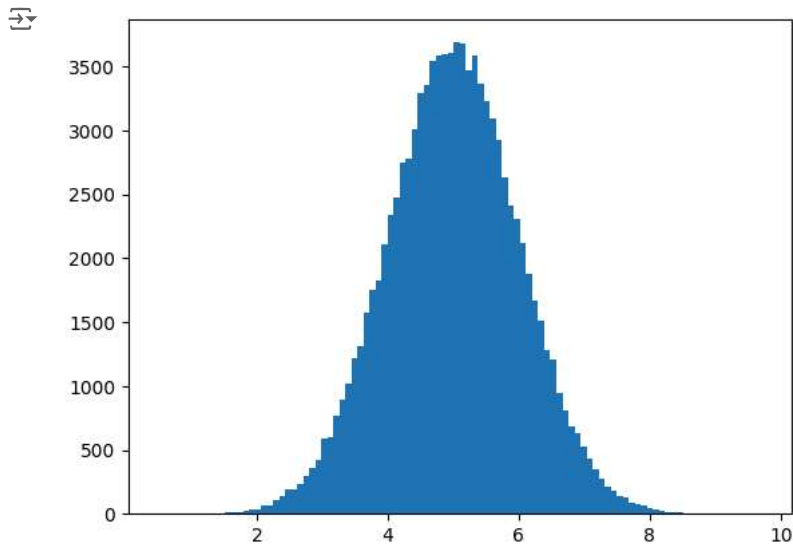


```
import numpy
import matplotlib.pyplot as plt

x = numpy.random.normal(5.0, 1.0, 100000)

plt.hist(x, 100)
plt.show()
```



```
y = numpy.random.normal(10.0, 2.0, 1000)

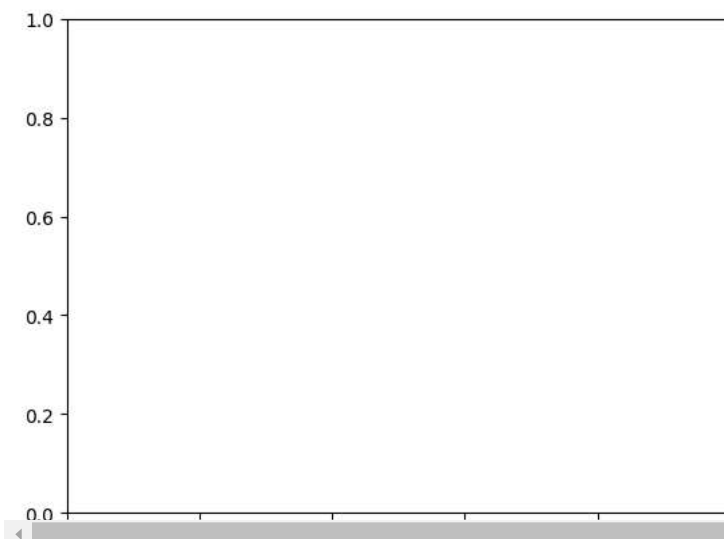
plt.scatter(x, y)
plt.show()
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-26-37c35772ffa1> in <cell line: 3>()
      1 y = numpy.random.normal(10.0, 2.0, 1000)
      2
----> 3 plt.scatter(x, y)
      4 plt.show()
```

2 frames

```
/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_axes.py in scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha,
linewidths, edgecolors, plotnonfinite, **kwargs)
    4582     y = np.ma.ravel(y)
    4583     if x.size != y.size:
-> 4584         raise ValueError("x and y must be the same size")
    4585
    4586     if s is None:
```

