# Supervised Learning I

## Hemant Thapa

In [90]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

In [91]:
```
df = load_diabetes(as_frame=True).data[['s1', 's2']]
```

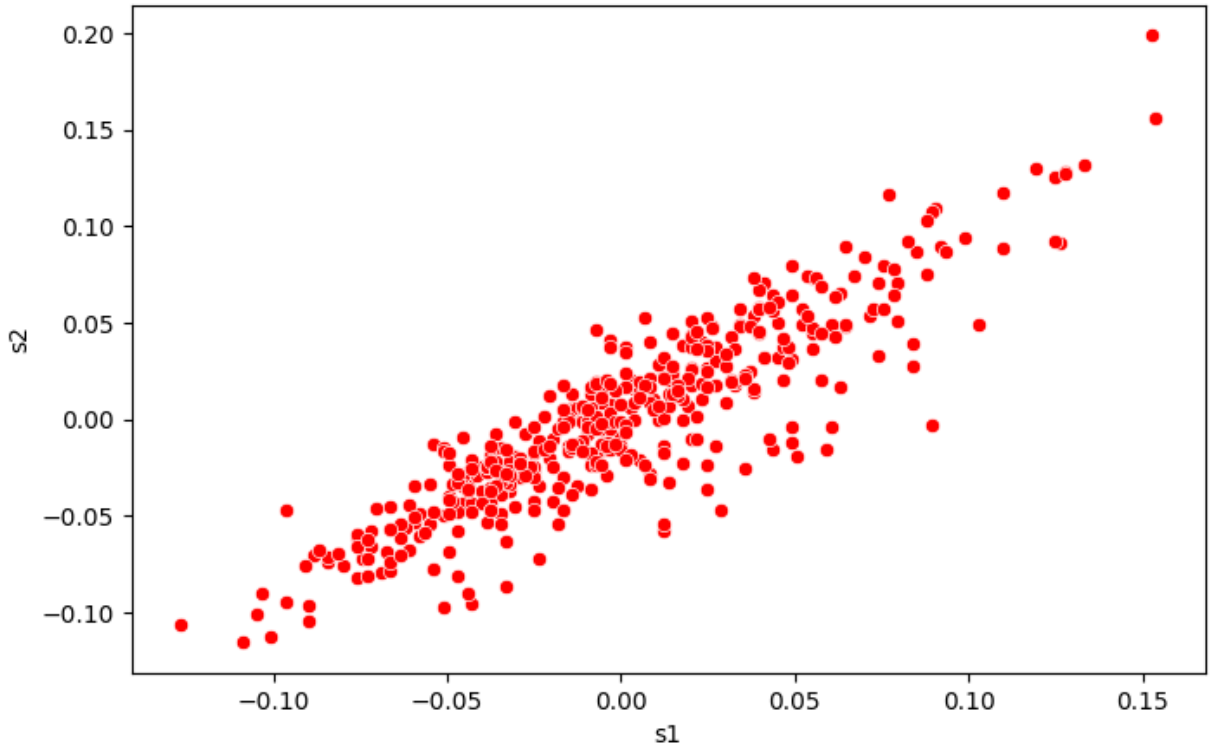
In [92]:
```
df
```

Out[92]:

| | s1 | s2 |
|---|---|---|
| **0** | -0.044223 | -0.034821 |
| **1** | -0.008449 | -0.019163 |
| **2** | -0.045599 | -0.034194 |
| **3** | 0.012191 | 0.024991 |
| **4** | 0.003935 | 0.015596 |
| **...** | ... | ... |
| **437** | -0.005697 | -0.002566 |
| **438** | 0.049341 | 0.079165 |
| **439** | -0.037344 | -0.013840 |
| **440** | 0.016318 | 0.015283 |
| **441** | 0.083740 | 0.027809 |

