华 院

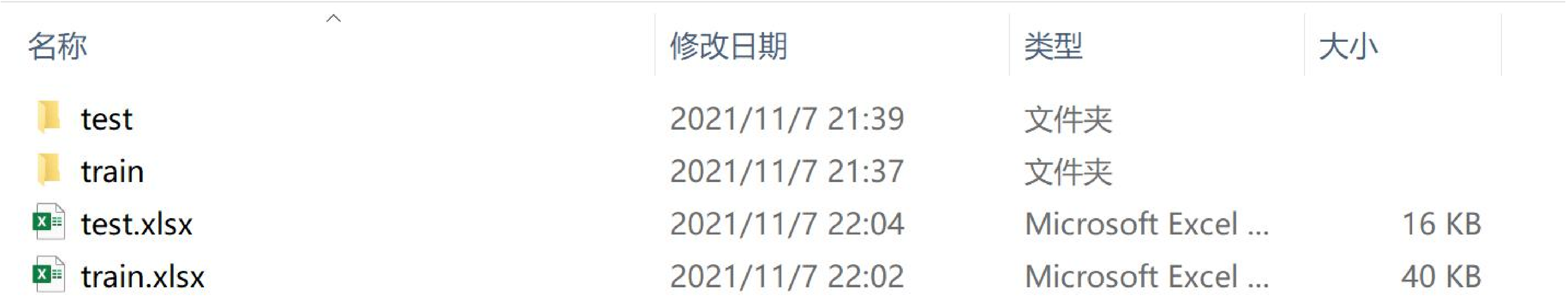
www.UniDT.com

课程项目

2021.11.9

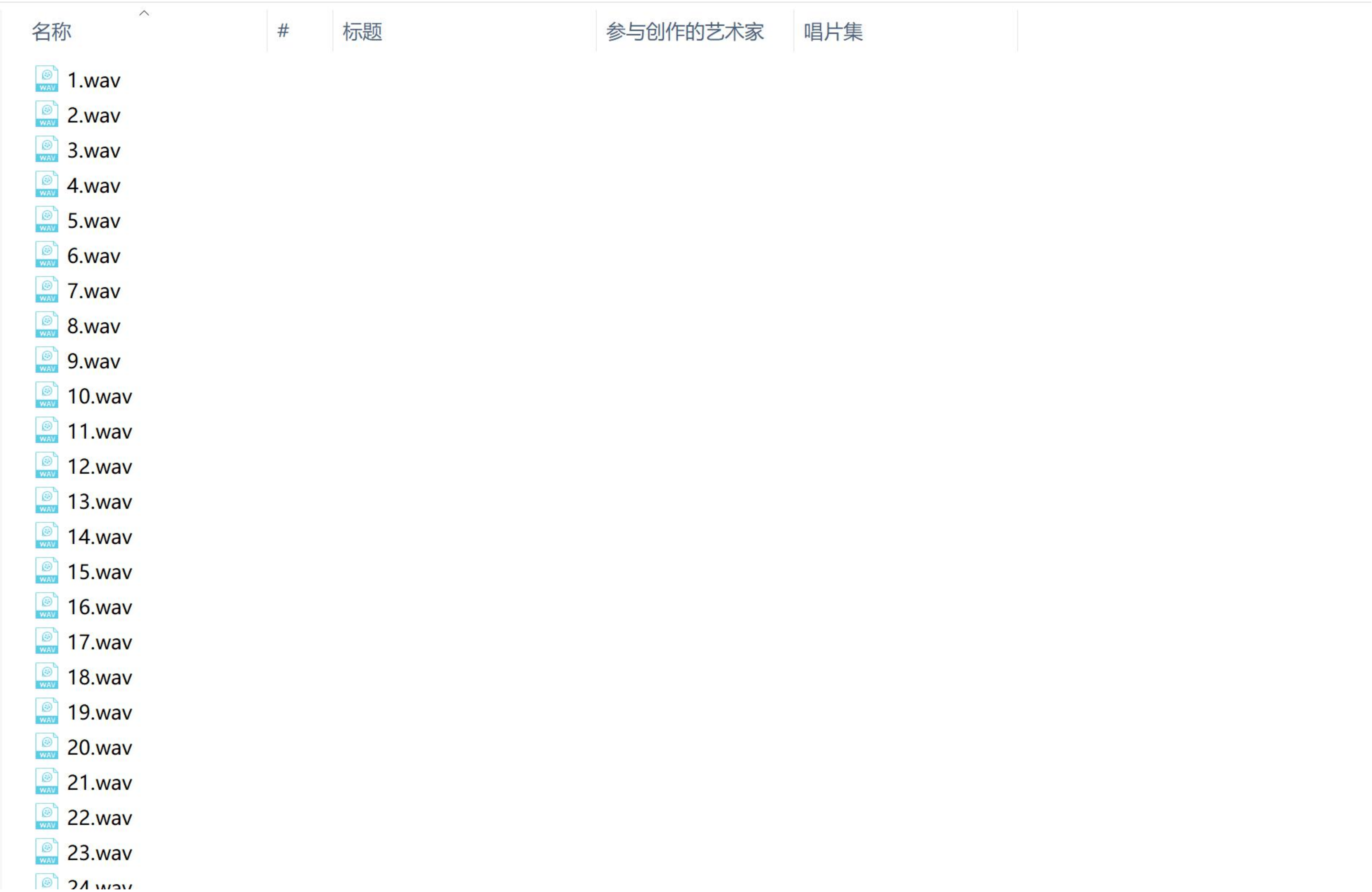
语音性别识别

- 训练集：2000段语音文件，每段语音2-12秒，标签为该语音文件对应说话人的性别
- 测试集：500段语音文件
- 要求：构建一个模型，输入语音文件，输出对应说话人的性别
- 结果提交：1份报告，1份代码，1个在测试集上的结果文件



名称	修改日期	类型	大小
test	2021/11/7 21:39	文件夹	
train	2021/11/7 21:37	文件夹	
test.xlsx	2021/11/7 22:04	Microsoft Excel ...	16 KB
train.xlsx	2021/11/7 22:02	Microsoft Excel ...	40 KB

数据说明——train文件夹



名称	#	标题	参与创作的艺术家	唱片集
1.wav				
2.wav				
3.wav				
4.wav				
5.wav				
6.wav				
7.wav				
8.wav				
9.wav				
10.wav				
11.wav				
12.wav				
13.wav				
14.wav				
15.wav				
16.wav				
17.wav				
18.wav				
19.wav				
20.wav				
21.wav				
22.wav				
23.wav				
24.wav				

