In [1]:

```
import numpy as np
```

In [5]:

```
X = np.array([2,5,1])
X.shape
```

Out[5]:

```
(3,)
```

In [6]:

```
np.linalg.norm(X, ord=1)
```

Out[6]:

```
8.0
```

In [8]:

```
a = np.array([
    [1, 2],
    [2, 3],
    [3, 4],
    [4, 5]
])

b = np.array([
    [2, 3],
    [3, 4],
    [4, 5],
    [5, 6]
])

a + b
```

Out[8]:

```
array([[ 3,  5],
       [ 5,  7],
       [ 7,  9],
       [ 9, 11]])
```

In [10]:

```
a = np.array([
    [2, 1, 3],
    [3, 4, -1],
    [6, 5, 1],
    [7, 1, -2]
])

b = np.array([
    [1, -2, 2, 3],
    [2, 3, -2, 2],
    [-1, 8, 5, 9],
])

a.dot(b)
```

Out[10]:

```
array([[  1,  23,  17,  35],
       [ 12,  -2,  -7,   8],
       [ 15,  11,   7,  37],
       [ 11, -27,   2,   5]])
```

In [11]:

```
B= np.array([[2,1,-2],[3,-2,1],[2,-3,2]])

#逆行列の計算
inv_B = np.linalg.inv(B)
```

In [12]:

```
inv_B
```

Out[12]:

```
array([[-0.25,  1.  , -0.75],
       [-1.  ,  2.  , -2.  ],
       [-1.25,  2.  , -1.75]])
```

In [13]:

```
def sigmoid(a):
    return 1 / (1 + np.exp(-a))
```

In [20]:

```
sigmoid(100)
```

Out[20]:

```
1.0
```

In [22]:

```
def RSS(y, t):
    return 0.5 * (np.sum((y - t)**2))
```

In [25]:

```
w = np.array([0.2])
x = np.array([25])
t = np.array([1])
a = w.dot(x)
y = sigmoid(a)
E = RSS(y, t)
E
```

Out[25]:

```
2.2397126747349496e-05
```

In [27]:

```
from keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train.shape
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
mnist.npz
11493376/11490434 [==============================] - 2s 0us/step
11501568/11490434 [==============================] - 2s 0us/step
```

Out[27]:

```
(60000, 28, 28)
```

In [28]:

```
x_train.ndim
```

Out[28]:

```
3
```

In [29]:

```
import pandas as pd
s = pd.Series([3,2,3,2,2])
s2 = pd.Series([2,4,6,8,10])
s
```

Out[29]:

```
0    3
1    2
2    3
3    2
4    2
dtype: int64
```

In [30]:

```
s2
```

Out[30]:

```
0     2
1     4
2     6
3     8
4    10
dtype: int64
```

In [37]:

```
df = pd.concat([s, s2], axis=1)
# df = pd.merge([s, s2], axis=1)
df
```

Out[37]:

	0	1
0	3	2
1	2	4
2	3	6
3	2	8
4	2	10

In [38]:

```
type(df)
```

Out[38]:

```
pandas.core.frame.DataFrame
```

In [41]:

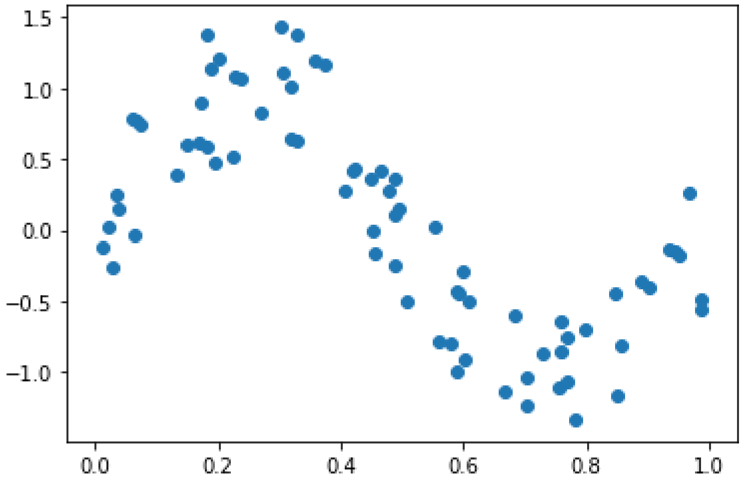
```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split as tts
def f(x):
    return np.sin(2*np.pi*x)
```