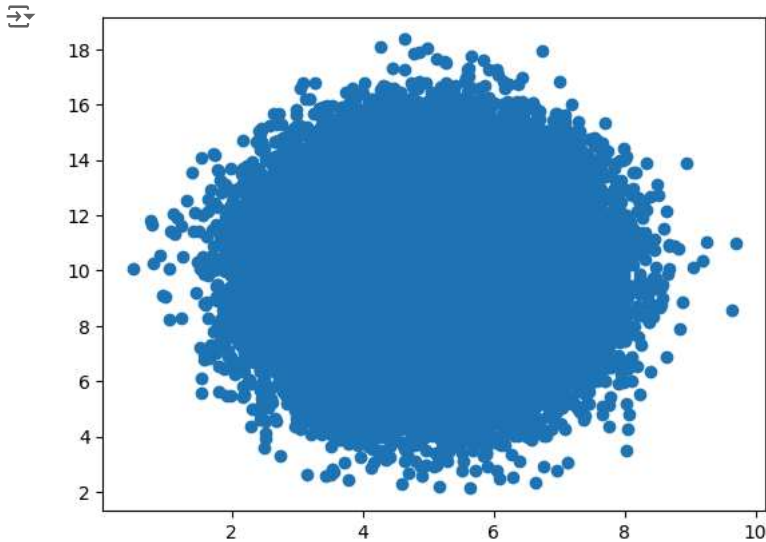
ValueError: x and y must be the same size



後續步驟: [說明錯誤](#)

```
y = numpy.random.normal(10.0, 2.0, 100000)
```

```
plt.scatter(x, y)
plt.show()
```



```
from google.colab import drive
drive.mount('/gdrive')
```

↪ Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdrive", force\_remount=True).

```
with open('/gdrive/My Drive/foo.txt', 'w') as f:
    f.write('您好 Google Drive!')
!cat '/gdrive/My Drive/foo.txt'
```

↪ 您好 Google Drive!

#針對google drive的存取, 也可以利用python的PyDrive函式庫簡化對Google Drive API的使用, 相關範例如下:

```
# Import PyDrive and associated libraries.
# This only needs to be done once in a notebook.
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
```

```
# Authenticate and create the PyDrive client.
# This only needs to be done once in a notebook.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

```
# Create & upload a text file.
uploaded = drive.CreateFile({'title': 'PyDriveSample.txt'})
uploaded.SetContentString('Sample upload file content 範例')
uploaded.Upload()
print('Uploaded file with ID {}'.format(uploaded.get('id')))
```

↪ Uploaded file with ID 1cxeFNj0XN0x-LaC-Q1ZZz9FEq7XwW5aG

```
# List .txt files in the root.
# Search query reference:
# https://developers.google.com/drive/v2/web/search-parameters
listed = drive.ListFile({'q': "title contains '.txt' and 'root' in parents"}).GetList()
for file in listed:
    print('title {}, date {}, id {}'.format(file['title'], file['createdDate'], file['id']))
```

↪ title foo.txt, date 2024-10-30T07:56:55.458Z, id 1m8HlnU4tS6mSj3\_Z-rNgSZMpcdZXZ2d1  
title PyDriveSample.txt, date 2024-10-30T08:17:43.375Z, id 1cxeFNj0XN0x-LaC-Q1ZZz9FEq7XwW5aG  
title PyDriveSample.txt, date 2024-10-30T08:05:35.003Z, id 1RM23RNLMTMvdtCGMQqnPa6sS92uzjaou

```
title cat.txt, date 2021-05-11T08:44:26.422Z, id 1hjoylmceZBDJGP1jXKb5tnohCTpToXZ0
title 030.txt, date 2021-04-27T08:43:18.663Z, id 1jOXFEH0qo6XvpKQzwnDX6Vh8X0eteo7C
```

```
# Download a file based on its file ID.
# A file ID looks like: laggyWshwcyP6kEI-y_W3P8D26sz
file_id = '1RM23RNLMTMVdtCGMQqnPa6sS92uzjaou'
downloaded = drive.CreateFile({'id': file_id})
print('Downloaded content "{}".format(downloaded.GetContentString())
```

🔄 Downloaded content "Sample upload file content 範例"

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
```

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-2.0, 1.0, 4.0, 7.0, 10.0, 13.0], dtype=float)
# 以上code定義此neural network
```