- QQ影音
- PotPlayer
- QuickTime
- ……

1.wav—2000.wav

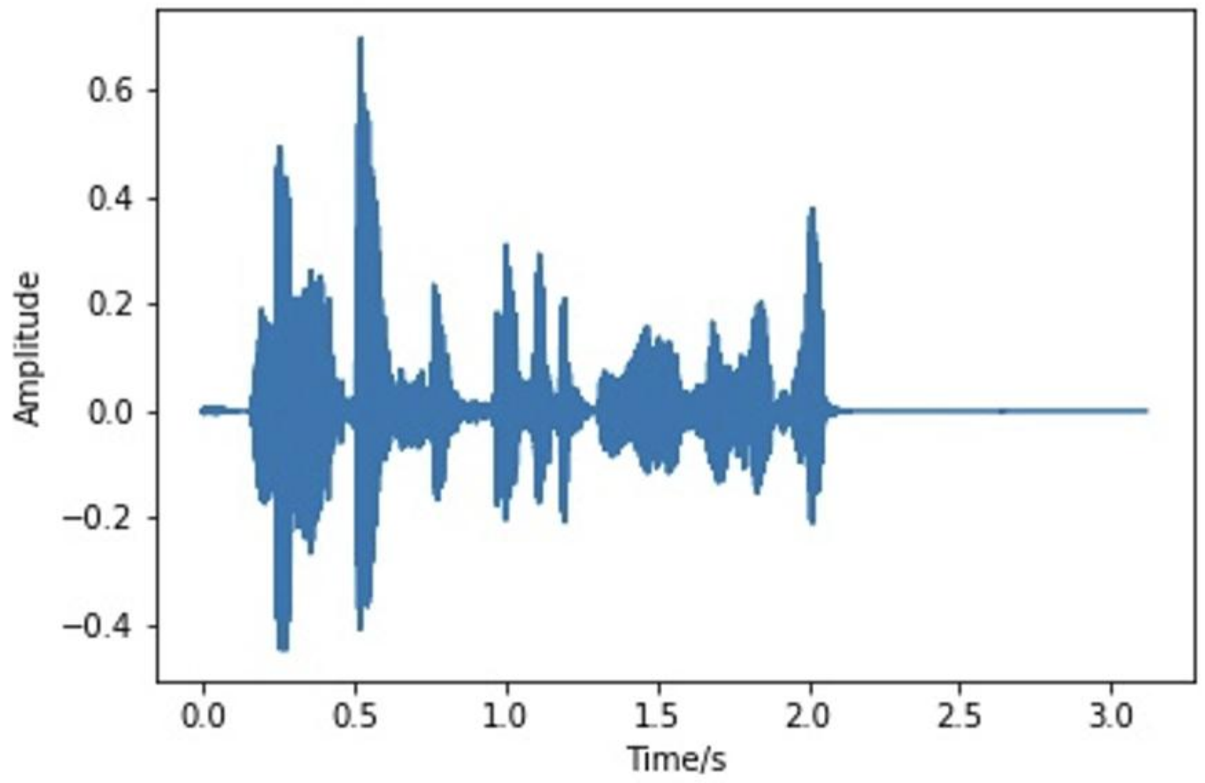
数据说明——train文件夹

1.wav 🔊

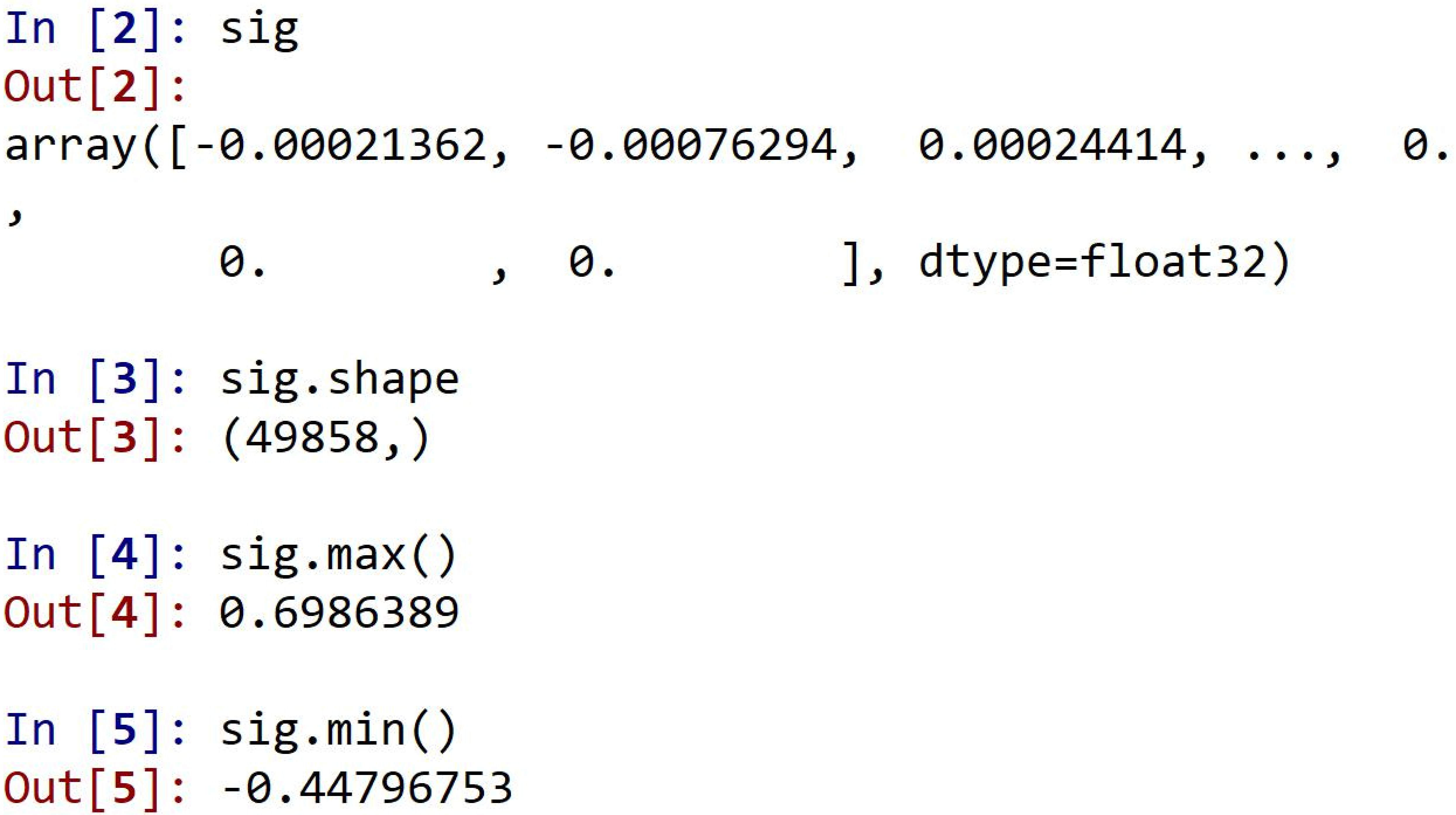
3.wav 🔊

数据说明——train文件夹

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

fs = 16000
sig, _ = librosa.load("1.wav", sr=fs)
time = np.arange(0, len(sig)) * (1.0 / fs)
plt.plot(time, sig, color='C0')
plt.xlabel("Time/s")
plt.ylabel("Amplitude")
plt.savefig('1.jpg')
```

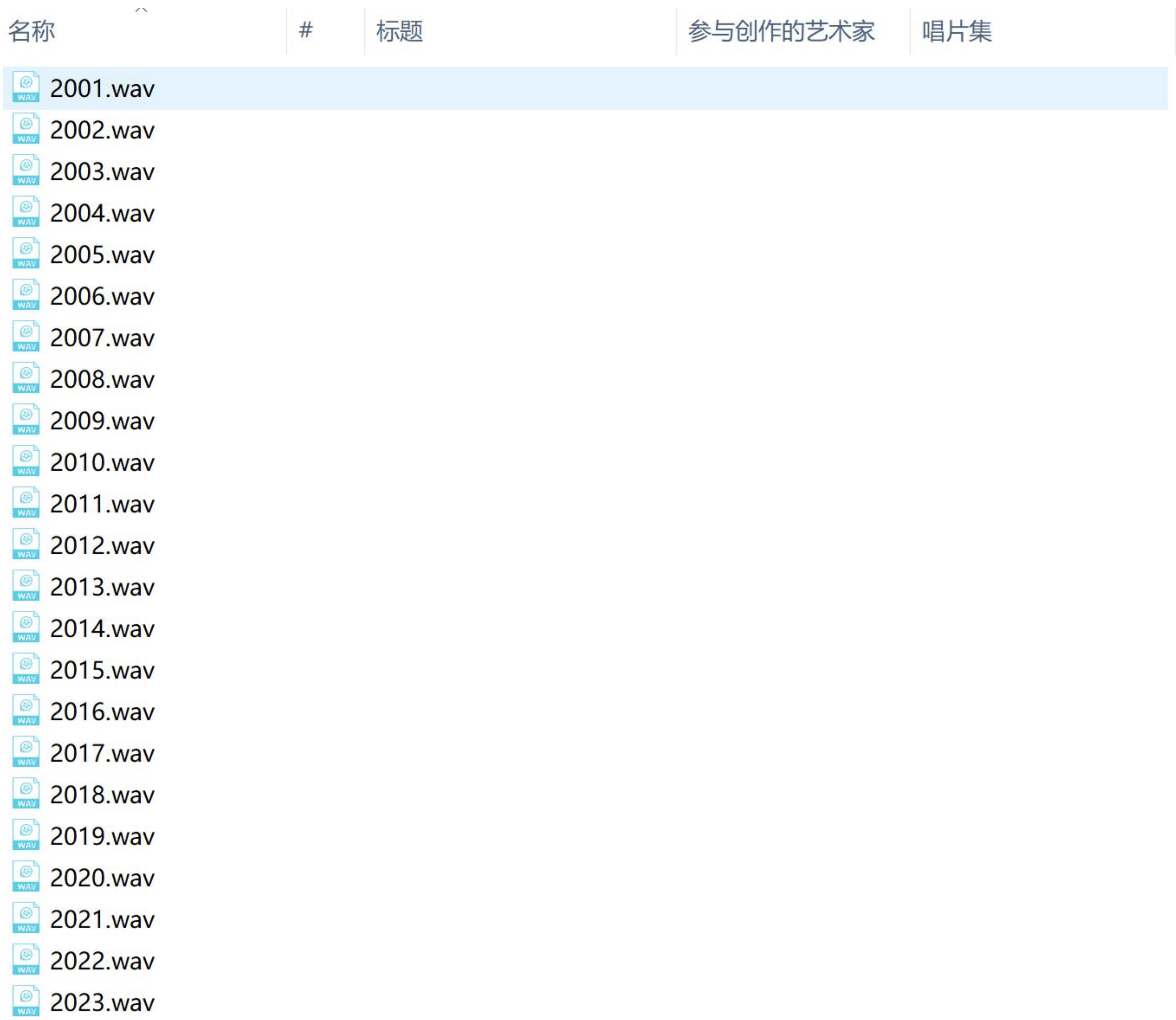


数据说明——train文件夹



```
In [2]: sig
Out[2]:
array([-0.00021362, -0.00076294,  0.00024414, ...,  0.
,
        0.        ,  0.        ], dtype=float32)

In [3]: sig.shape
Out[3]: (49858,)

In [4]: sig.max()
Out[4]: 0.6986389

In [5]: sig.min()
Out[5]: -0.44796753
```

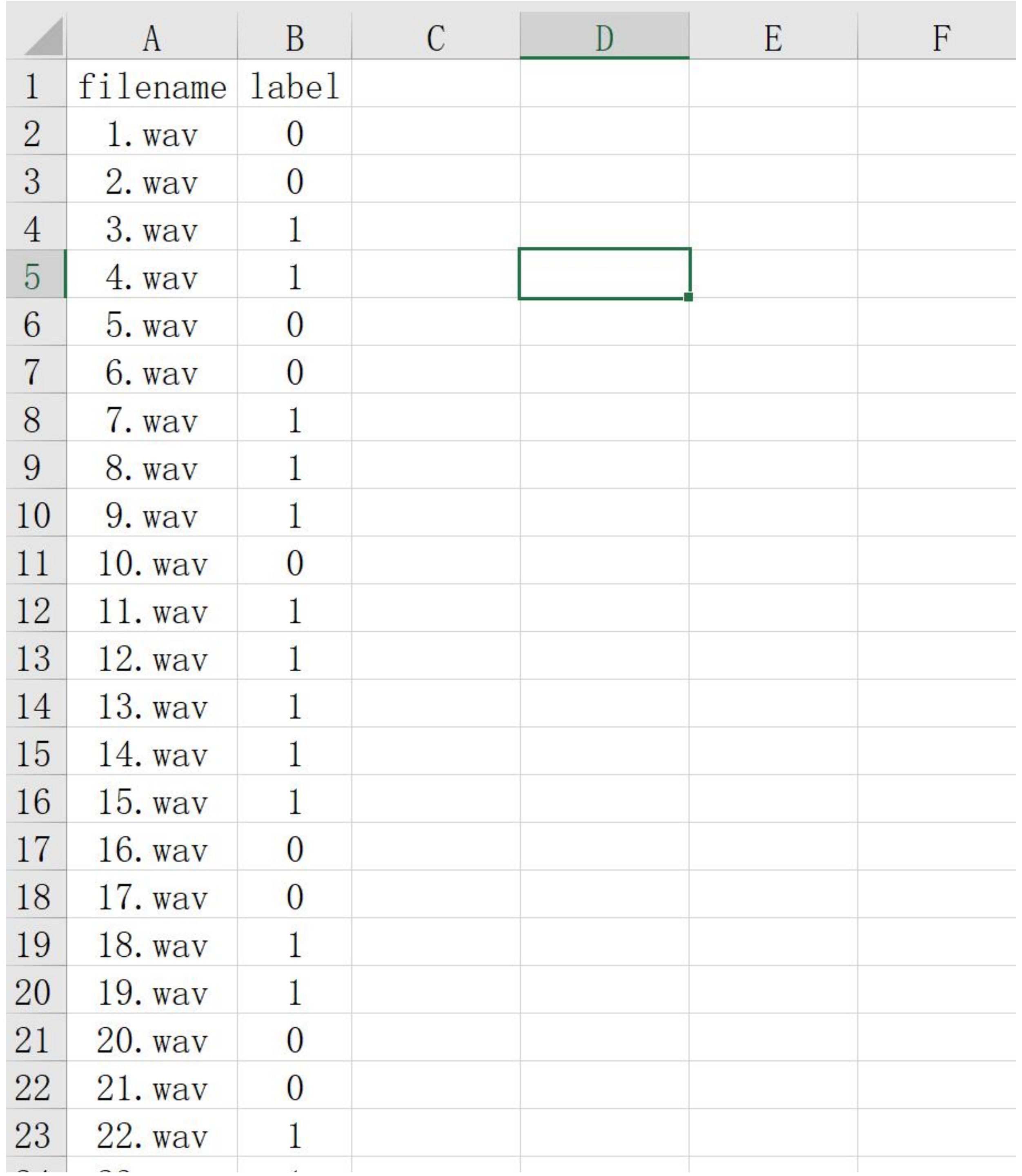
数据说明——test文件夹



名称	#	标题	参与创作的艺术家	唱片集
2001.wav				
2002.wav				
2003.wav				
2004.wav				
2005.wav				
2006.wav				
2007.wav				
2008.wav				
2009.wav				
2010.wav				
2011.wav				
2012.wav				
2013.wav				
2014.wav				
2015.wav				
2016.wav				
2017.wav				
2018.wav				
2019.wav				
2020.wav				
2021.wav				
2022.wav				
2023.wav				

2001.wav—2500.wav

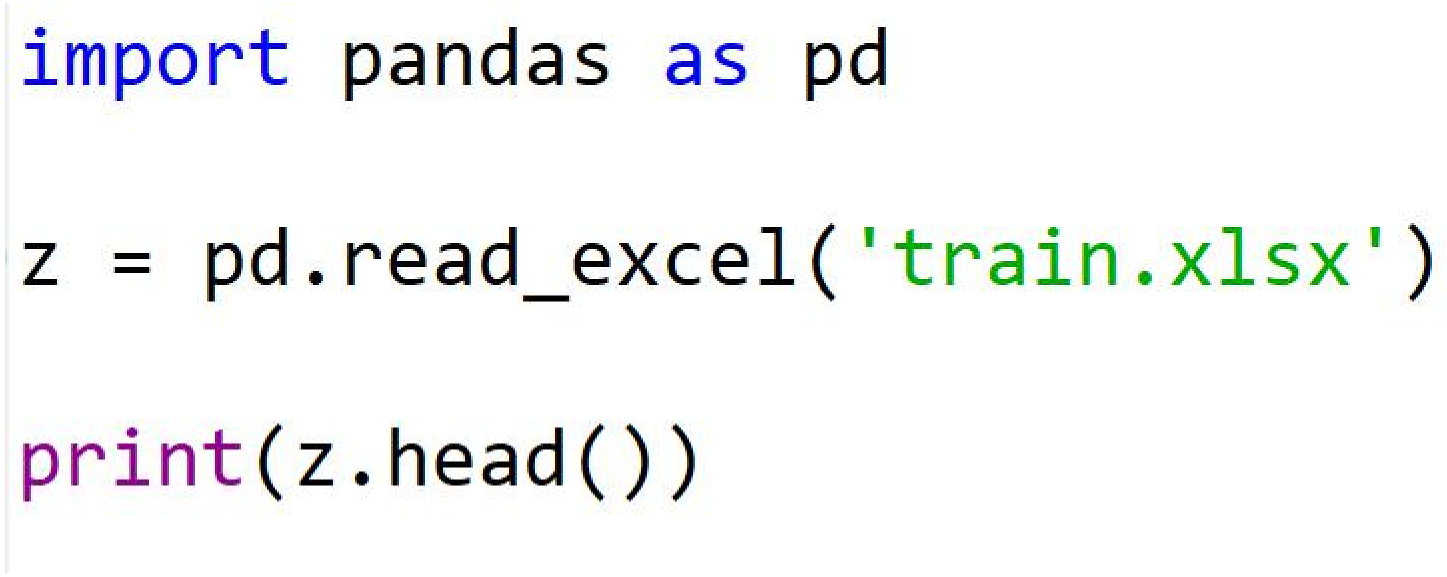
数据说明——train.xlsx



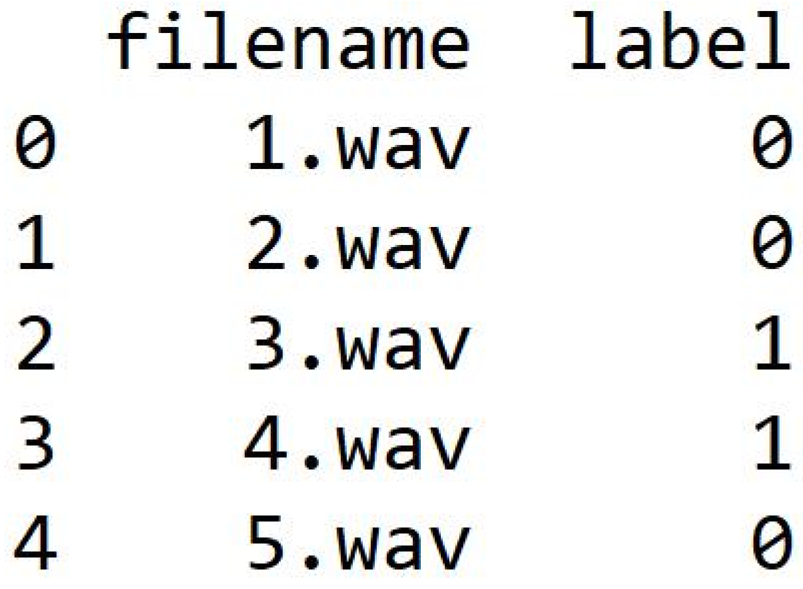
	A	B	C	D	E	F
1	filename	label				
2	1.wav	0				
3	2.wav	0				
4	3.wav	1				
5	4.wav	1				
6	5.wav	0				
7	6.wav	0				
8	7.wav	1				
9	8.wav	1				
10	9.wav	1				
11	10.wav	0				
12	11.wav	1				
13	12.wav	1				
14	13.wav	1				
15	14.wav	1				
16	15.wav	1				
17	16.wav	0				
18	17.wav	0				
19	18.wav	1				
20	19.wav	1				
21	20.wav	0				
22	21.wav	0				
23	22.wav	1				

0代表女性，1代表男性

数据说明——train.xlsx

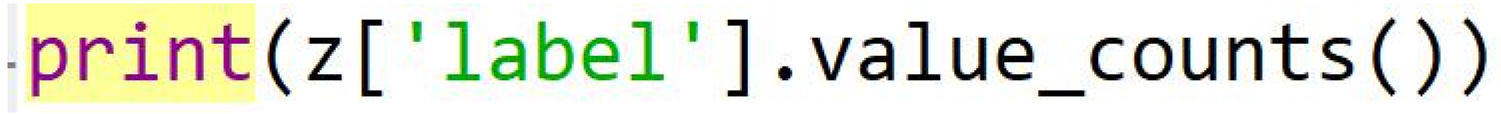