442 rows × 2 columns

In [93]:
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x='s1', y='s2', color='red')
plt.show()
plt.close()
```




In [94]:
```
model = LinearRegression()
```

In [95]:
```
X = df['s1'].values.reshape(-1,1)
```

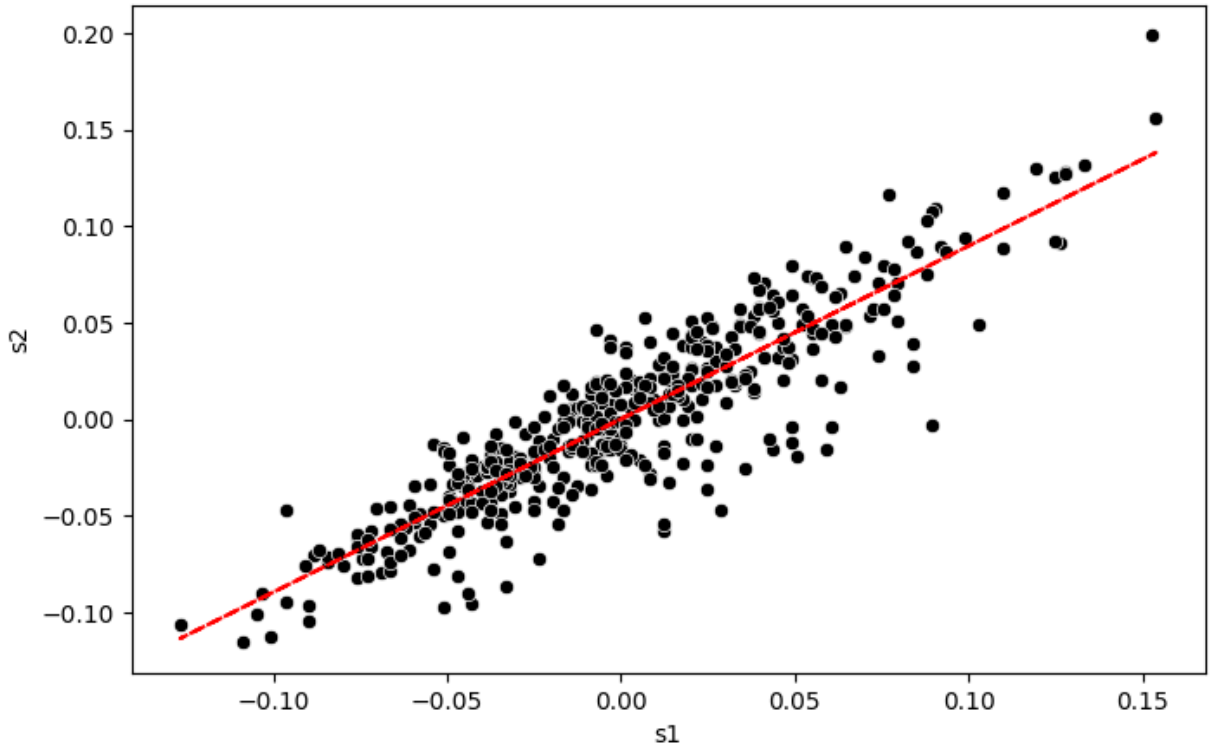
In [96]:
```
y = df['s2'].values
```

In [97]:
```
model.fit(X,y)
```

Out[97]:
```
▾ LinearRegression
LinearRegression()
```

In [98]:
```
y_pred = model.predict(X)
```

In [100…
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x='s1', y='s2', color='black')
plt.plot(X, y_pred, color='red',linestyle='--')
plt.show()
plt.close()
```




## Compulsory Task1

1. Importing dataset

In [101…
```
df = pd.read_csv('insurance.csv')
df
```

Out[101]:

| | 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 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **1333** | 50 | male | 30.970 | 3 | no | northwest | 10600.54830 |
| **1334** | 18 | female | 31.920 | 0 | no | northeast | 2205.98080 |
| **1335** | 18 | female | 36.850 | 0 | no | southeast | 1629.83350 |
| **1336** | 21 | female | 25.800 | 0 | no | southwest | 2007.94500 |
| **1337** | 61 | female | 29.070 | 0 | yes | northwest | 29141.36030 |