```
model.fit(xs, ys, epochs=100)
```

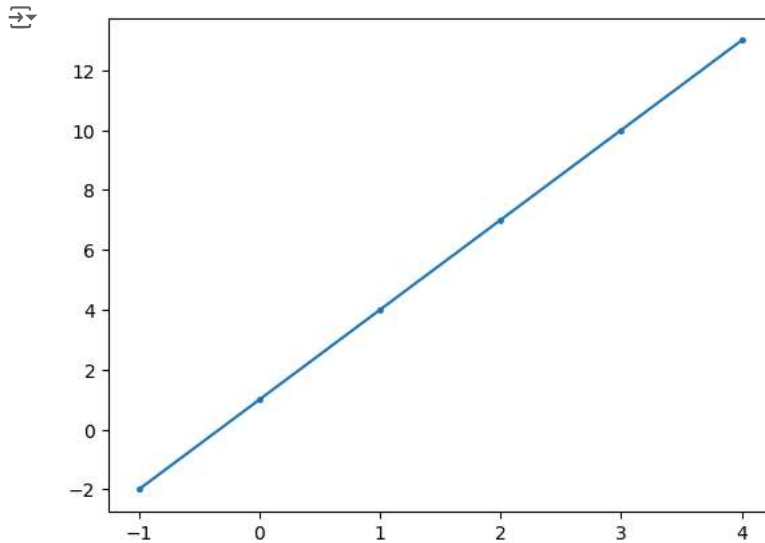
🔄 Epoch 1/100  
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input\_shape`/`input\_dim` argument to a layer.  
super().\_\_init\_\_(activity\_regularizer=activity\_regularizer, \*\*kwargs)

1/1	0s 228ms/step - loss: 46.4915
Epoch 2/100	
1/1	0s 26ms/step - loss: 36.5788
Epoch 3/100	
1/1	0s 30ms/step - loss: 28.7800
Epoch 4/100	
1/1	0s 28ms/step - loss: 22.6442
Epoch 5/100	
1/1	0s 58ms/step - loss: 17.8168
Epoch 6/100	
1/1	0s 27ms/step - loss: 14.0189
Epoch 7/100	
1/1	0s 31ms/step - loss: 11.0308
Epoch 8/100	
1/1	0s 28ms/step - loss: 8.6799
Epoch 9/100	
1/1	0s 58ms/step - loss: 6.8303
Epoch 10/100	
1/1	0s 30ms/step - loss: 5.3751
Epoch 11/100	
1/1	0s 30ms/step - loss: 4.2301
Epoch 12/100	
1/1	0s 35ms/step - loss: 3.3293
Epoch 13/100	
1/1	0s 54ms/step - loss: 2.6206
Epoch 14/100	
1/1	0s 59ms/step - loss: 2.0630
Epoch 15/100	
1/1	0s 35ms/step - loss: 1.6243
Epoch 16/100	
1/1	0s 33ms/step - loss: 1.2791
Epoch 17/100	
1/1	0s 57ms/step - loss: 1.0075
Epoch 18/100	
1/1	0s 36ms/step - loss: 0.7937
Epoch 19/100	
1/1	0s 35ms/step - loss: 0.6256
Epoch 20/100	
1/1	0s 40ms/step - loss: 0.4933
Epoch 21/100	
1/1	0s 60ms/step - loss: 0.3891
Epoch 22/100	
1/1	0s 58ms/step - loss: 0.3072
Epoch 23/100	
1/1	0s 38ms/step - loss: 0.2427
Epoch 24/100	
1/1	0s 32ms/step - loss: 0.1919
Epoch 25/100	
1/1	0s 27ms/step - loss: 0.1520
Epoch 26/100	
1/1	0s 29ms/step - loss: 0.1205
Epoch 27/100	
1/1	0s 28ms/step - loss: 0.0958
Epoch 28/100	

```
import matplotlib.pyplot as plt
from scipy import stats
slope, intercept, r, p, std_err = stats.linregress(xs, ys)

def myfunc(v):
    return slope * v + intercept

mymodel = list(map(myfunc, xs))
plt.scatter(xs, ys, 6)
plt.plot(xs, mymodel)
plt.show()
```



```
print(model.predict([10.0]))
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-36-5458a4003a80> in <cell line: 1>()
----> 1 print(model.predict([10.0]))

----- 1 frames -----
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/_init_.py in get_data_adapter(x, y, sample_weight,
batch_size, steps_per_epoch, shuffle, class_weight)
    118         # )
    119     else:
--> 120         raise ValueError(f"Unrecognized data type: x={x} (of type {type(x)})")
    121
    122

ValueError: Unrecognized data type: x=[10.0] (of type <class 'list'>)
```

後續步驟:

[說明錯誤](#)

```
# 將數據轉換為 2D 結構
input_data = np.array([[10.0]])
print(model.predict(input_data))
```

```
1/1 0s 21ms/step
[[31.094072]]
```