```python
import pandas as pd

z = pd.read_excel('train.xlsx')

print(z.head())
```

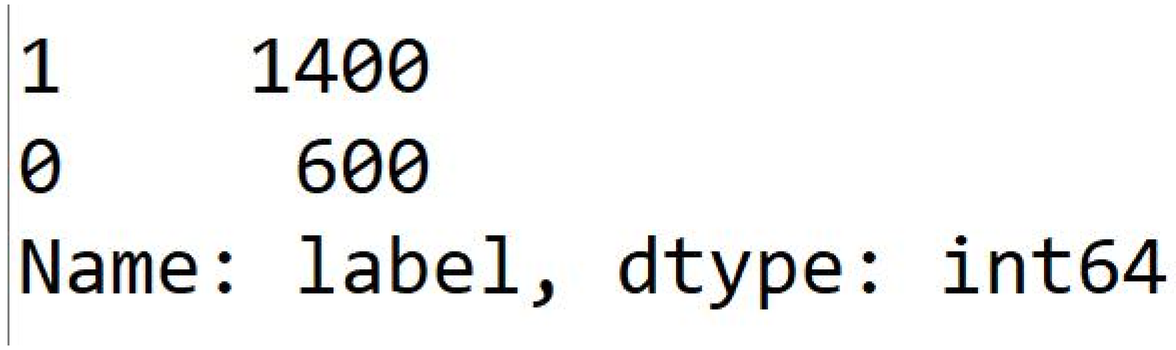


```
  filename  label
0    1.wav      0
1    2.wav      0
2    3.wav      1
3    4.wav      1
4    5.wav      0
```

数据说明——train.xlsx

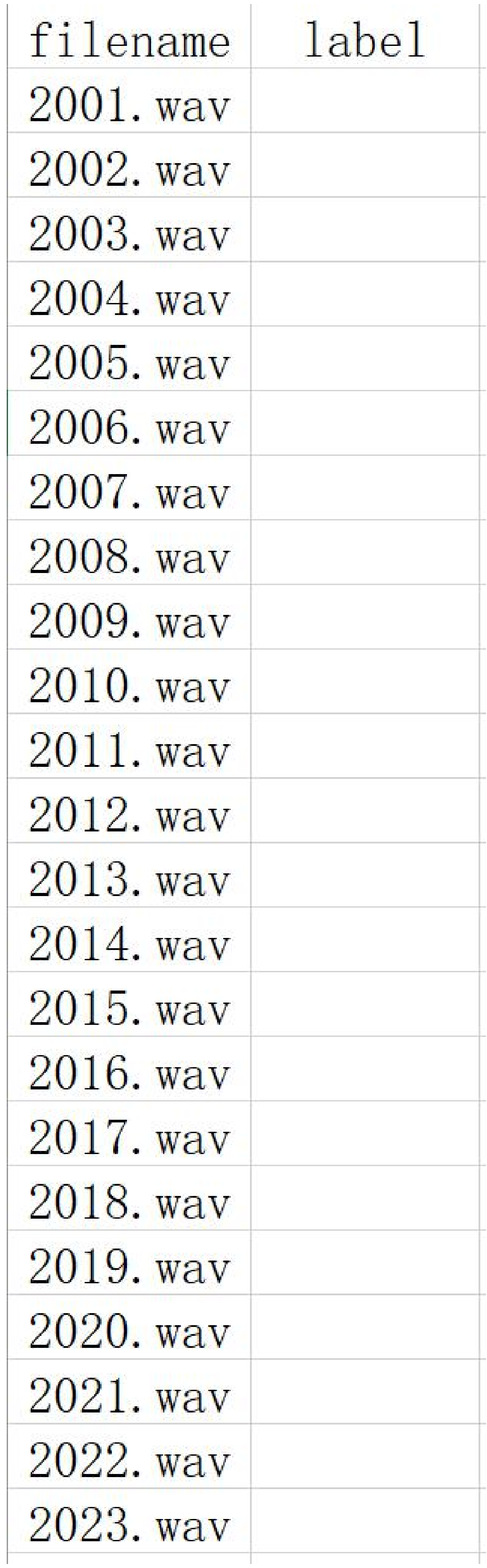


```
print(z['label'].value_counts())
```



```
1    1400
0     600
Name: label, dtype: int64
```

数据说明——test.xlsx

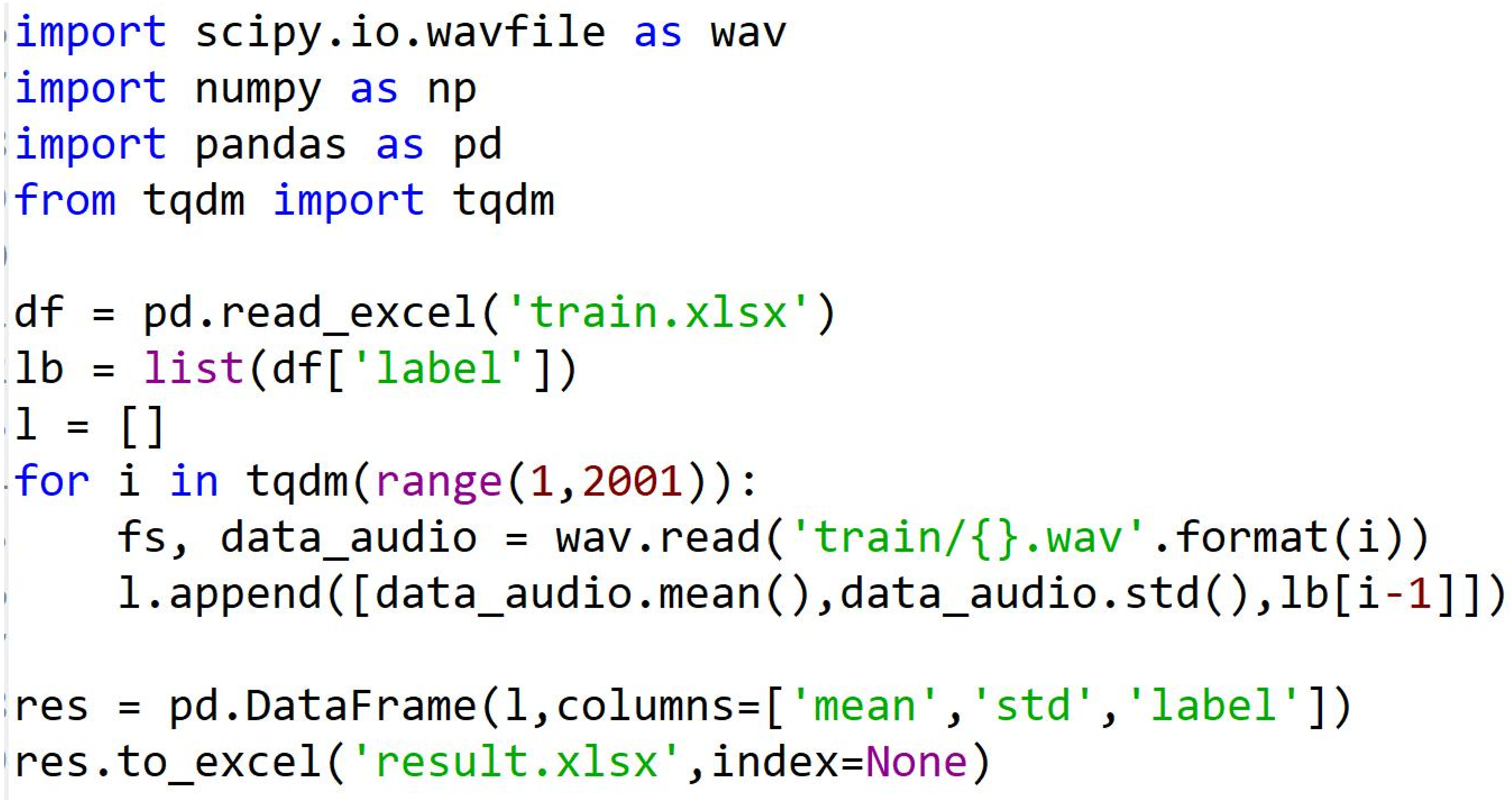


filename	label
2001.wav	
2002.wav	
2003.wav	
2004.wav	
2005.wav	
2006.wav	
2007.wav	
2008.wav	
2009.wav	
2010.wav	
2011.wav	
2012.wav	
2013.wav	
2014.wav	
2015.wav	
2016.wav	
2017.wav	
2018.wav	
2019.wav	
2020.wav	
2021.wav	
2022.wav	
2023.wav	

结果提交说明

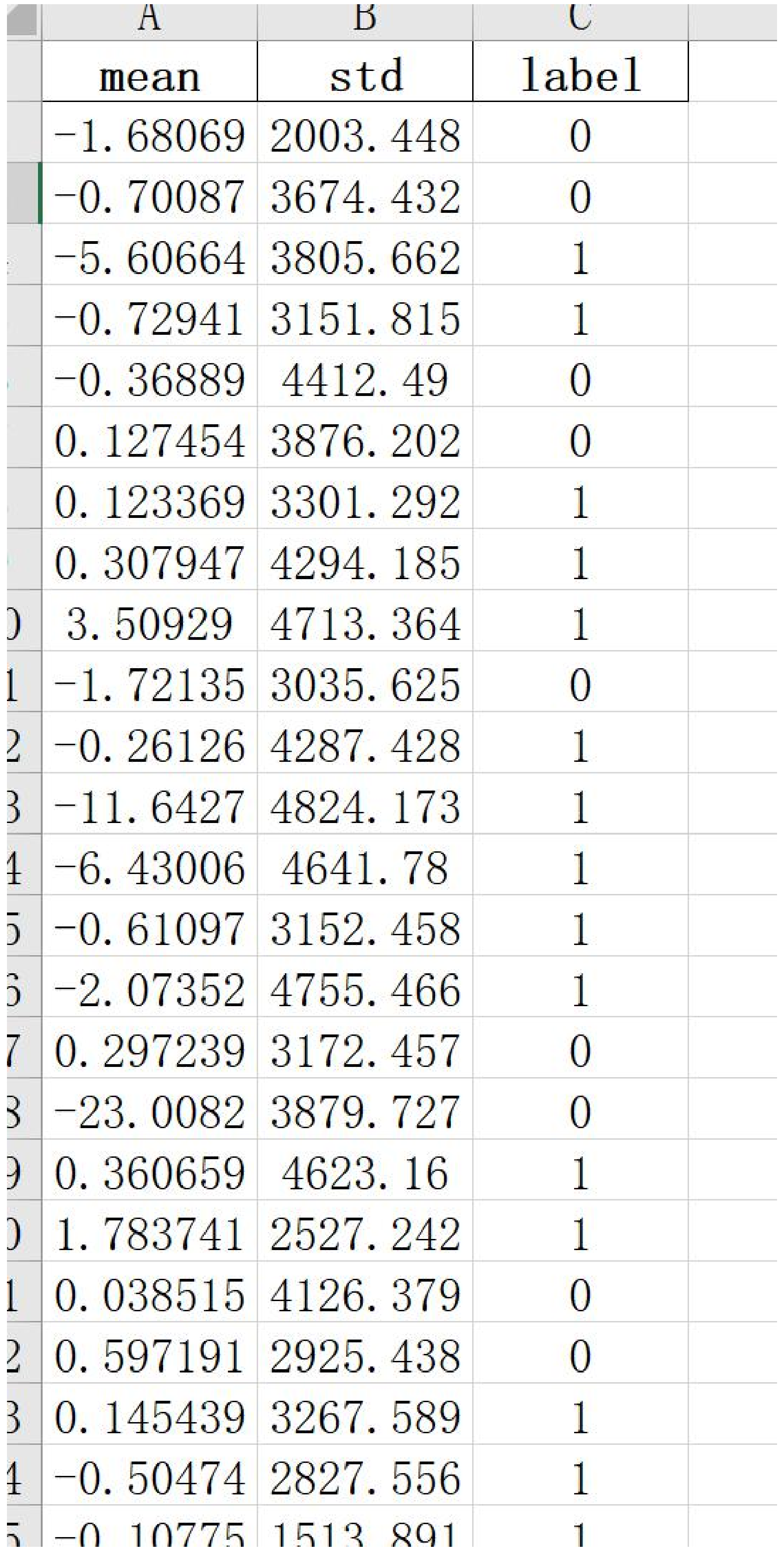
- 1份报告（docx或pdf）
- 1份代码（py文件）
- 1个在测试集上的结果文件（在test.xlsx中打上label列的值，在代码中要能体现label值是模型打出的）

数据探索：波形的均值、方差



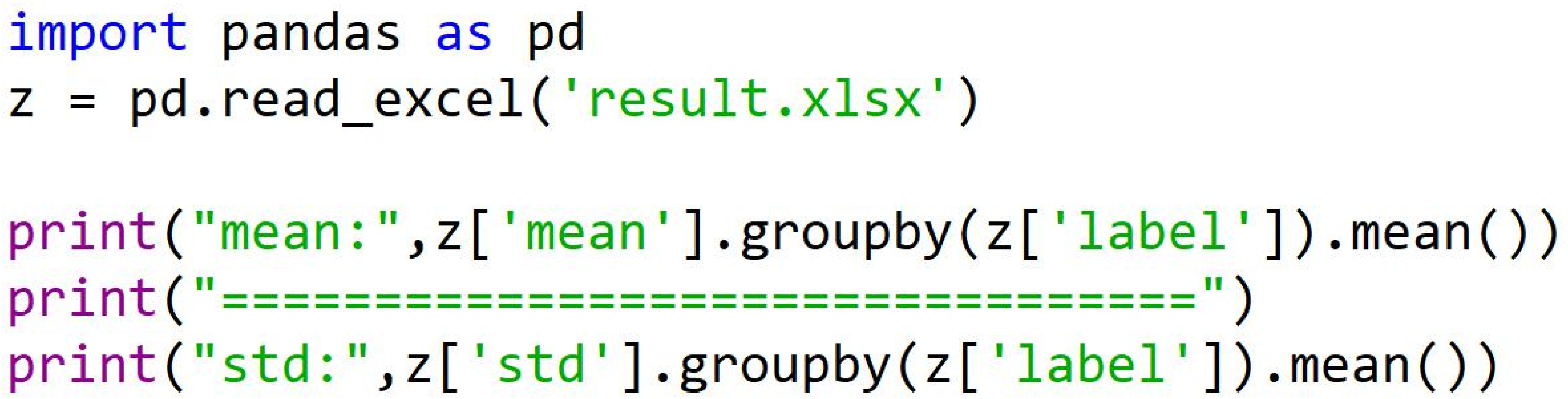
```
import scipy.io.wavfile as wav
import numpy as np
import pandas as pd
from tqdm import tqdm

