

# 在開始之前

Github(簡報與資源)

<https://reurl.cc/GrN0jW>



Slido 有問題可以問

Code: 1017voice





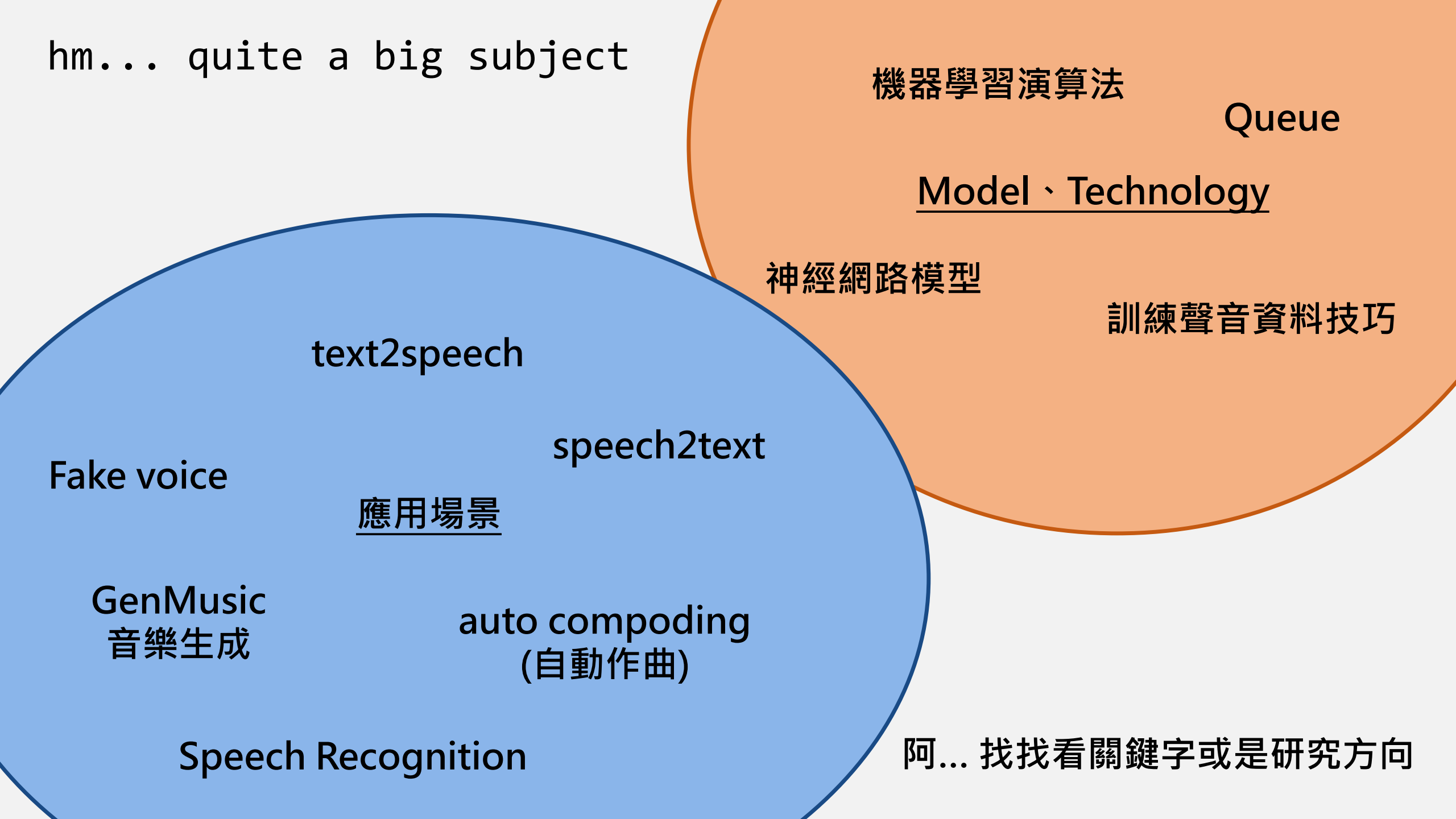
# {Voice} Analysis in R

---

操作、萃取、視覺化聲音數據

Presented by: 余佑駿(you-jun, yu)

hm... quite a big subject



機器學習演算法

Queue

Model、Technology

神經網路模型

訓練聲音資料技巧

text2speech

speech2text

Fake voice

應用場景

GenMusic  
音樂生成

auto componding  
(自動作曲)

