```
import pandas as pd
import numpy as np
import seaborn as sns
```

```
df=pd.read_csv('heart_data.csv')
```

```
df
```

```
     Unnamed: 0     biking    smoking  heart.disease
0             1  30.801246  10.896608      11.769423
1             2  65.129215   2.219563       2.854081
2             3   1.959665  17.588331      17.177803
3             4  44.800196   2.802559       6.816647
4             5  69.428454  15.974505       4.062224
..          ...        ...        ...            ...
493         494  47.660440  27.562464      11.294392
494         495  45.097203  21.385620       9.616762
495         496   8.279743   6.423720      13.495168
496         497  42.345863  20.741328      10.115865
497         498  30.774254  23.610175      11.843556

[498 rows x 4 columns]
```

```
df=df.drop('Unnamed: 0', axis='columns')
```

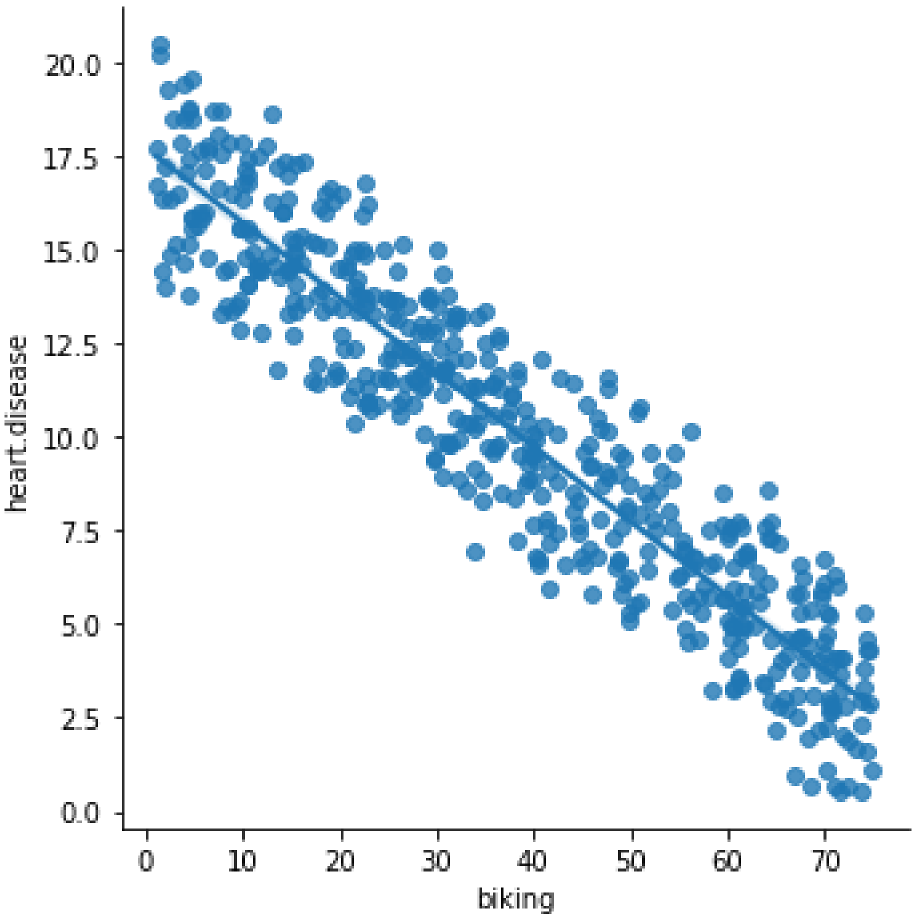
```
df
```

```
        biking    smoking  heart.disease
0    30.801246  10.896608      11.769423
1    65.129215   2.219563       2.854081
2     1.959665  17.588331      17.177803
3    44.800196   2.802559       6.816647
4    69.428454  15.974505       4.062224
..         ...        ...            ...
493  47.660440  27.562464      11.294392
494  45.097203  21.385620       9.616762
495   8.279743   6.423720      13.495168
496  42.345863  20.741328      10.115865
497  30.774254  23.610175      11.843556

[498 rows x 3 columns]
```

