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

```
x=np.arange(10,120,10)
y=np.arange(100,210,10)
```

```
x
```

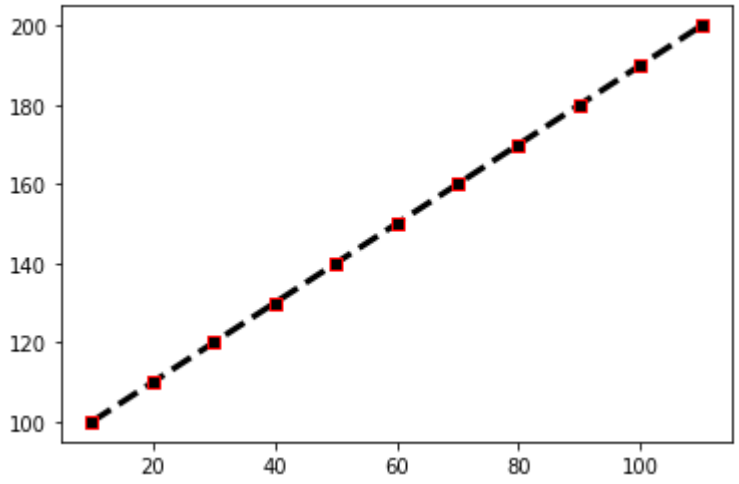
```
array([ 10,  20,  30,  40,  50,  60,  70,  80,  90, 100, 110])
```

```
y
```

```
array([100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200])
```

```
plt.plot(x,y,marker='s',ls='--',lw=3,mec='r',color='k')
```

```
[<matplotlib.lines.Line2D at 0x7f003f550990>]
```




```
model=LinearRegression()  #Y=MX+C
x=np.reshape(x,(-1,1))
y=np.reshape(y,(-1,1))

model.fit(x,y)
```

```
LinearRegression()
```

```
yp=model.predict(x)
yp
```

```
array([[100.],
```

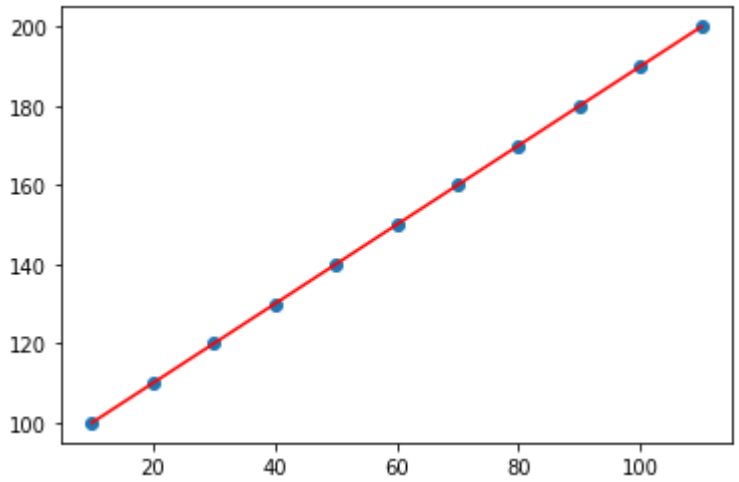
```
            [110.],
            [120.],
            [130.],
            [140.],
            [150.],
            [160.],
            [170.],
            [180.],
            [190.],
            [200.]])
```

```
sqe=(y-yp)**2
sqe
```

```
array([[2.01948392e-28],
       [2.01948392e-28],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00]])
```

```
plt.scatter(x,y)
plt.plot(x,yp,color='r')
```

```
[<matplotlib.lines.Line2D at 0x7f003f042e50>]
```




```
model.intercept_
```

```
array([90.])
```

```
model.coef_
```

```
array([[1.]])
```

```
ssr=np.sum(sqe)
sst=np.sum((y-np.mean(y))**2)
r2_score=1-(ssr/sst)
```

```
ssr#sum of square of residuals
```

```
4.0389678347315804e-28
```

```
sst#sum of square total
```

```
11000.0
```

```
r2_score
```

```
1.0
```

```
print("Absolute mean error",mean_absolute_error(y,yp))
```

```
Absolute mean error 2.5837917664003644e-15
```

```
print("Mean squared error",mean_squared_error(y,yp))
```

```
Mean squared error 3.671788940665073e-29
```

```
print("Root mean squared error",np.sqrt(mean_squared_error(y,yp)))
```

```
Root mean squared error 6.059528810613143e-15
```

```
np.std(sqe)
```

```
7.789040577090127e-29
```

```
np.mean(sqe)
```

```
3.671788940665073e-29
```

```
  from numpy import arange
  from pandas import read_csv
  from sklearn.linear_model import RidgeCV
  from sklearn.model_selection import RepeatedKFold
```

```
url='https://raw.githubusercontent.com/jbrownlee/Datasets/master/housing.csv'
dataframe=read_csv(url,header=None)
```

```
data=dataframe.values
```

```
# define model evaluation method
cv=RepeatedKFold(n_splits=10,n_repeats=3,random_state=12)
```

```
model=RidgeCV(alphas=arange(0,1,0.01),cv=cv,scoring='neg_mean_absolute_error')
model.fit(x,y)
```

```
RidgeCV(alphas=array([0.  , 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ,
       0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 , 0.21,
       0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 , 0.31, 0.32,
       0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 , 0.41, 0.42, 0.43,
       0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 , 0.51, 0.52, 0.53, 0.54,
       0.55, 0.56, 0.57, 0.58, 0.59, 0.6 , 0.61, 0.62, 0.63, 0.64, 0.65,
       0.66, 0.67, 0.68, 0.69, 0.7 , 0.71, 0.72, 0.73, 0.74, 0.75, 0.76,
       0.77, 0.78, 0.79, 0.8 , 0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87,
       0.88, 0.89, 0.9 , 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98,
       0.99]),
        cv=RepeatedKFold(n_repeats=3, n_splits=10, random_state=12),
        scoring='neg_mean_absolute_error')
```

```
# summarize chosen configuration
print('alpha:%f'%model.alpha_)
```

```
alpha:0.610000
```