Speech Recognition

阿... 找找看關鍵字或是研究方向

# {Voice} Analysis in R

`{Vocal}` Analysis in R

# {Audio} Analysis in R

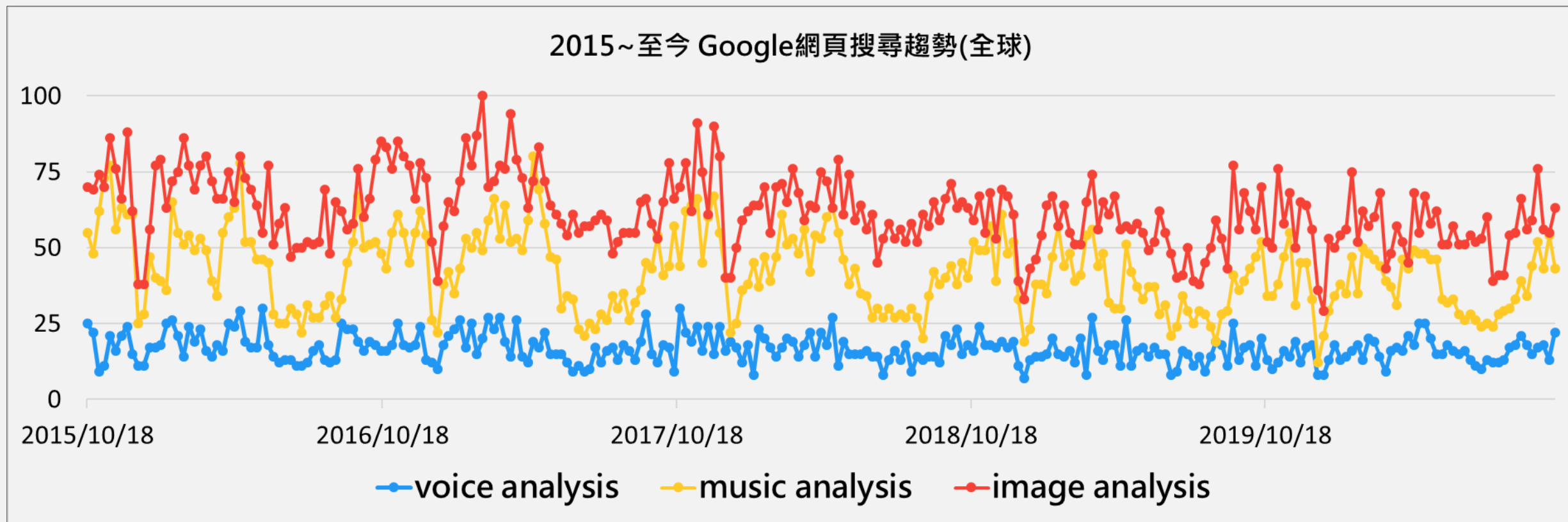
`{Acoustic}` Analysis in R

`{Vibration}` Analysis in R

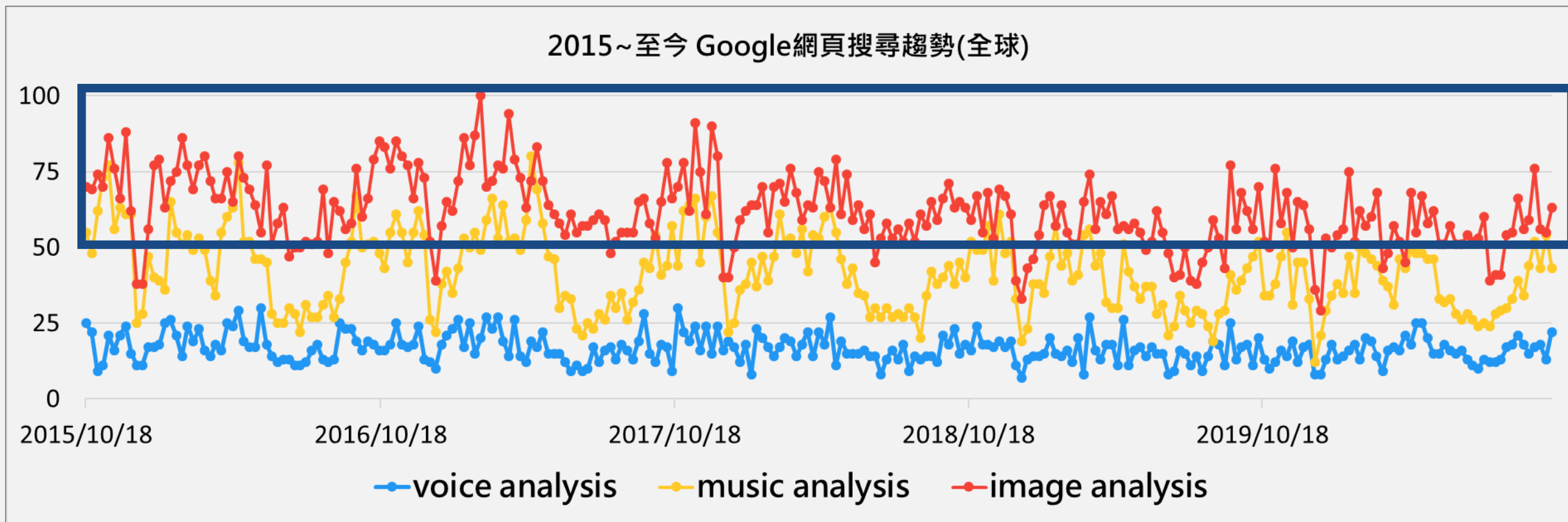


`{Music}` Analysis in R

# {Voice|Vocal|Audio|Acoustic|Vibration|Music} Analysis in R



# {Voice|Vocal|Audio|Acoustic|Vibration|Music} Analysis in R



人工智慧的發展，似乎沒有為聲音分析帶來分析研究的風潮（或局限於音樂？）。

資料不易取得？忽略聲音這個研究題材？聲音玩不出花樣？

# AGENDA

## 1. Introduce

- 聲音基本特性
- 時間域、頻率域
- 數位訊號與音訊檔案

## 2. manipulate

- 讀取、解讀、寫入
- 製造聲音

## 3. Extract Features

- 基本特性
- 頻率分析 by 傅立葉轉換

## 4. Visualization(?)

- 時域圖
- 頻譜圖
- 熱圖(時域+頻域)