df = pd.read_excel('train.xlsx')
lb = list(df['label'])
l = []
for i in tqdm(range(1,2001)):
    fs, data_audio = wav.read('train/{}.wav'.format(i))
    l.append([data_audio.mean(),data_audio.std(),lb[i-1]])

res = pd.DataFrame(l,columns=['mean','std','label'])
res.to_excel('result.xlsx',index=None)
```

数据探索：波形的均值、方差

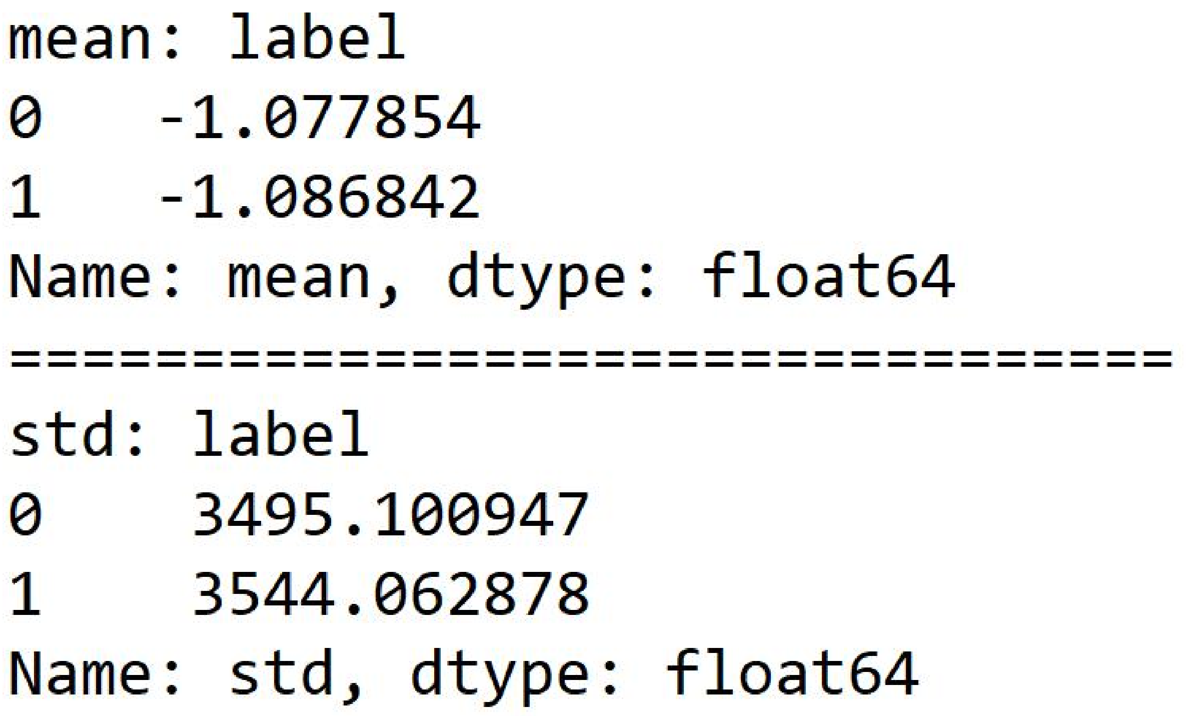


A	B	C
mean	std	label
-1.68069	2003.448	0
-0.70087	3674.432	0
-5.60664	3805.662	1
-0.72941	3151.815	1
-0.36889	4412.49	0
0.127454	3876.202	0
0.123369	3301.292	1
0.307947	4294.185	1
3.50929	4713.364	1
-1.72135	3035.625	0
-0.26126	4287.428	1
-11.6427	4824.173	1
-6.43006	4641.78	1
-0.61097	3152.458	1
-2.07352	4755.466	1
0.297239	3172.457	0
-23.0082	3879.727	0
0.360659	4623.16	1
1.783741	2527.242	1
0.038515	4126.379	0
0.597191	2925.438	0
0.145439	3267.589	1
-0.50474	2827.556	1
-0.10775	1513.891	1

数据探索：波形的均值、方差

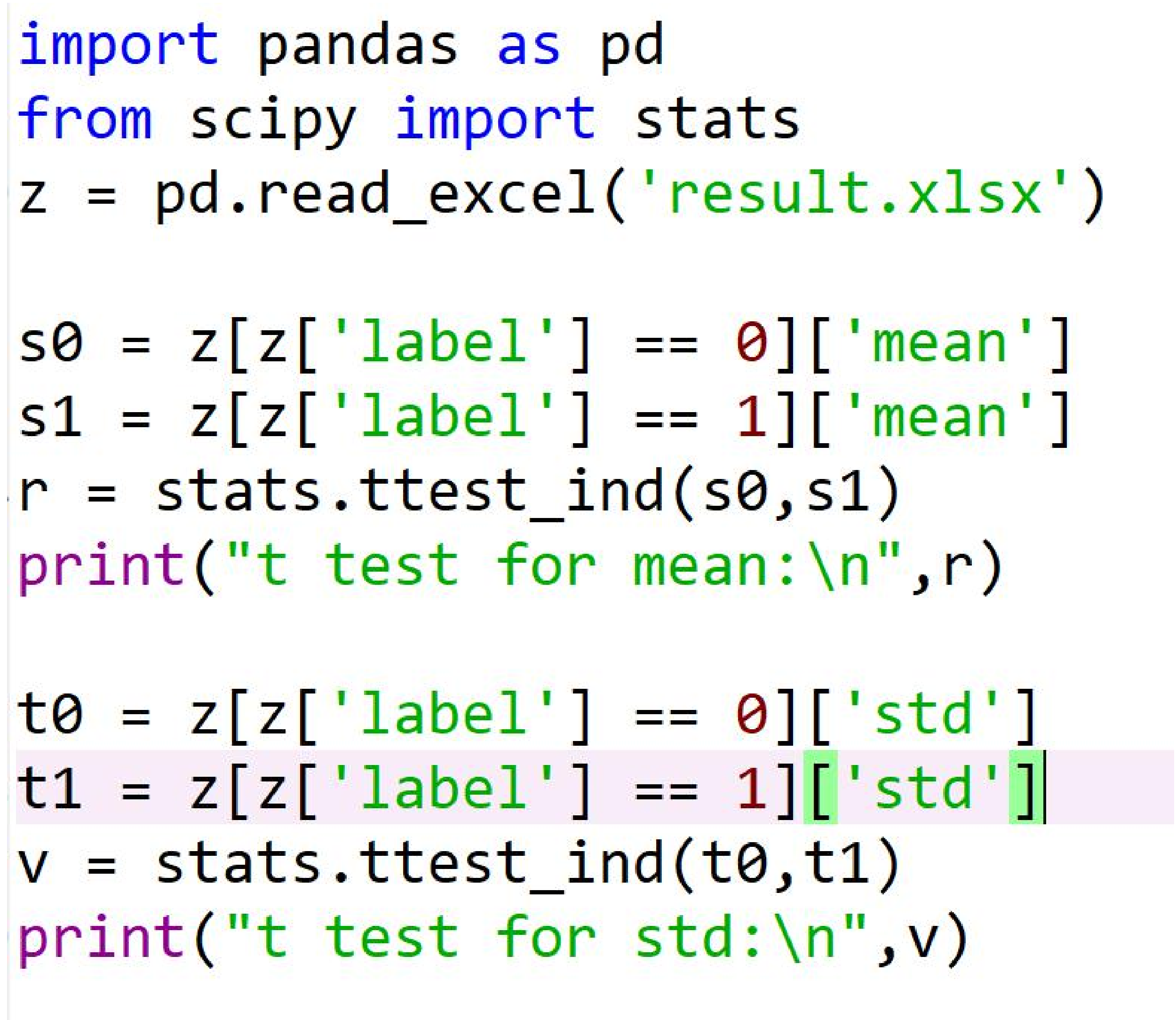


```python
import pandas as pd
z = pd.read_excel('result.xlsx')

print("mean:",z['mean'].groupby(z['label']).mean())
print("================================")
print("std:",z['std'].groupby(z['label']).mean())
```

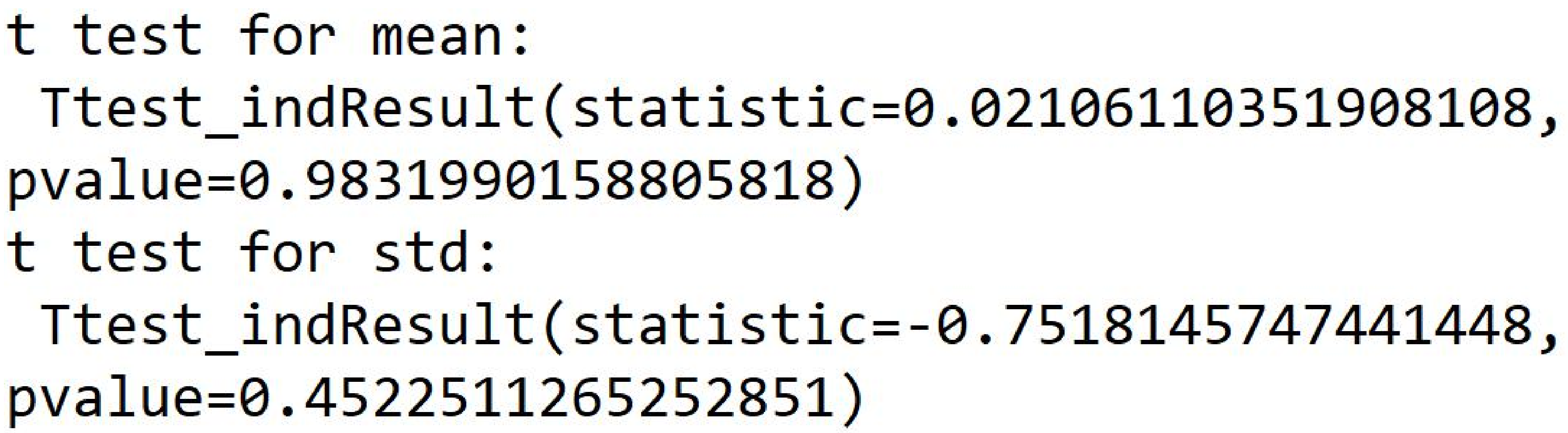