In [46]:

```
sin_x = np.linspace(0, 1, 100)
x = np.random.rand(100)[:, np.newaxis]
y = f(x) + np.random.rand(100)[:, np.newaxis] - 0.5
x_train, x_test, y_train, y_test = tts(x,y)
# plt.plot(sin_x, f(sin_x), ':')
plt.scatter(x_train, y_train)
```

Out[46]:

<matplotlib.collections.PathCollection at 0x7f9aa05207d0>

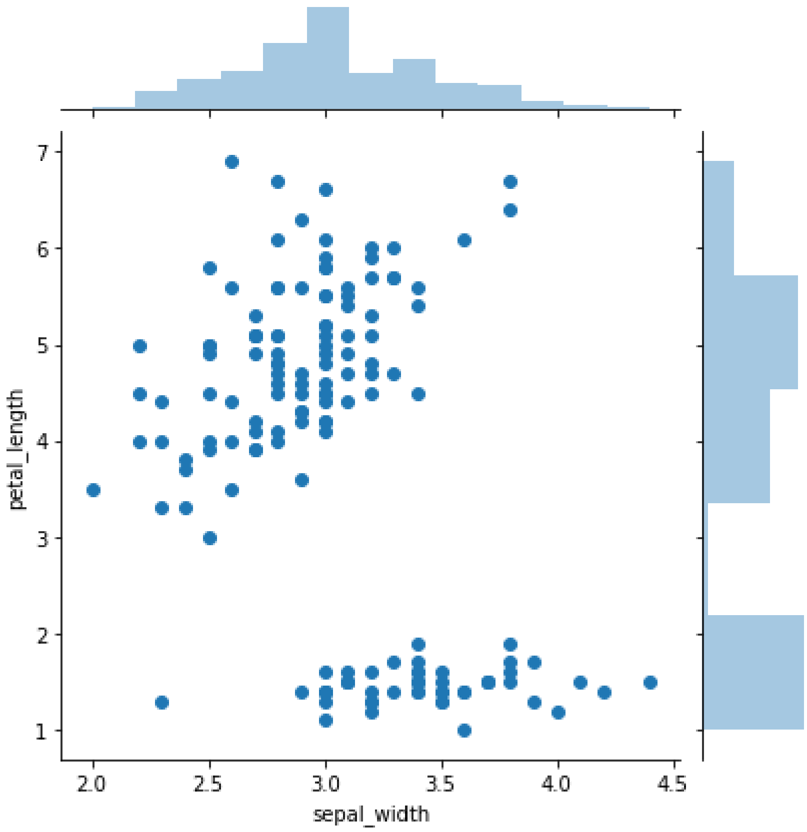


In [56]:

```
import seaborn as sns
iris = sns.load_dataset("iris")
type(iris)
sns.jointplot(x='sepal_width', y='petal_length', data=iris,  kind='scatter')
```

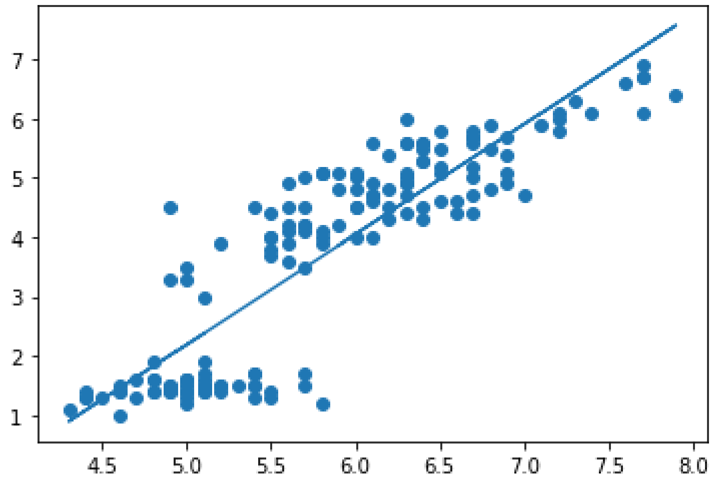
Out[56]:

<seaborn.axisgrid.JointGrid at 0x7f9aa366bd90>



In [66]:

```
from sklearn import datasets, linear_model
iris = datasets.load_iris()
x = iris.data[:, 0]
y = iris.data[:, 2]
model = linear_model.LinearRegression()
model.fit(x.reshape(-1, 1), y)
plt.scatter(x, y)
plt.plot(x, model.predict(x.reshape(-1,1)))
# plt.plot(x, model.predict(y))
plt.show()
```



Out[66]:

```
array([5.1, 4.9, 4.7, 4.6, 5. , 5.4, 4.6, 5. , 4.4, 4.9, 5.4, 4.8, 4.8,
       4.3, 5.8, 5.7, 5.4, 5.1, 5.7, 5.1, 5.4, 5.1, 4.6, 5.1, 4.8, 5. ,
       5. , 5.2, 5.2, 4.7, 4.8, 5.4, 5.2, 5.5, 4.9, 5. , 5.5, 4.9, 4.4,
       5.1, 5. , 4.5, 4.4, 5. , 5.1, 4.8, 5.1, 4.6, 5.3, 5. , 7. , 6.4,
       6.9, 5.5, 6.5, 5.7, 6.3, 4.9, 6.6, 5.2, 5. , 5.9, 6. , 6.1, 5.6,
       6.7, 5.6, 5.8, 6.2, 5.6, 5.9, 6.1, 6.3, 6.1, 6.4, 6.6, 6.8, 6.7,
       6. , 5.7, 5.5, 5.5, 5.8, 6. , 5.4, 6. , 6.7, 6.3, 5.6, 5.5, 5.5,
       6.1, 5.8, 5. , 5.6, 5.7, 5.7, 6.2, 5.1, 5.7, 6.3, 5.8, 7.1, 6.3,
       6.5, 7.6, 4.9, 7.3, 6.7, 7.2, 6.5, 6.4, 6.8, 5.7, 5.8, 6.4, 6.5,
       7.7, 7.7, 6. , 6.9, 5.6, 7.7, 6.3, 6.7, 7.2, 6.2, 6.1, 6.4, 7.2,
       7.4, 7.9, 6.4, 6.3, 6.1, 7.7, 6.3, 6.4, 6. , 6.9, 6.7, 6.9, 5.8,
       6.8, 6.7, 6.7, 6.3, 6.5, 6.2, 5.9])
```

In [70]:

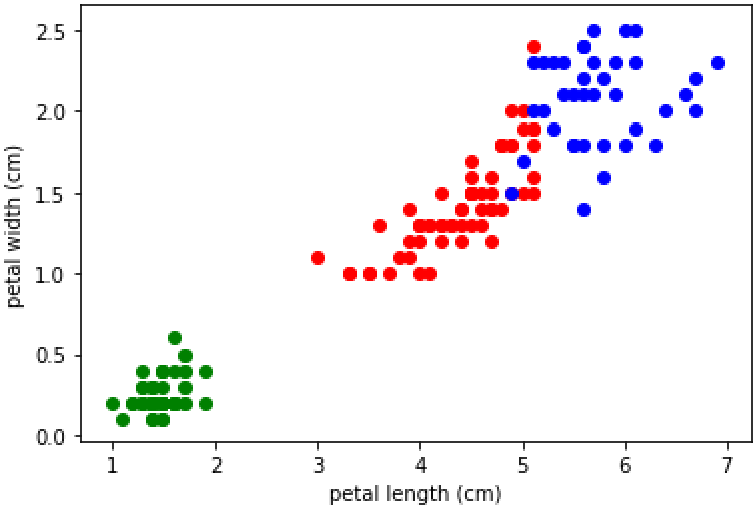
```
a = np.array([[1,2,3], [4,5,6]])
a.reshape(3, -1)
```

Out[70]:

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

In [84]:

```
from sklearn import cluster, datasets
iris = datasets.load_iris()
data = iris.data
model = cluster.KMeans(n_clusters=3)
model.fit(data)
labels = model.labels_
ldata = data[labels == 0]
plt.scatter(ldata[:,2], ldata[:,3], color='green')
ldata = data[labels == 1]
plt.scatter(ldata[:,2], ldata[:,3], color='red')
ldata = data[labels == 2]
plt.scatter(ldata[:,2], ldata[:,3], color='blue')
plt.xlabel(iris['feature_names'][2])
plt.ylabel(iris['feature_names'][3])
plt.show()
```



In [85]:

```
a = np.array([[1,2], [3,4]])
b = np.array([[8,7], [6,5]])
a.dot(b)
```

Out[85]:

```
array([[20, 17],
       [48, 41]])
```

In [108]:

```
x = np.array([
    [6],
    [9]
])
w = np.array([
    [3,5],
    [4,8]
])
b = np.array([
    [4],
    [6]
])
# w.shape
a1 = w.dot(x) + b
a1
```

Out[108]:

```
array([[ 67],
       [102]])
```

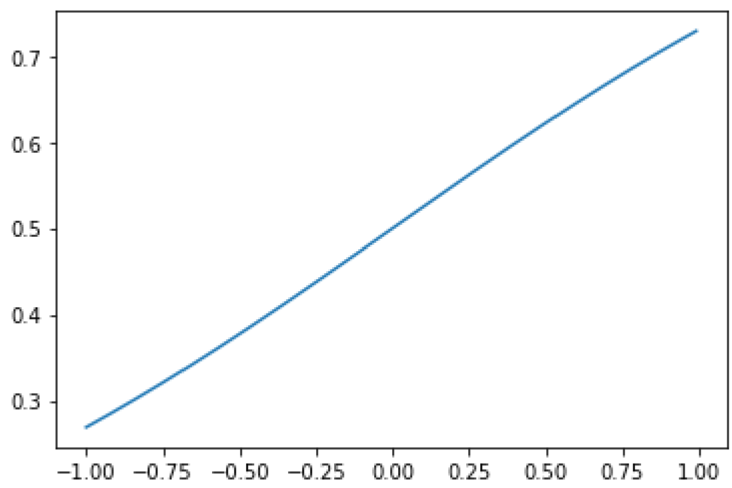
In [112]:

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(-1, 1, 0.01)
s = sigmoid(t)
plt.plot(t, s)
t
sigmoid(0.7), sigmoid(-0.3)
```

Out[112]:

```
(0.6681877721681662, 0.425557483188341)
```



In [114]:

```
a = np.array([
    [1,2],
    [3,4]
])

b = np.array([
    [1,2],
    [2,1]
])
a * b
```

Out[114]:

```
array([[1, 4],
       [6, 4]])
```

In [115]:

```
y = np.array([
    [0.6],
    [0.2]
])

t = np.array([
    [0],
    [1]
])

(y - t) * ((1 - y) * y)
```

Out[115]:

```
array([[ 0.144],
       [-0.128]])
```

微分と積分の具体例

x^n の微分

$$nx^{n-1} \quad \underset{\frac{d}{dx}x^n = nx^{n-1}}{\overset{\int nx^{n-1}dx = x^n}{\rightleftharpoons}} \quad x^n$$

指数関数の微分

$$e^x \quad \underset{\frac{d}{dx}e^x \, = \, e^x}{\overset{\int e^x dx \, = \, e^x}{\rightleftharpoons}} \quad e^x$$

対数関数の微分

$$\frac{1}{x} \quad \underset{\frac{d}{dx}\log_e x \, = \, \dfrac{1}{x}}{\overset{\int \frac{1}{x} dx \, = \, log_e x}{\rightleftharpoons}} \quad log_e x$$

三角関数の微分

$$-\sin x \quad \underset{\frac{d}{dx}\cos x \, = \, -\sin x}{\overset{\int (-\sin x) dx \, = \, \cos x}{\rightleftharpoons}} \quad \cos x$$

In [ ]:

```

```