1338 rows × 7 columns

2. Inspecting columns

In [102…
```
df.columns
```

Out[102]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'],
dtype='object')
```

3. Inspecting Row and Columns

In [103…
```
df.shape
```

Out[103]:
```
(1338, 7)
```

4. Inspecting null values

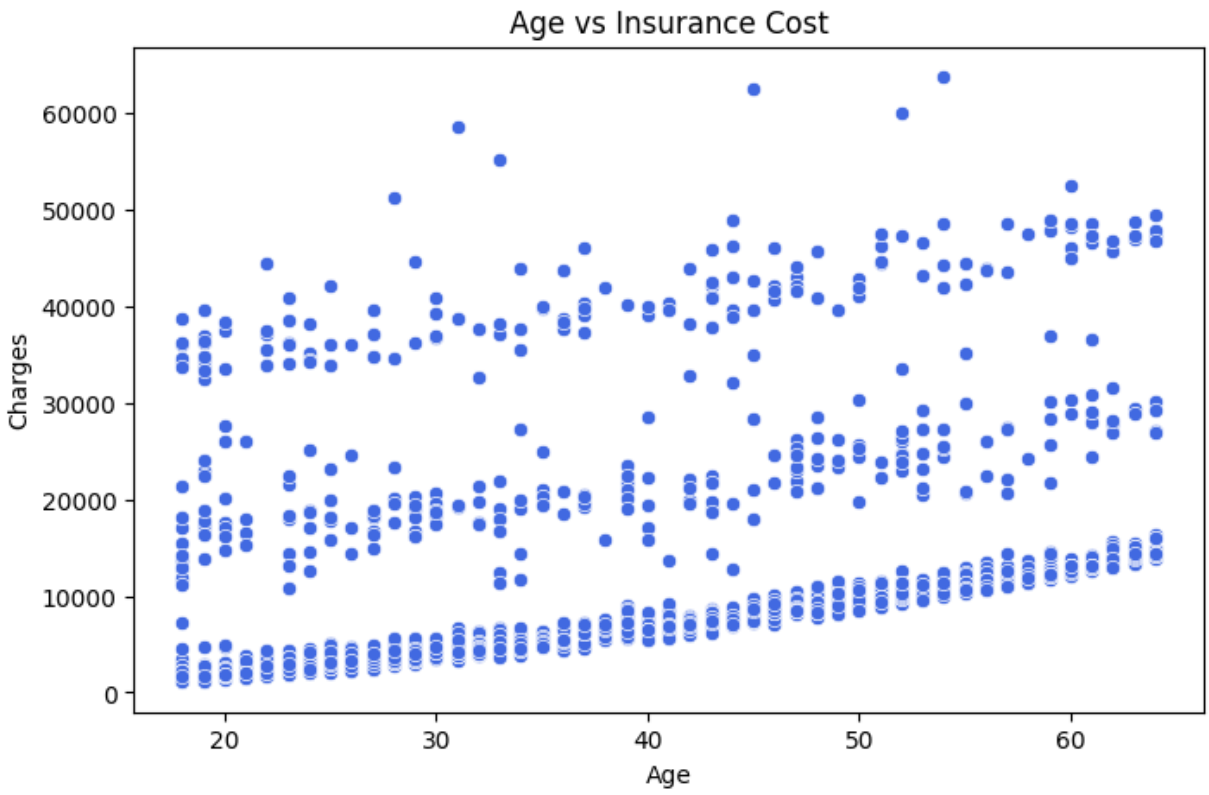
In [104…
```
df.isnull().sum()
```

Out[104]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

We don't have any missing values

## 1. Plotting a scatter plot with age on the x-axis and charges on the y-axis.

In [105…
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x = df['age'], y=df['charges'], color='royalblue
plt.title('Age vs Insurance Cost')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
plt.close()
```




## 2. Using linear_model.LinearRegression() from sklearn, fit a model to data, and making predictions on data.

In [106…
```
#X is age
X = df['age'].values.reshape(-1,1)
#y is charges
y = df['charges']
```

To allow compatibility, we must ensure that our model inputs are of shape (n_samples, n_features). Because there is only one feature, it will be (n_samples, 1):