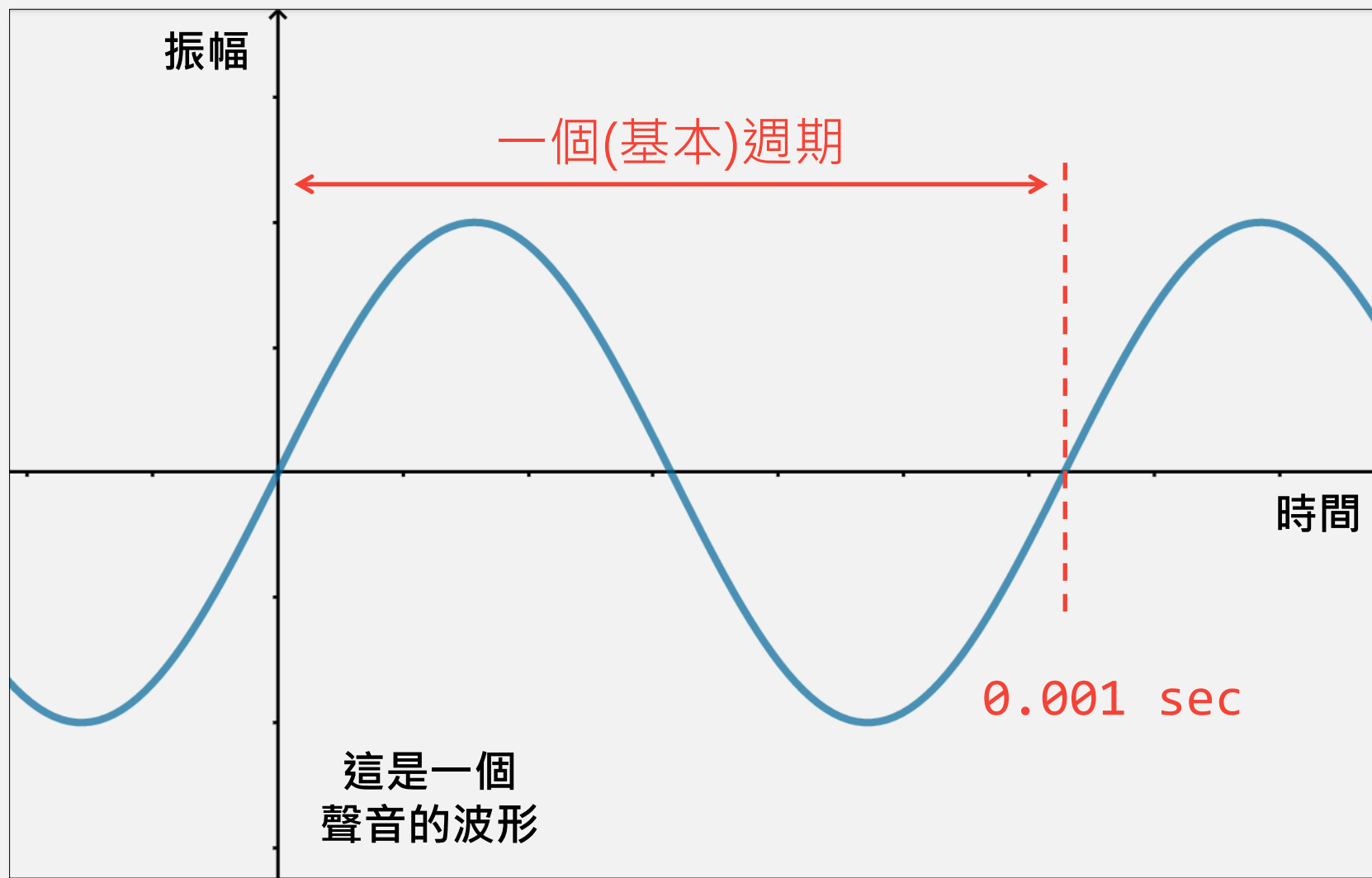
## 5. Some extra

- 諧波

## 6. Conclusion

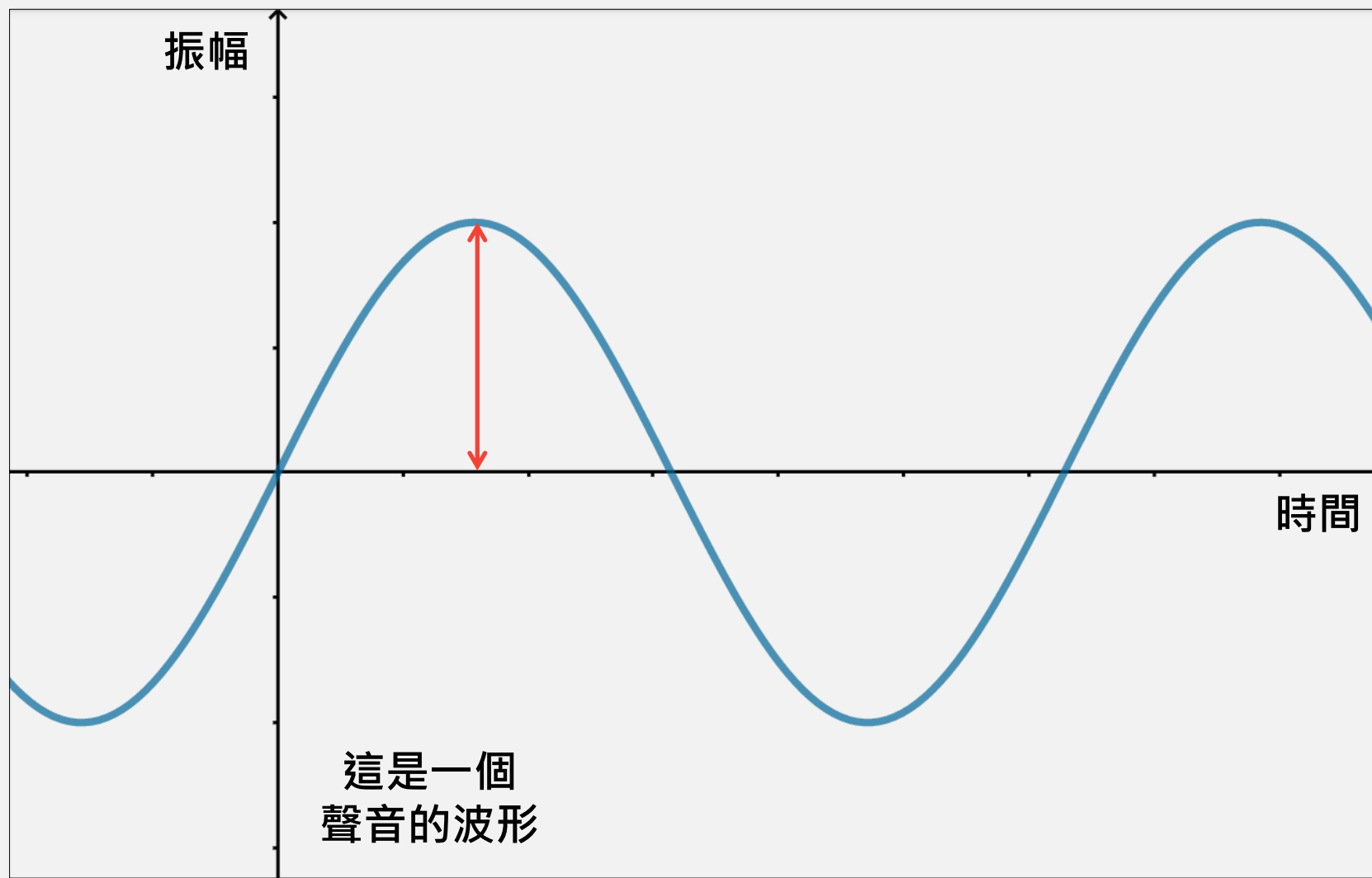
