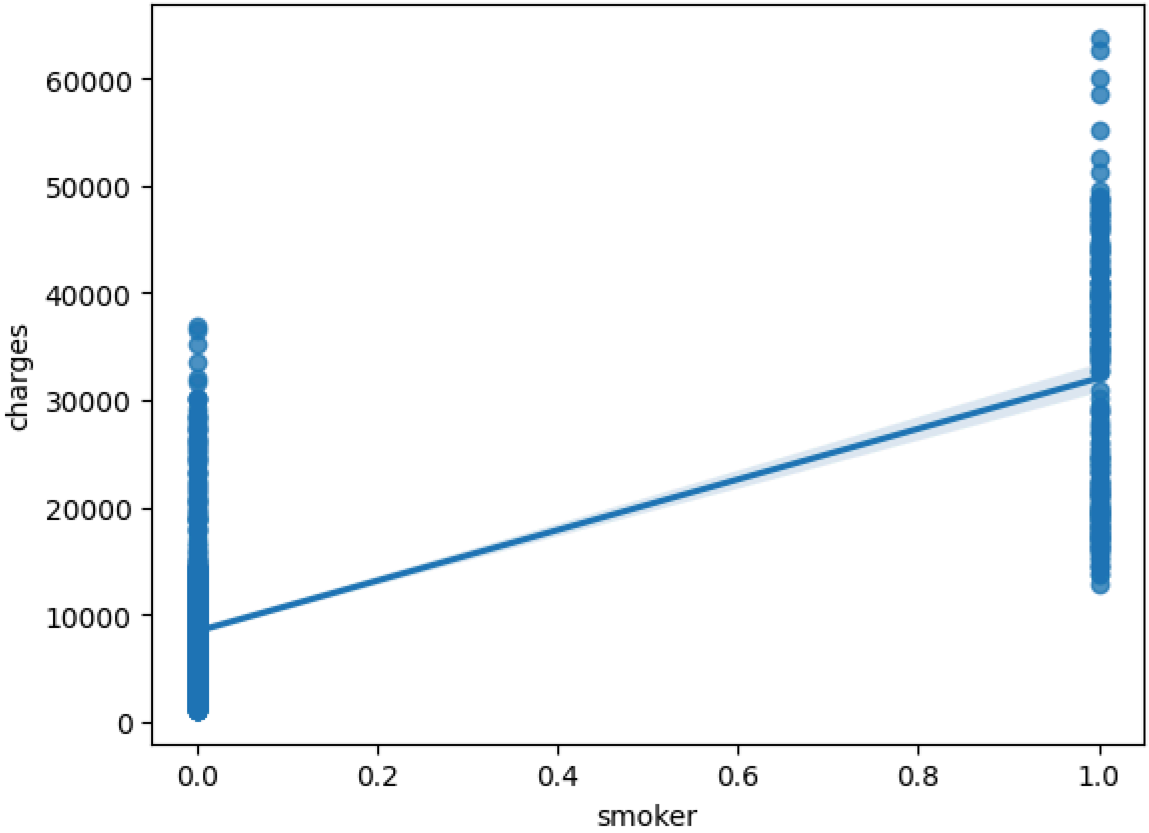
In [67]:
```
sns.regplot(x=df['age'], y=df["charges"],scatter=True)
```

Out[67]:
```
<Axes: xlabel='age', ylabel='charges'>
```



In [68]:
```
sns.regplot(x=df['smoker'], y=df["charges"],scatter=True)
```

Out[68]:
```
<Axes: xlabel='smoker', ylabel='charges'>
```



In [70]:

```
x2
```

Out[70]:

	age	sex	bmi	children	smoker	region
0	19	0	27.900	0	1	3
1	18	1	33.770	1	0	2
2	28	1	33.000	3	0	2
3	33	1	22.705	0	0	1
4	32	1	28.880	0	0	1
...	...	...	...	...	...	...
1333	50	1	30.970	3	0	1
1334	18	0	31.920	0	0	0
1335	18	0	36.850	0	0	2
1336	21	0	25.800	0	0	3
1337	61	0	29.070	0	1	1

1338 rows × 6 columns

In [69]:

```
import numpy as np
new_input = np.array([[30, 1, 28.5, 2, 0, 2]])
```

```
predict_charges=model2.predict(new_input)
print(f"the predicted amount for charges for the unseen  input:{predict_charges}")
```

```
the predicted amount for charges for the unseen  input:[[5590.25172292]]
```

```
C:\Users\Lenovo\AppData\Roaming\Python\Python313\site-packages\sklearn\utils\validat
ion.py:2749: UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```