```
mean: label
0   -1.077854
1   -1.086842
Name: mean, dtype: float64
================================
std: label
0    3495.100947
1    3544.062878
Name: std, dtype: float64
```

数据探索：波形的均值、方差

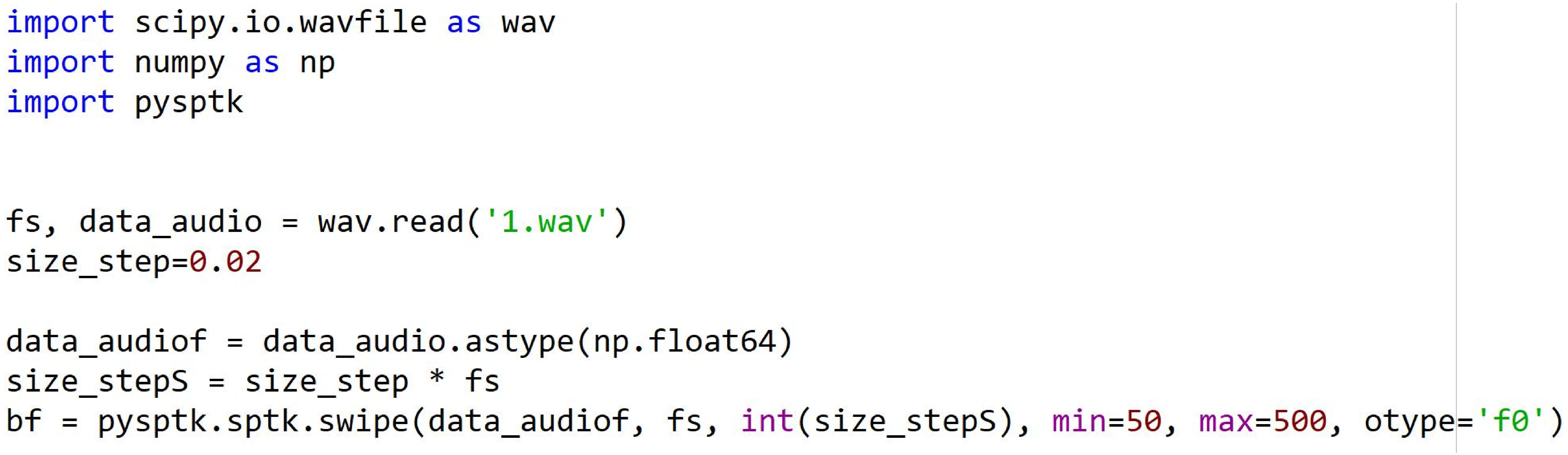


```
import pandas as pd
from scipy import stats
z = pd.read_excel('result.xlsx')

s0 = z[z['label'] == 0]['mean']
s1 = z[z['label'] == 1]['mean']
r = stats.ttest_ind(s0,s1)
print("t test for mean:\n",r)

t0 = z[z['label'] == 0]['std']
t1 = z[z['label'] == 1]['std']
v = stats.ttest_ind(t0,t1)
print("t test for std:\n",v)
```



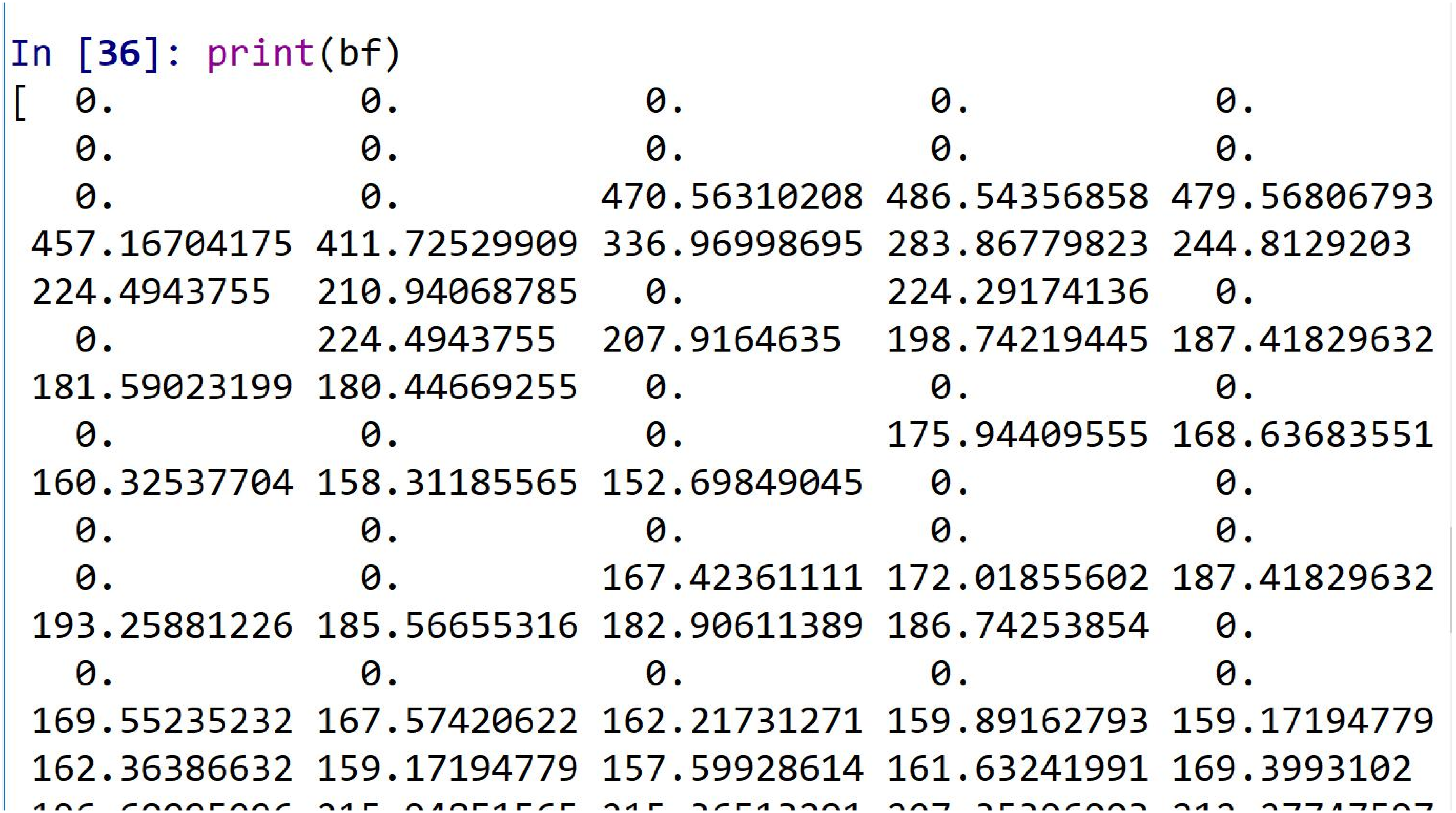
```
t test for mean:
 Ttest_indResult(statistic=0.02106110351908108,
pvalue=0.9831990158805818)
t test for std:
 Ttest_indResult(statistic=-0.7518145747441448,
pvalue=0.4522511265252851)
```

数据探索：计算基频



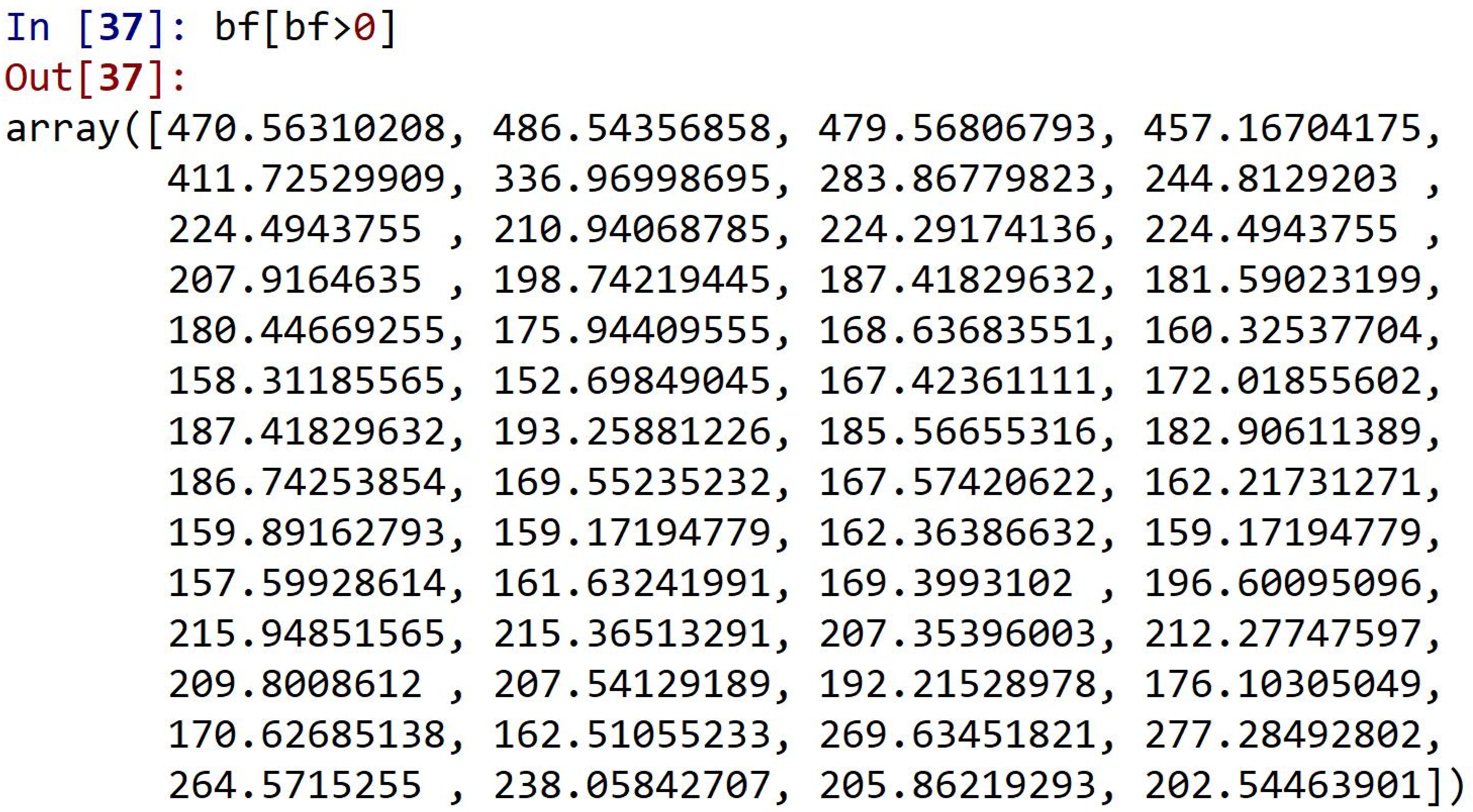
```
import scipy.io.wavfile as wav
import numpy as np
import pysptk


fs, data_audio = wav.read('1.wav')
size_step=0.02

data_audiof = data_audio.astype(np.float64)
size_stepS = size_step * fs
bf = pysptk.sptk.swipe(data_audiof, fs, int(size_stepS), min=50, max=500, otype='f0')
```

数据探索：计算基频



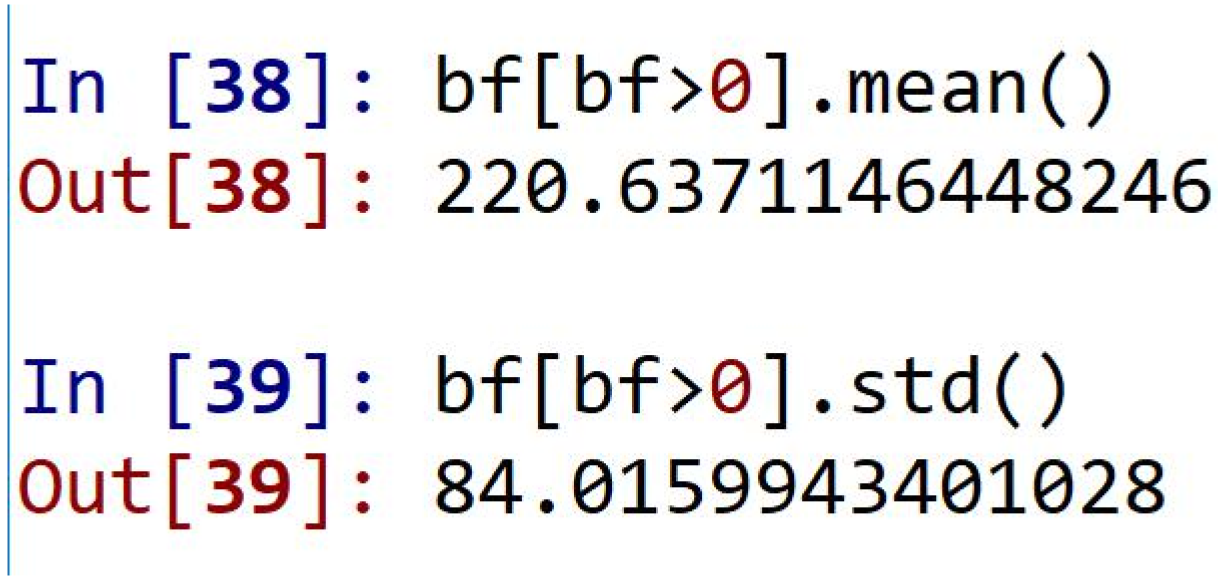
```
In [36]: print(bf)
[   0.            0.            0.            0.            0.
    0.            0.            0.            0.            0.
    0.            0.          470.56310208 486.54356858 479.56806793
  457.16704175 411.72529909 336.96998695 283.86779823 244.8129203
  224.4943755  210.94068785   0.          224.29174136   0.
    0.          224.4943755  207.9164635  198.74219445 187.41829632
  181.59023199 180.44669255   0.            0.            0.
    0.            0.            0.          175.94409555 168.63683551
  160.32537704 158.31185565 152.69849045   0.            0.
    0.            0.            0.            0.            0.
    0.            0.          167.42361111 172.01855602 187.41829632
  193.25881226 185.56655316 182.90611389 186.74253854   0.
    0.            0.            0.            0.            0.
  169.55235232 167.57420622 162.21731271 159.89162793 159.17194779
  162.36386632 159.17194779 157.59928614 161.63241991 169.3993102
```

数据探索：计算基频



