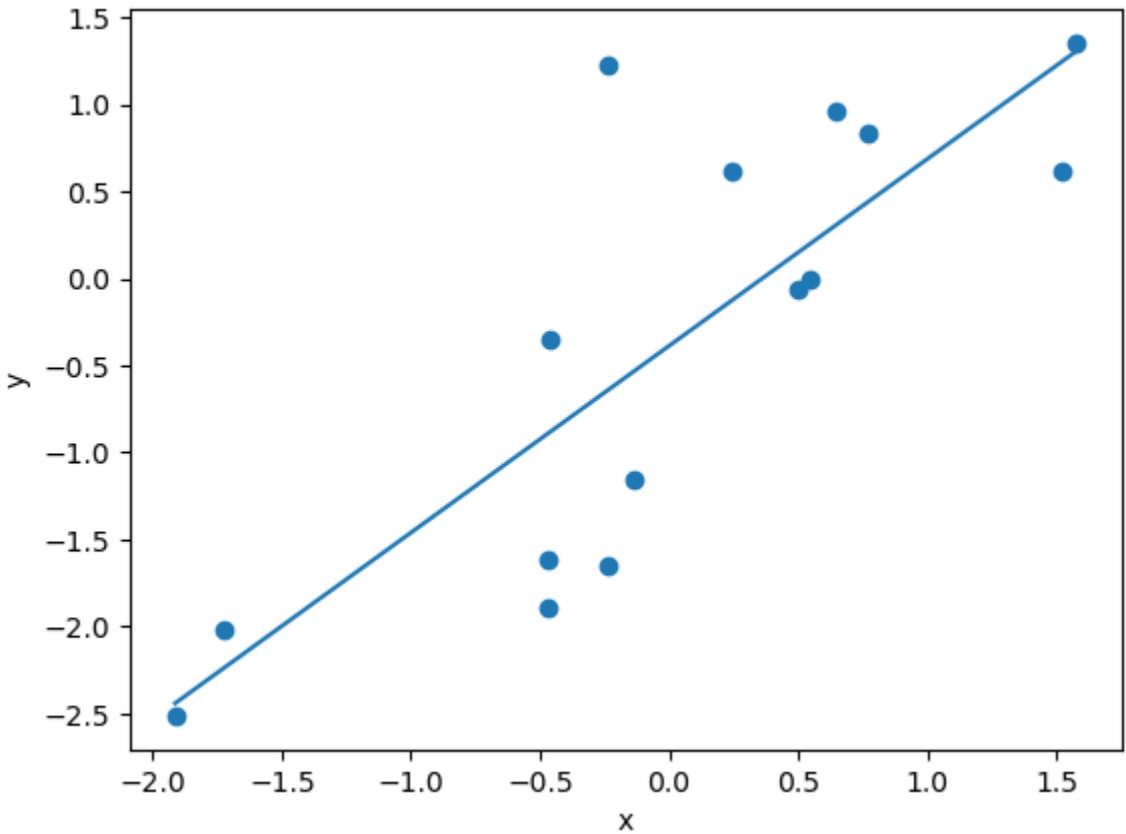


```
In [15]: import numpy as np
from scipy.stats import pearsonr
import matplotlib.pyplot as plt
np.random.seed(42)
x=np.random.randn(15)
y=x+np.random.randn(15)
plt.scatter(x,y)
plt.plot(np.unique(x),np.poly1d(np.polyfit(x,y,1))(np.unique(x)))
plt.xlabel('x')
plt.ylabel('x')
plt.ylabel('y')
```

Out[15]: Text(0, 0.5, 'y')



```
In [11]: corr,_=pearsonr(x,y)
print("pearsons correlation:%3f"%corr)

pearsons correlation:0.809537
```

```
In [20]: doc_trump="Mr.Trump became president after winning the political election through the lost the support of some republican friends,Trump is friends with president Putin"
doc_election="President Trump says putin had no politicians interference"
doc_putin="Vladimir putin is a Russian politician and former intelligence officer"
documents=[doc_trump,doc_election,doc_putin]
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd
Count_Vectorizer=CountVectorizer(stop_words='english')
Count_Vectorizer=CountVectorizer()
sparse_matrix=Count_Vectorizer.fit_transform(documents)
doc_term_matrix=sparse_matrix.todense()
df=pd.DataFrame(doc_term_matrix,columns=Count_Vectorizer.get_feature_names(),index=['doc_trump','doc_election','doc_putin'])
df
```

Out[20]:

	after	and	became	election	former	friends	had	intelligence	interference	is	...	russian	says	some	support	the	through	trump	vladimir	winning	with
doc_trump	1	0	1	1	0	2	0	0	0	1	...	0	0	1	1	3	1	2	0	1	1
doc_election	0	0	0	0	0	0	1	0	1	0	...	0	1	0	0	0	0	1	0	0	0
doc_putin	0	1	0	0	1	0	0	1	0	1	...	1	0	0	0	0	0	0	1	0	0

3 rows × 31 columns

```
In [1]: import numpy as np
a=[4,6,3,2,7,2]
b=[0,2,3,4,6,7,8]
def jaccard(list1,list2):
    intersection=len(list(set(list1).intersection(list2)))
    union=(len(list1)+len(list2))-intersection
    return float(intersection)/union
jaccard(a,b)
```

Out[1]: 0.625

```
In [26]: from scipy.spatial import distance
A=[1,2,3,4,5,6]
B=[7,8,9,10,11,12]
A,B
euclidean_distance=distance.euclidean(A,B)
print('Euclidean distance b/w',A , 'and',B,'is',euclidean_distance)

Euclidean distance b/w [1, 2, 3, 4, 5, 6] and [7, 8, 9, 10, 11, 12] is 14.696938456699069
```

```
In [27]: A=[1,2,3,4,5,6]
B=[7,8,9,10,11,12]
A,B
manhattan_distance=distance.cityblock(A,B)
print('manhattan distance b/w',A , 'and',B,'is',manhattan_distance)

manhattan distance b/w [1, 2, 3, 4, 5, 6] and [7, 8, 9, 10, 11, 12] is 36
```