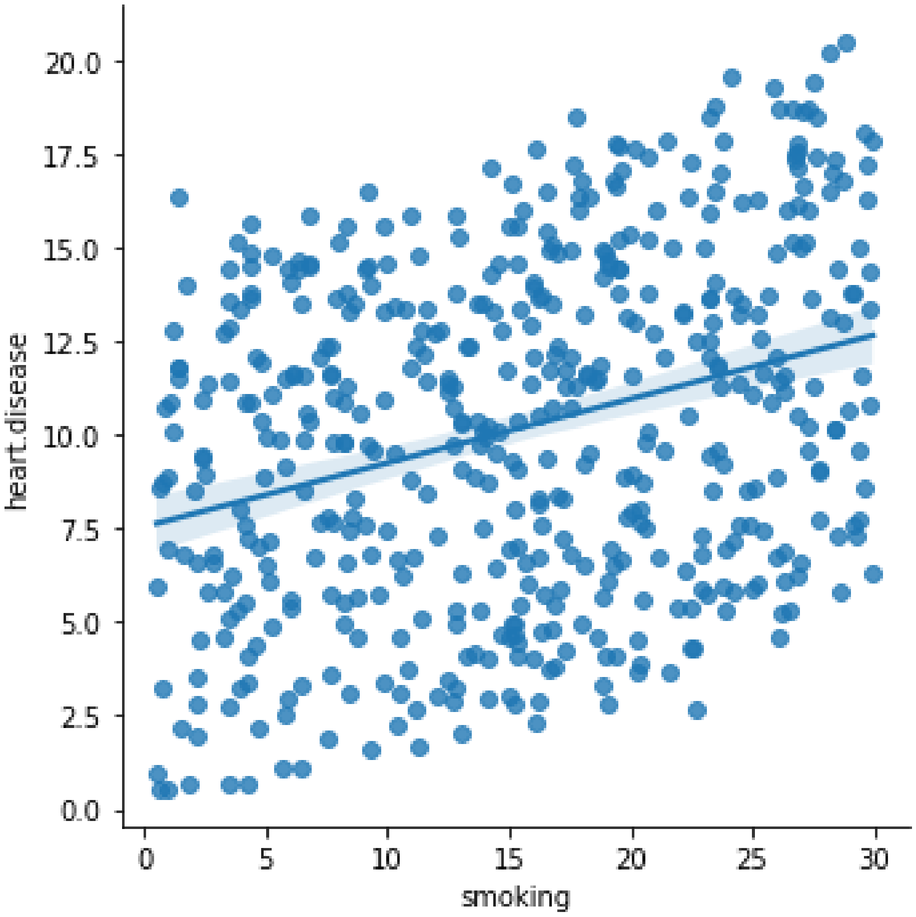
```
sns.lmplot(x='biking',y='heart.disease',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1db90419d00>
```



```
sns.lmplot(x='smoking',y='heart.disease',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1db8aa40a30>
```



```
x=df.drop('heart.disease', axis=1)

x

        biking    smoking
0    30.801246  10.896608
1    65.129215   2.219563
2     1.959665  17.588331
3    44.800196   2.802559
4    69.428454  15.974505
..         ...        ...
493  47.660440  27.562464
494  45.097203  21.385620
495   8.279743   6.423720
496  42.345863  20.741328
497  30.774254  23.610175

[498 rows x 2 columns]

y=df['heart.disease']

y
```

```
0      11.769423
1       2.854081
2      17.177803
3       6.816647
4       4.062224
         ...
493    11.294392
494     9.616762
495    13.495168
496    10.115865
497    11.843556
Name: heart.disease, Length: 498, dtype: float64
```