```
In [37]: bf[bf>0]
Out[37]:
array([470.56310208, 486.54356858, 479.56806793, 457.16704175,
       411.72529909, 336.96998695, 283.86779823, 244.8129203 ,
       224.4943755 , 210.94068785, 224.29174136, 224.4943755 ,
       207.9164635 , 198.74219445, 187.41829632, 181.59023199,
       180.44669255, 175.94409555, 168.63683551, 160.32537704,
       158.31185565, 152.69849045, 167.42361111, 172.01855602,
       187.41829632, 193.25881226, 185.56655316, 182.90611389,
       186.74253854, 169.55235232, 167.57420622, 162.21731271,
       159.89162793, 159.17194779, 162.36386632, 159.17194779,
       157.59928614, 161.63241991, 169.3993102 , 196.60095096,
       215.94851565, 215.36513291, 207.35396003, 212.27747597,
       209.8008612 , 207.54129189, 192.21528978, 176.10305049,
       170.62685138, 162.51055233, 269.63451821, 277.28492802,
       264.5715255 , 238.05842707, 205.86219293, 202.54463901])
```

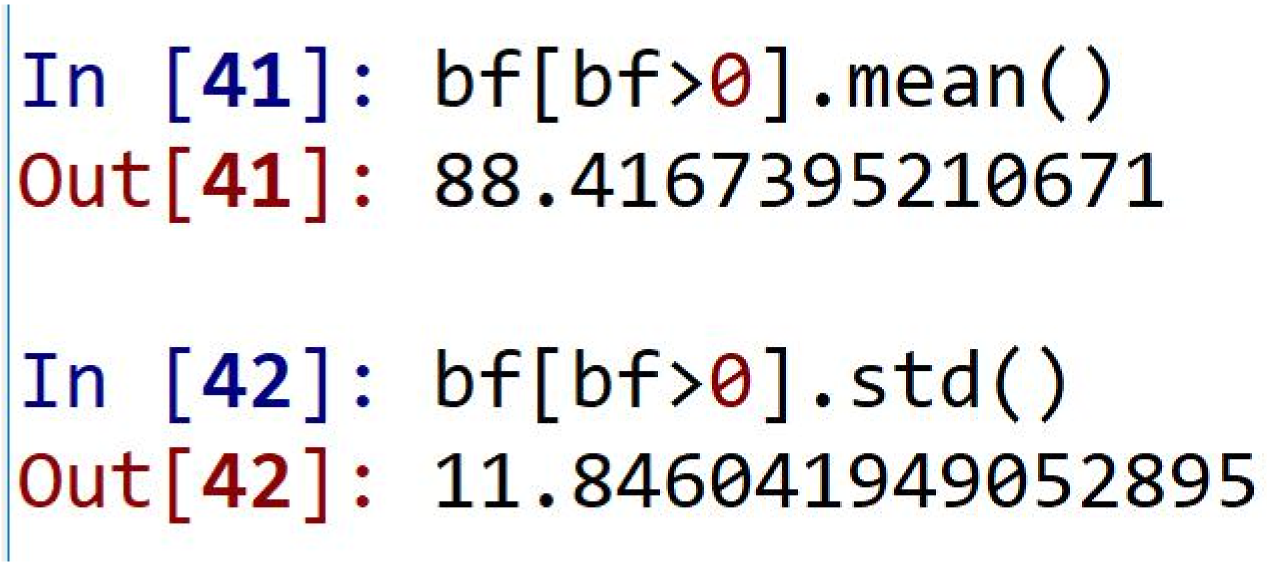
数据探索：计算平均基频



```
In [38]: bf[bf>0].mean()
Out[38]: 220.6371146448246

In [39]: bf[bf>0].std()
Out[39]: 84.0159943401028
```

1.wav 对应女性

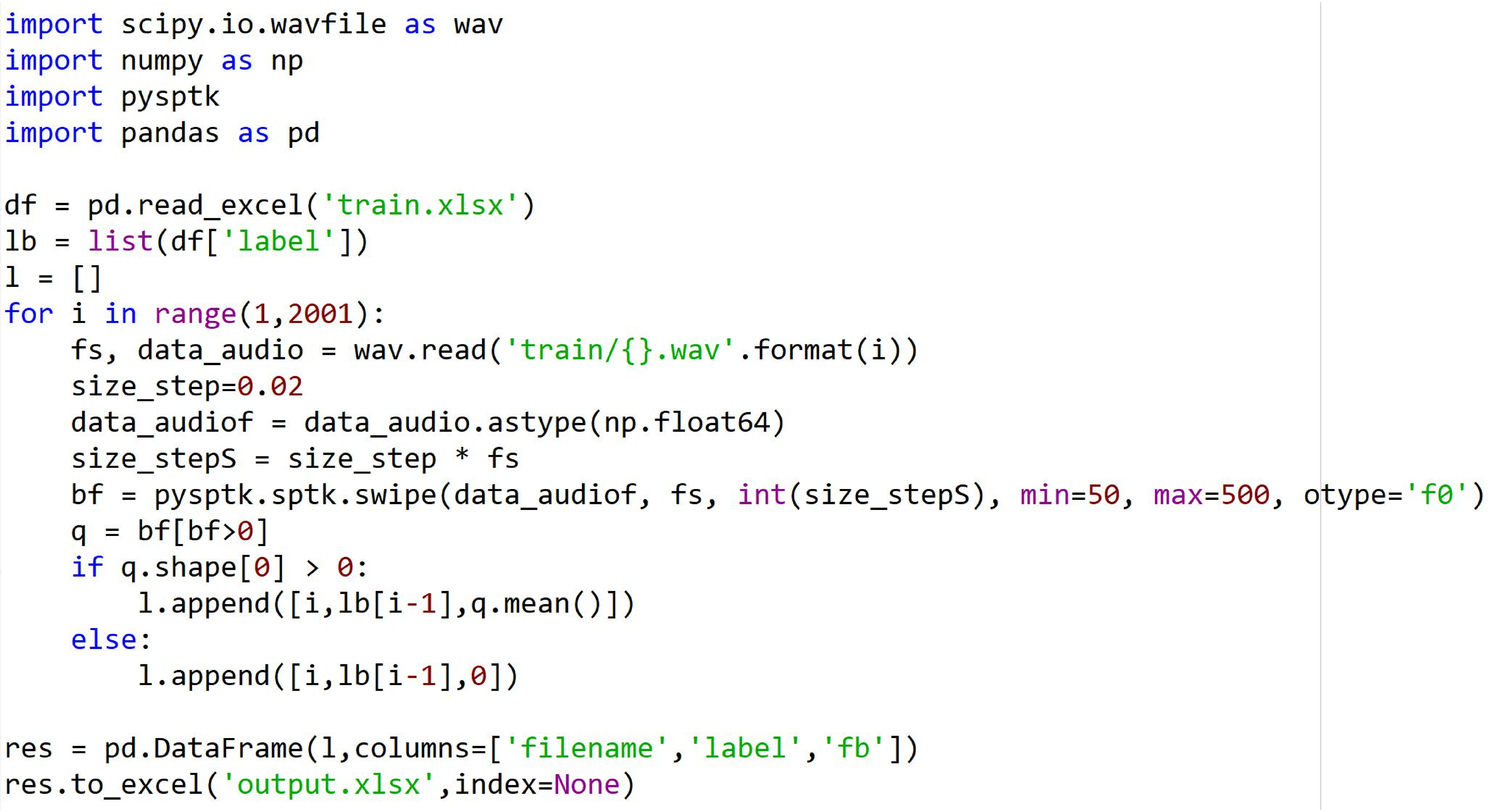
数据探索：计算平均基频



```
In [41]: bf[bf>0].mean()
Out[41]: 88.4167395210671

In [42]: bf[bf>0].std()
Out[42]: 11.846041949052895
```

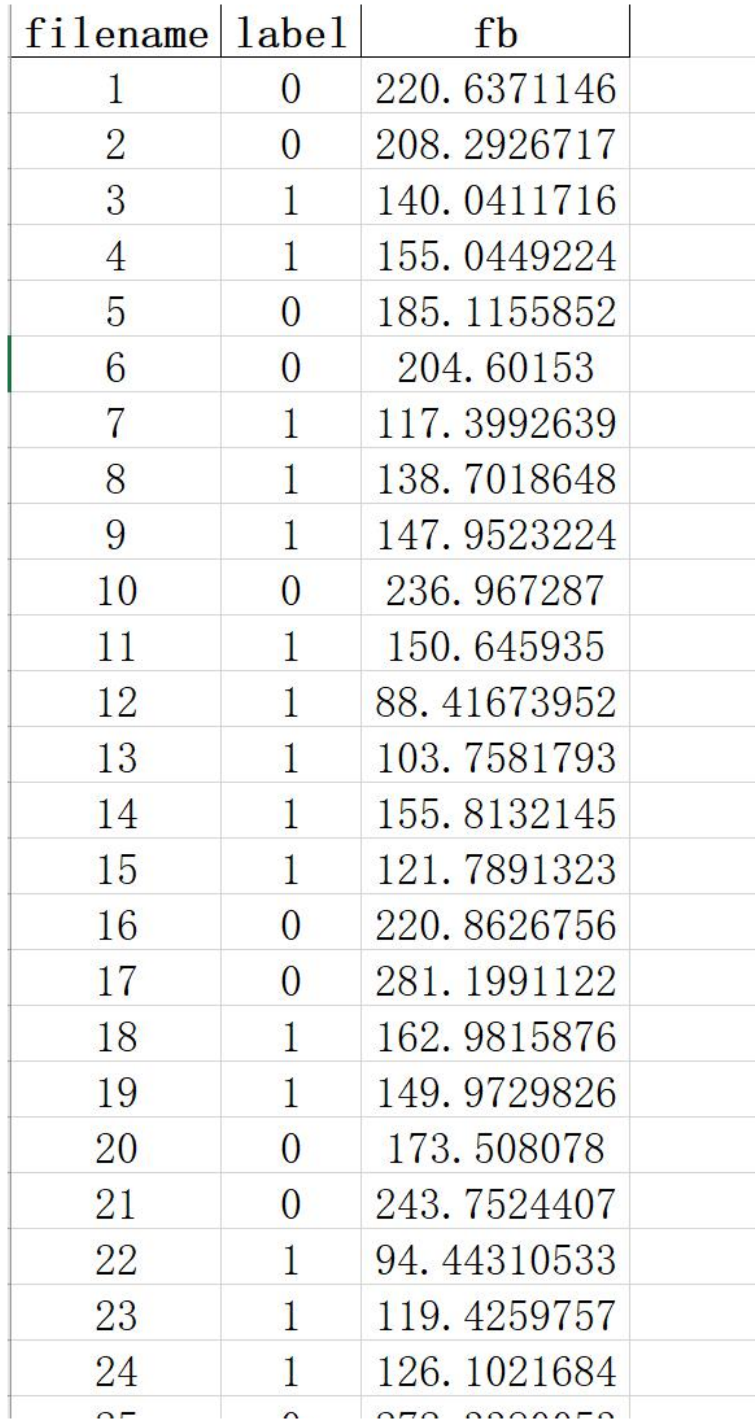
12.wav 对应男性

数据探索：计算所有人的平均基频



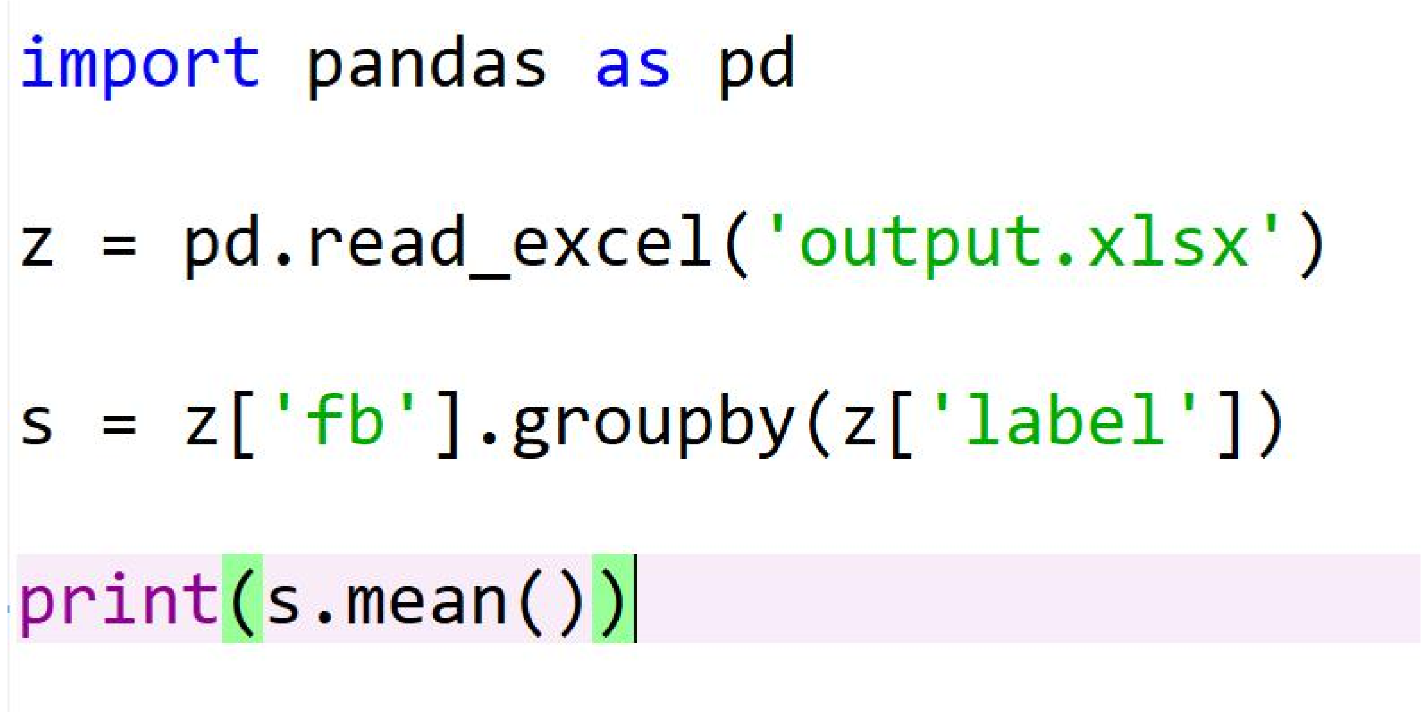
```python
import scipy.io.wavfile as wav
import numpy as np
import pysptk
import pandas as pd