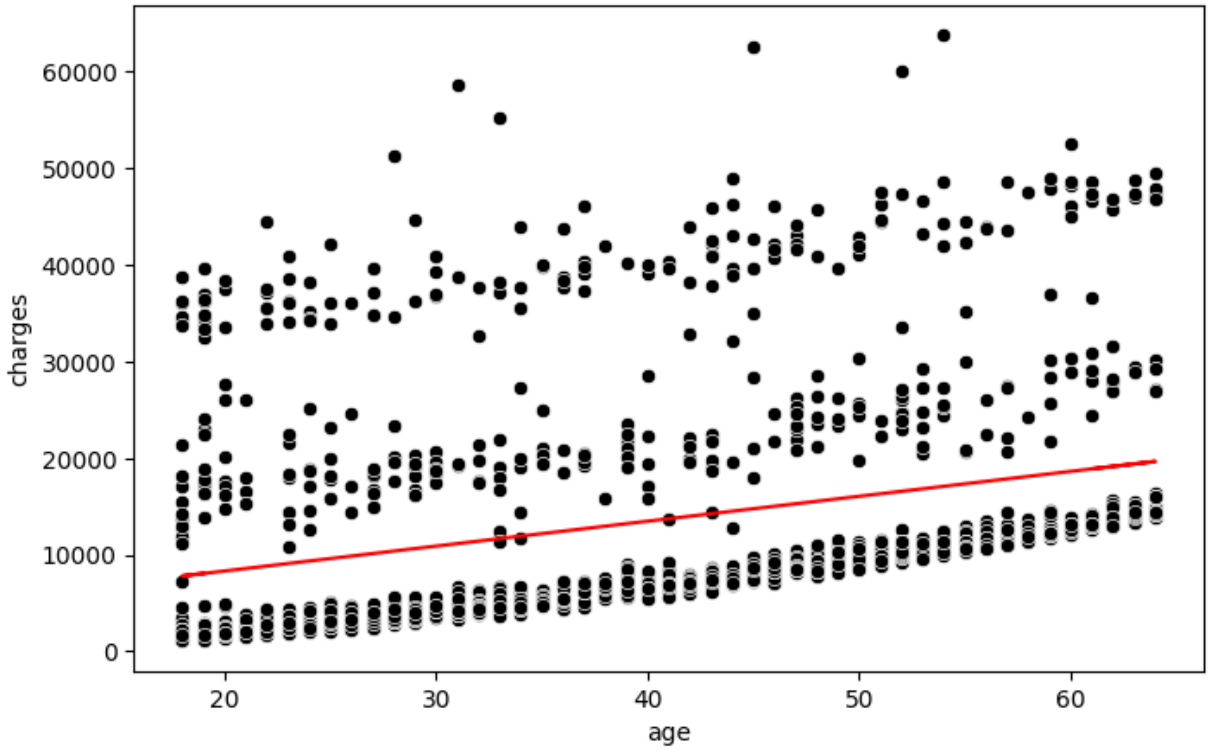
In [107…
```
#Training model
model.fit(X,y)
```

Out[107]:
```
▾ LinearRegression
LinearRegression()
```

In [108…
```
#Predicting value
y_pred = model.predict(X)
```

## 3. Plotting another scatter plot with the best-fit line.

In [109…
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x=df['age'], y=df['charges'], color='black')
plt.plot(X, y_pred, color='red')
plt.show()
plt.close()
```



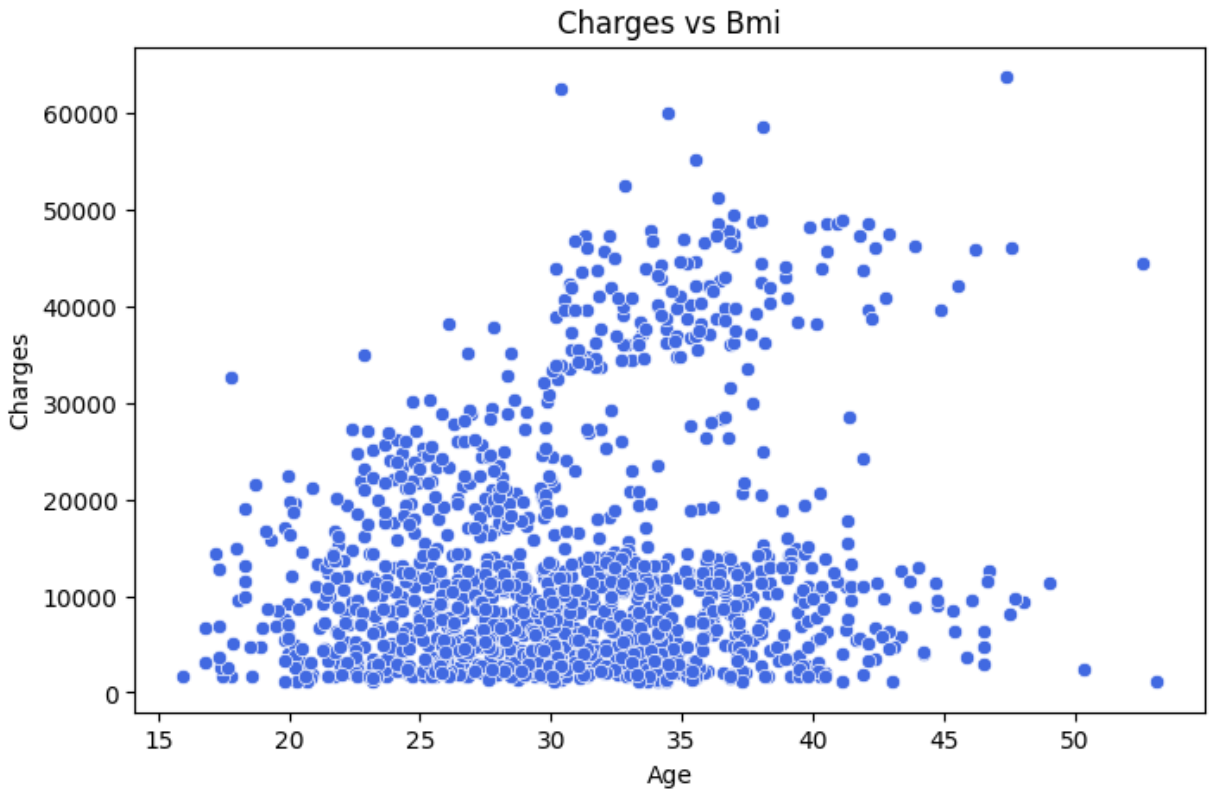

4. Now repeat the above using the data in the relevant columns to determine how BMI affects insurance costs.

In [110…
```
df.columns
```

Out[110]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'],
dtype='object')
```