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

```python
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,
random_state=32)
```

```python
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```python
model.fit(x_train,y_train)
```

```
LinearRegression()
```

```python
model.score(x_train,y_train)
```

```
0.9817880512250096
```

```python
prediction = model.predict(x_test)
```

```python
print(y_test,prediction)
```

```
343    14.485812
330    14.298363
290    13.779836
79      6.516704
58      5.306818
         ...
328    12.734497
480     5.351619
313     4.626966
352     6.060989
102    10.526029
Name: heart.disease, Length: 150, dtype: float64 [14.11198335
14.84299461 14.94782327  6.07636572  5.80543019 16.02971083
 15.15707624 17.88513655  7.2158337   7.36001009 12.8206966
8.91449898
  3.15723147 17.68275775 11.5849347  12.8166386  17.80572847
6.42516064
 15.00553538  6.31605238 17.06328221  6.23645092 14.06091716
16.06813535
  9.33471974  6.66848373 13.06052547 12.5077536  12.93108569
17.5253187
 12.13277106  9.14025604 12.94809288 16.37426223 12.15497081
8.80618862
 14.19212307 12.38443042 10.96418608 19.03443184  6.02660778
15.41156526
  3.49941125 17.15935573  5.62818025  7.33034488 16.06320233
13.24340282
 14.24016109  6.66282708 11.17400879  4.81133686 15.3157782
2.31809729
 12.63920403 14.94678984 10.93597266 11.30345374 10.66625624
12.61084238
 10.74279651  2.90752925  4.03736868  7.31776506  7.18872359
14.99913225
  6.73103309 14.19770316  4.00544653 13.08225193 12.68175644
15.24906841
 14.43110114 10.976225    7.79870324  2.37748667  7.62469119
11.81203621
  7.19560567 11.10152917  4.05513833 11.93436895 11.25663789
3.99453093
 15.00732817  6.18624485 14.72596032  5.67254495  7.07691186
14.65967615
  6.96846204  8.75214639 15.04798859  4.55789856 17.93068035
6.33312069
  3.40253827 14.53006449  9.21100374  4.50733221  7.49157621
12.58092696
 10.42097012 10.5201324   5.21904613  3.96935198 14.84973116
12.22522759
 13.41626974 14.43191369  7.31303331  5.69524511 13.44409765
9.5588992
 10.16341489 14.71224831 19.06499004 11.44672175  4.86034767
3.81168431
  3.71701175  6.10670621 15.58085921  1.68986037 14.46212541
7.22936352
 10.54201521 13.43230702 15.12137697 17.41327713 12.84148929
7.43041492
 11.57354599  6.32432753 15.84634964  9.89744296 17.16885359
3.1005956
 11.96332019  3.56309432  4.20424383 13.86280603 12.25296235
15.98580284
  7.69552419 12.60138061  5.19600911  4.12976115  5.51875935
11.54161476]
```

```
print('mean squared error btw y_test and prediction :
',np.mean(prediction-y_test)**2)
```

```
mean squared error btw y_test and prediction :  0.00860252018315578
```

```
print("Mean sq. errror between y_test and predicted =",
np.mean(prediction-y_test)**2)
```

```
Mean sq. errror between y_test and predicted = 0.00860252018315578
```

```
import pickle
pickle.dump(model, open('model.pkl','wb'))
```

```
import os
print(os.getcwd())
```

```
C:\Users\akshu
```

```
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[20.1, 56.3]]))
```

```
[21.08910481]
```

```
C:\Users\akshu\anaconda3\lib\site-packages\sklearn\base.py:450:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```

```
print(model.coef_,"intercept :", model.intercept_)
```

```
[-0.20194791  0.17899177] intercept : 15.07102126126535
```

```
print(model.predict([[13, 23]]))
```

```
[16.5625091]
```

```
C:\Users\akshu\anaconda3\lib\site-packages\sklearn\base.py:450:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(
```