df = pd.read_excel('train.xlsx')
lb = list(df['label'])
l = []
for i in range(1,2001):
    fs, data_audio = wav.read('train/{}.wav'.format(i))
    size_step=0.02
    data_audiof = data_audio.astype(np.float64)
    size_stepS = size_step * fs
    bf = pysptk.sptk.swipe(data_audiof, fs, int(size_stepS), min=50, max=500, otype='f0')
    q = bf[bf>0]
    if q.shape[0] > 0:
        l.append([i,lb[i-1],q.mean()])
    else:
        l.append([i,lb[i-1],0])

res = pd.DataFrame(l,columns=['filename','label','fb'])
res.to_excel('output.xlsx',index=None)
```

数据探索：计算所有人的平均基频

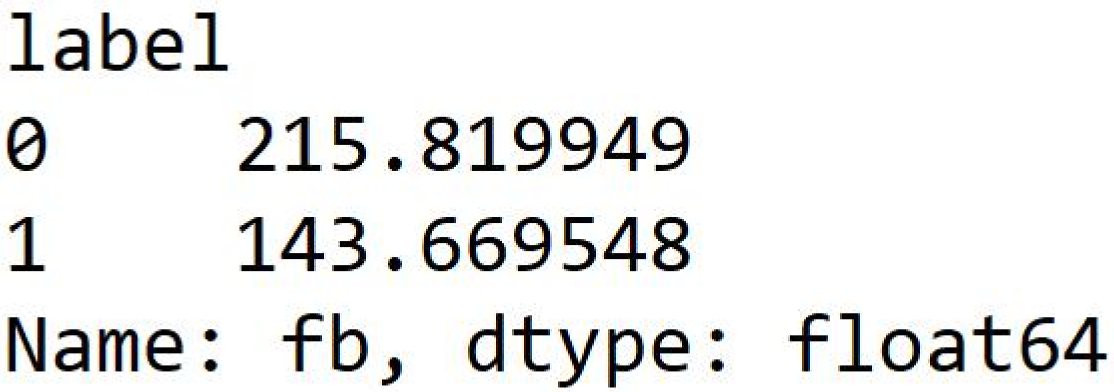


filename	label	fb
1	0	220.6371146
2	0	208.2926717
3	1	140.0411716
4	1	155.0449224
5	0	185.1155852
6	0	204.60153
7	1	117.3992639
8	1	138.7018648
9	1	147.9523224
10	0	236.967287
11	1	150.645935
12	1	88.41673952
13	1	103.7581793
14	1	155.8132145
15	1	121.7891323
16	0	220.8626756
17	0	281.1991122
18	1	162.9815876
19	1	149.9729826
20	0	173.508078
21	0	243.7524407
22	1	94.44310533
23	1	119.4259757
24	1	126.1021684

数据探索：计算所有人的平均基频

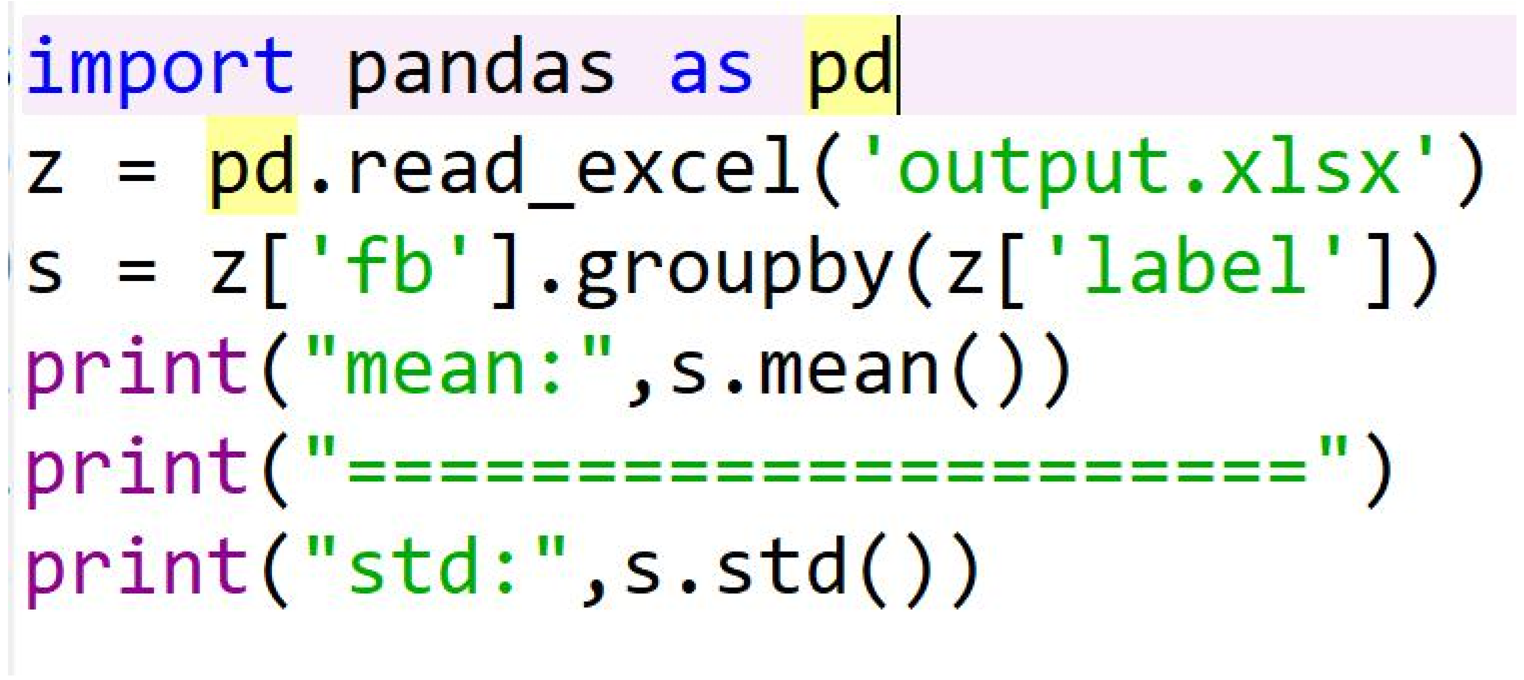


```
import pandas as pd

z = pd.read_excel('output.xlsx')

s = z['fb'].groupby(z['label'])

print(s.mean())
```

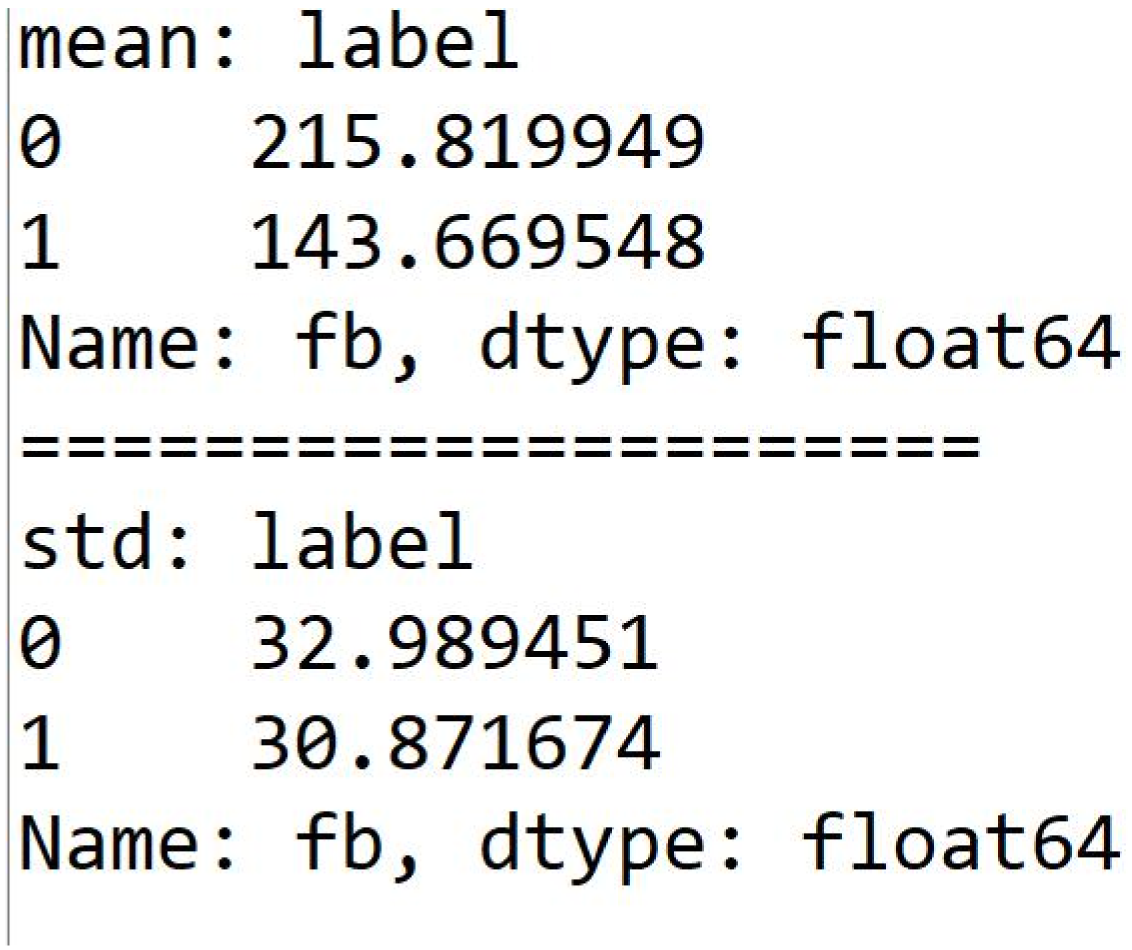


```
label
0     215.819949
1     143.669548
Name: fb, dtype: float64
```

数据探索：计算所有人的平均基频

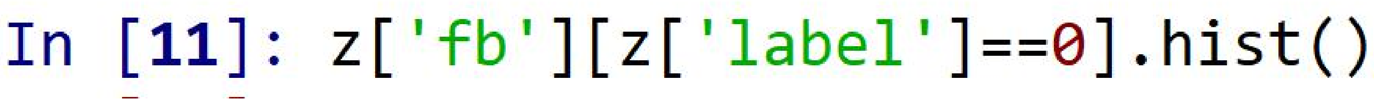


```python
import pandas as pd
z = pd.read_excel('output.xlsx')
s = z['fb'].groupby(z['label'])
print("mean:",s.mean())
print("=====================")
print("std:",s.std())
```

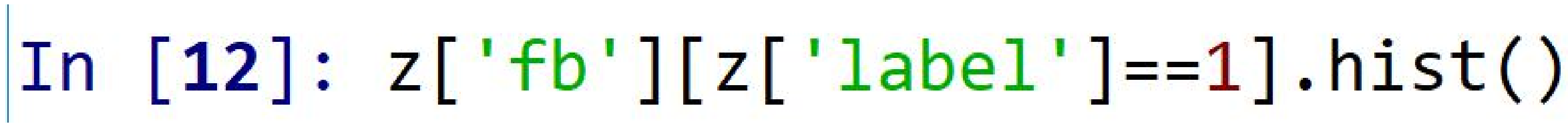