Plotting scatter graph

In [125…
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x=df['bmi'],y = df['charges'], color='royalblue'
plt.title('Charges vs Bmi')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
plt.close()
```




Declaring X and y

In [119…
```
#X represent bmi
X = df['bmi'].values.reshape(-1,1)
#y represent charges
y = df['charges']
```

In [120…
```
#Training Linear Model
model.fit(X,y)
```

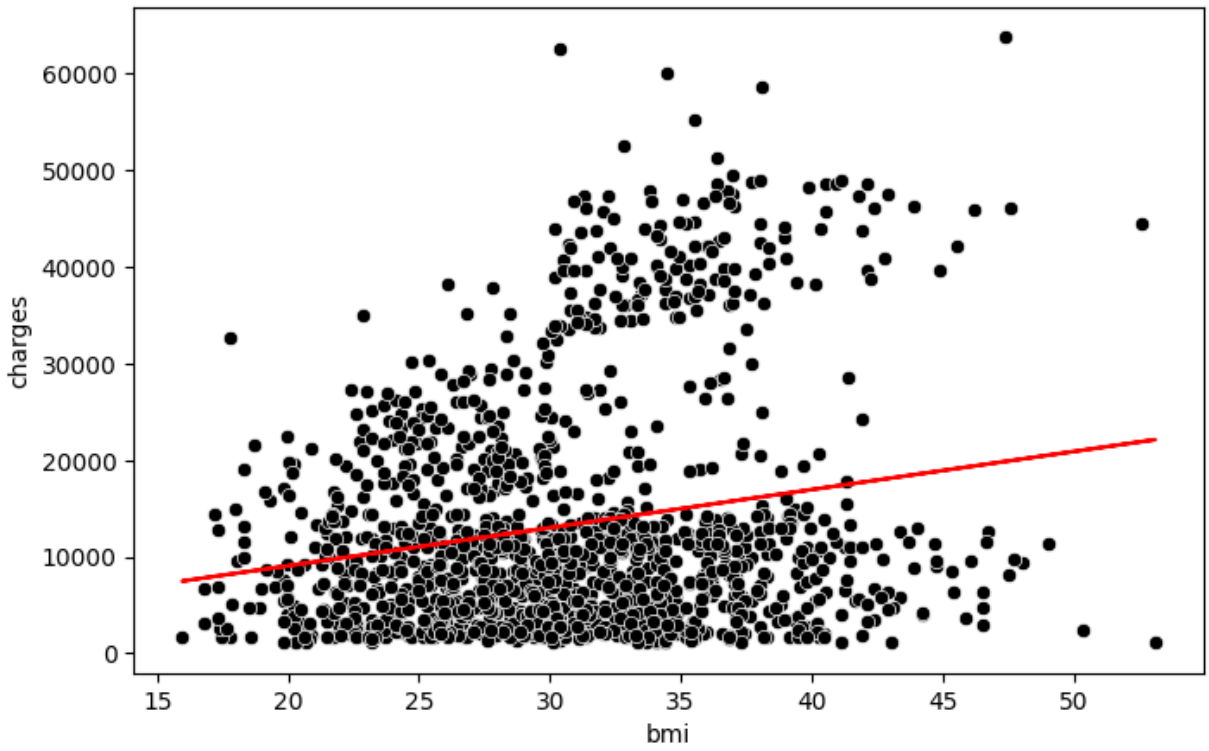
Out[120]:
```
▾ LinearRegression

LinearRegression()
```

In [121…
```
#Predicting value
y_pred = model.predict(X)
```

## Plotting best fit line on scatter plot

In [122…
```
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x=df['bmi'], y=df['charges'], color='black')
plt.plot(X, y_pred, color='red')
plt.show()
plt.close()
```


charges
60000
50000
40000
30000
20000
10000
0
15
20
25
30
35
40
45
50
bmi