In [1]:

```python
import numpy as np
import pandas as pd
```

In [2]:

```python
data= pd.read_csv('abc.csv')
data
```

Out[2]:

	first	second
0	0	1
1	1	3
2	2	2
3	3	5
4	4	7
5	5	8
6	6	8
7	7	9
8	8	10
9	9	12

In [3]:

```python
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
```

In [4]:

```python
x
```

Out[4]:

```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5],
       [6],
       [7],
       [8],
       [9]], dtype=int64)
```

In [5]:

```python
y
```

Out[5]:

```
array([ 1,  3,  2,  5,  7,  8,  8,  9, 10, 12], dtype=int64)
```

In [6]:

```python
from  sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x,y)
```

Out[6]:

```
LinearRegression()
```

In [7]:

```python
y_pred=reg.predict(x)
```

In [8]:

```python
y_pred
```

Out[8]:

```
array([ 1.23636364,  2.40606061,  3.57575758,  4.74545455,  5.91515152,
        7.08484848,  8.25454545,  9.42424242, 10.59393939, 11.76363636])
```

In [9]:

```python
sse = np.sum((y-y_pred)**2)
ssr=np.sum((y_pred-np.mean(y))**2)
sst=sse+ssr
```

In [10]:

```python
sst
```

Out[10]:

```
118.49999999999986
```

In [11]:

```python
sse
```

Out[11]:

```
5.624242424242425
```

In [12]:

```python
ssr
```

Out[12]:

```
112.87575757575743
```

In [13]:

```python
r=1-(sse/sst)
```

In [14]:

```
1 r
```

Out[14]:

0.9525380386139879

In [15]:

```
1 n=x.shape[0]
2 p=x.shape[1]
3
```

In [16]:

```
1 ad=1-(1-r)*((n-1)/(n-p-1))
2 ad
```

Out[16]:

0.9466052934407364

In [ ]:

```
1
```