```
mean: label
0    215.819949
1    143.669548
Name: fb, dtype: float64
=====================
std: label
0    32.989451
1    30.871674
Name: fb, dtype: float64
```

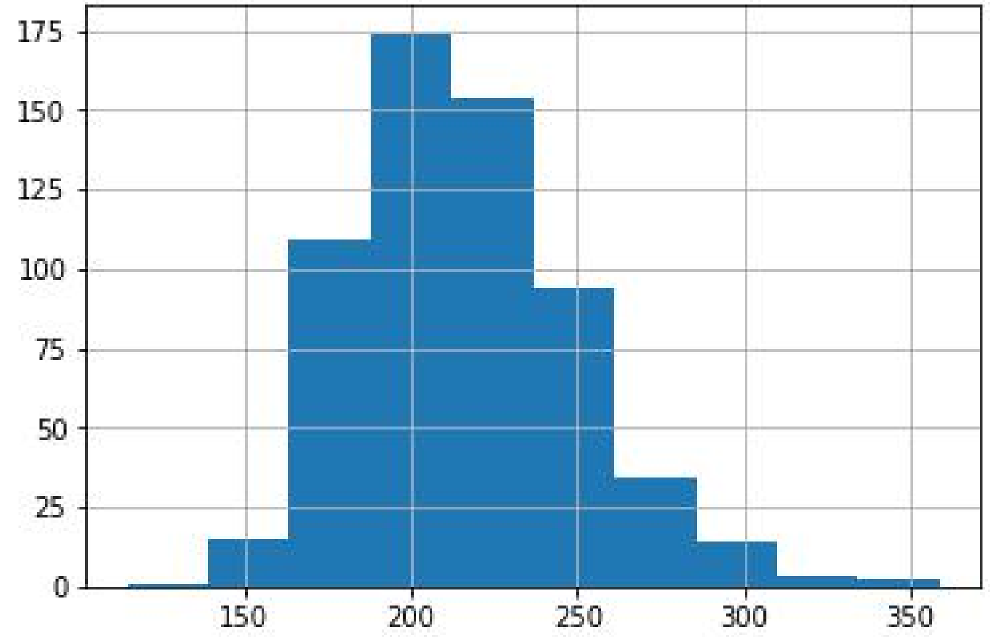
数据探索：计算所有人的平均基频

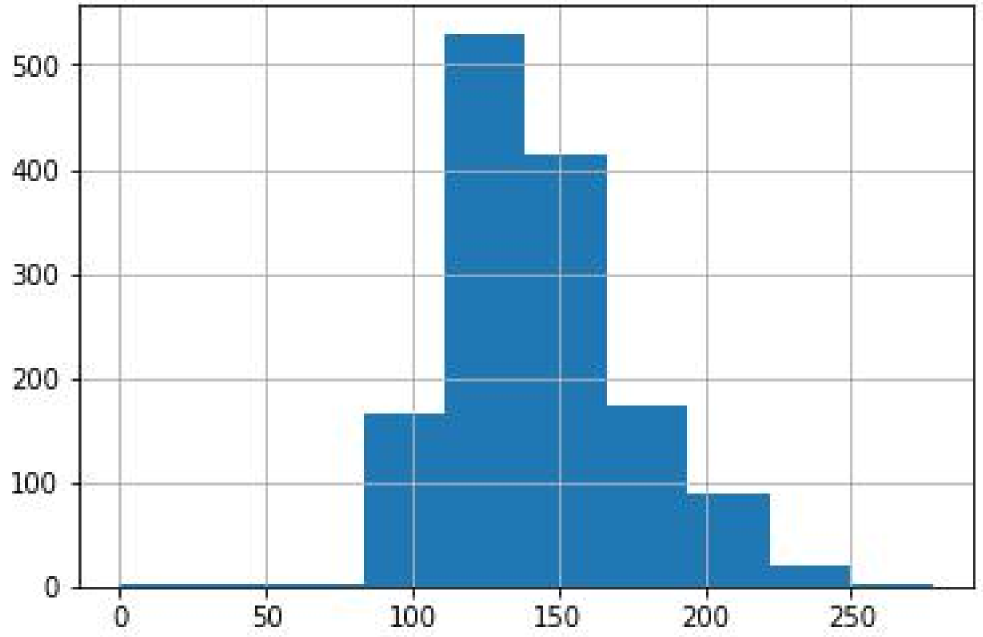


```
In [11]: z['fb'][z['label']==0].hist()
```

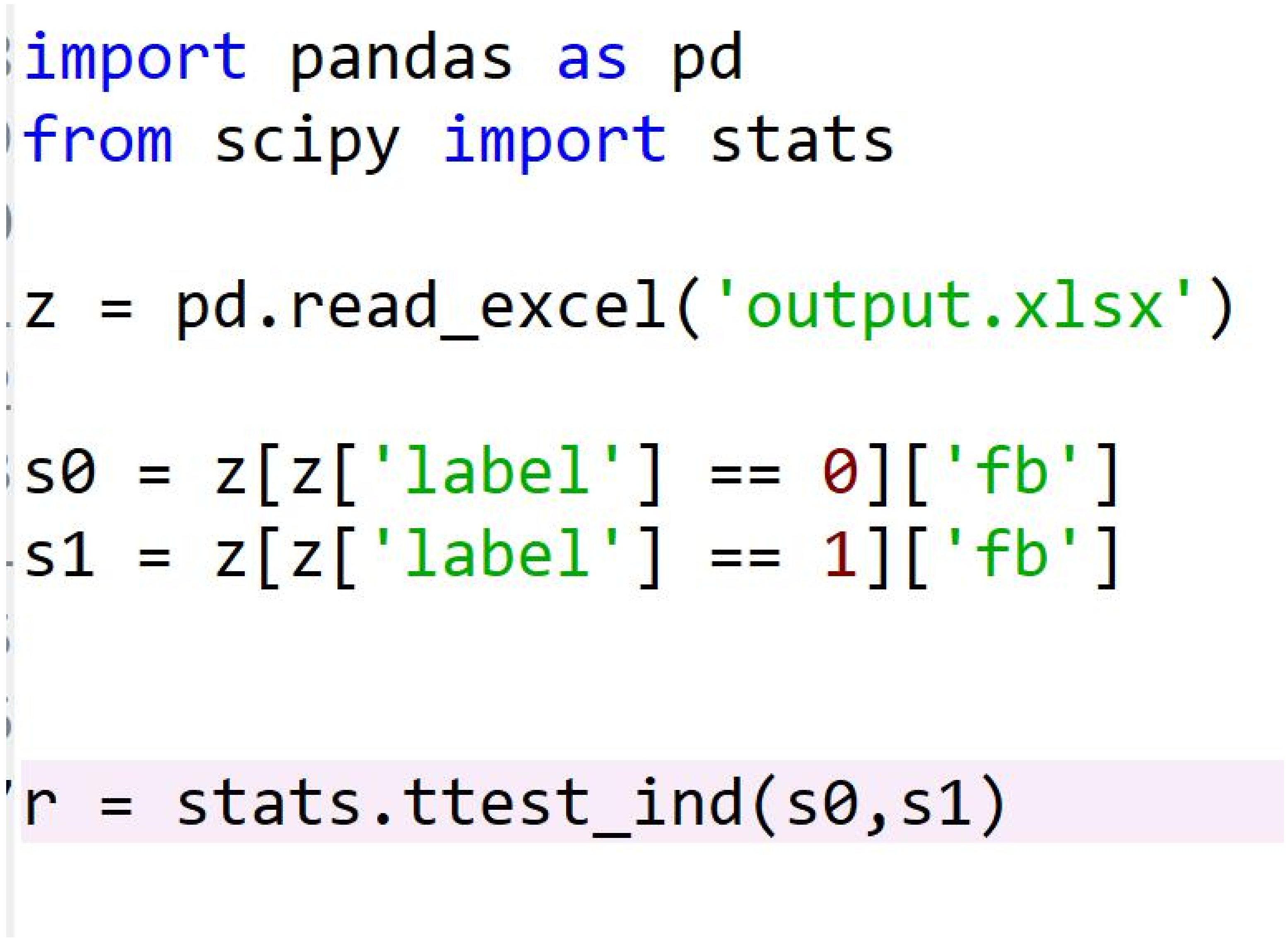


```
In [12]: z['fb'][z['label']==1].hist()
```

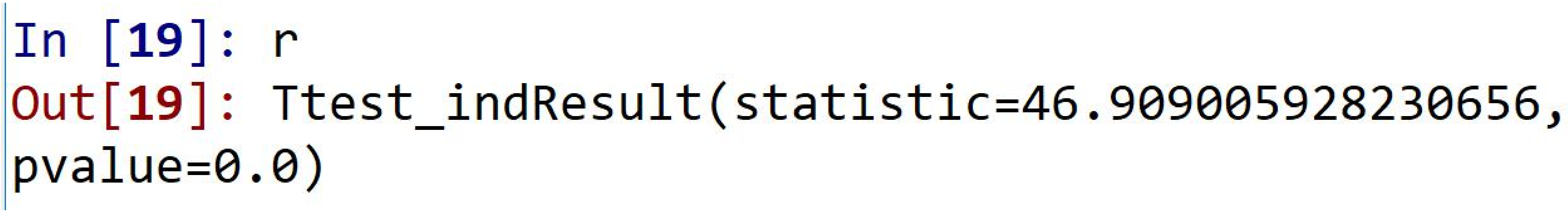




数据探索：检验男女在平均基频上的差异性

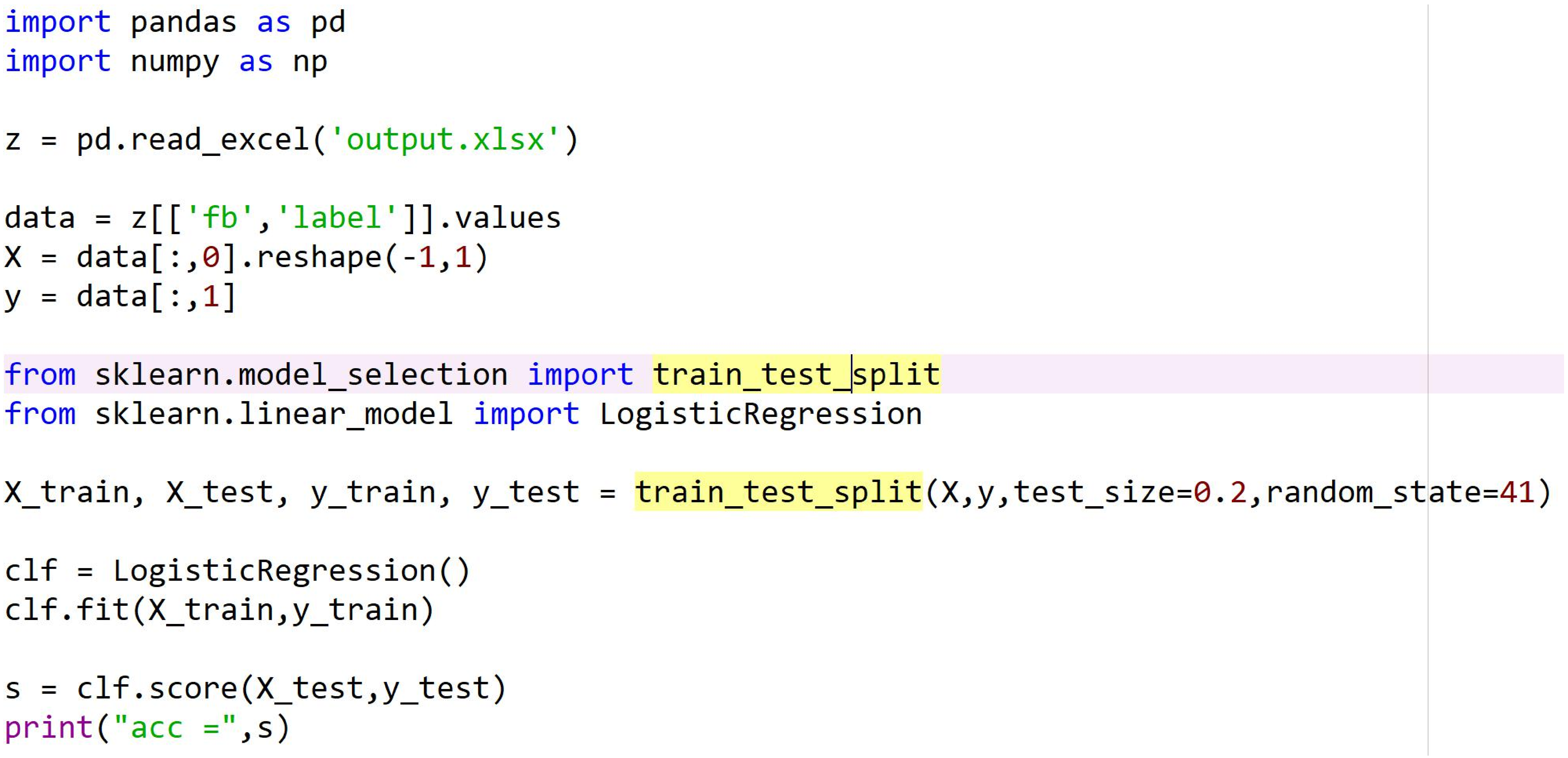


```python
import pandas as pd
from scipy import stats

z = pd.read_excel('output.xlsx')

s0 = z[z['label'] == 0]['fb']
s1 = z[z['label'] == 1]['fb']


r = stats.ttest_ind(s0,s1)
```

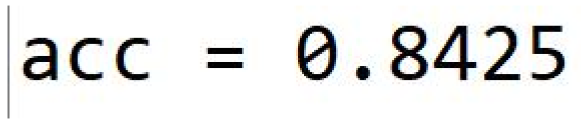


```
In [19]: r
Out[19]: Ttest_indResult(statistic=46.909005928230656,
pvalue=0.0)
```

baseline



```
import pandas as pd
import numpy as np

z = pd.read_excel('output.xlsx')

data = z[['fb','label']].values
X = data[:,0].reshape(-1,1)
y = data[:,1]

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=41)

clf = LogisticRegression()
clf.fit(X_train,y_train)

s = clf.score(X_test,y_test)
print("acc =",s)
```



```
acc = 0.8425
```

进一步

- 更丰富的数据探索
- 更多的特征（傅里叶变换、语谱图、梅尔频谱……）
- 更复杂的模型
- 更全面的评价指标（precision，recall，f1，……）
- ……

华院计算技术（上海）股份有限公司

上海 · 北京 · 成都 · 西安 · 硅谷

地址 · 上海市静安区万荣路1268号云立方大厦A座9楼

电话 · 021-63617288 传真 · 021-63617299

网址 · www.UniDT.com