In [17]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
```

In [2]:

```
df=pd.read_csv("../input/body-fat-prediction-dataset/bodyfat.csv")
```

In [3]:

```
df.head()
```

In [23]:

```
def plot_hist(variables):
    df[variables].plot.hist()
    print(variables)
    plt.show()
```

In [24]:

```
for c1 in list(df.columns):
    plot_hist(c1)
```

In [26]:

```
import seaborn as sns
sns.pairplot(df)
```

In [ ]:

```

```

In [27]:

```
Y=df['BodyFat']
X=df.drop('BodyFat',axis=1)
```

In [28]:

```
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

In [ ]:

```
xtrain,xtest,ytrain,ytest=train_test_split(X,Y,test_size=0.2)
```

In [29]:

```
ss= StandardScaler()
scaled_train= ss.fit_transform(xtrain)
scaled_test = ss.fit_transform(xtest)
```

In [30]:

```
from sklearn.neighbors import KNeighborsRegressor
knn= KNeighborsRegressor()
knn.fit(scaled_train,ytrain)
print(knn.score(scaled_test,ytest))
```

In [32]:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(scaled_train, ytrain)
print(rf.score(scaled_test,ytest))
```

In [34]:

```
from sklearn.tree import DecisionTreeRegressor
dtc = DecisionTreeRegressor()
dtc.fit(scaled_train,ytrain)
print(dtc.score(scaled_test, ytest))
```

In [35]:

```
regr=LinearRegression()
regr.fit(xtrain,ytrain)
regr.score(xtest,ytest)
```

In [13]:

```

```

In [15]:

```

```

In [16]:

```

```

In [ ]:

```

```