- 期許發展

# 1. Introduce: 聲音基本特性



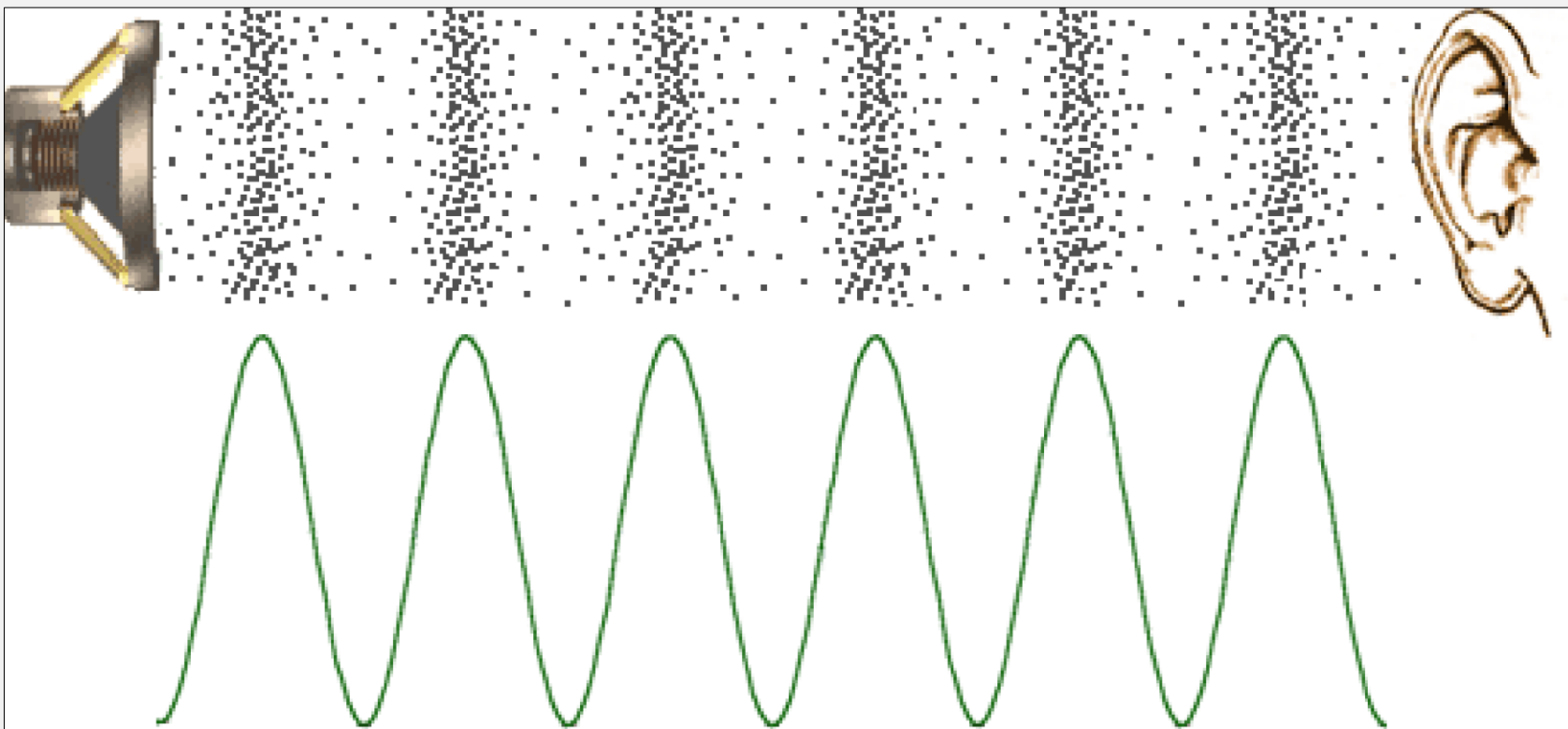
- 週期(period):  
震動一次所需的時間  
0.001 sec / 次
- 頻率(frequency):  
每秒震動的次數  
1000Hz = 1kHz

# 1. Introduce: 聲音基本特性



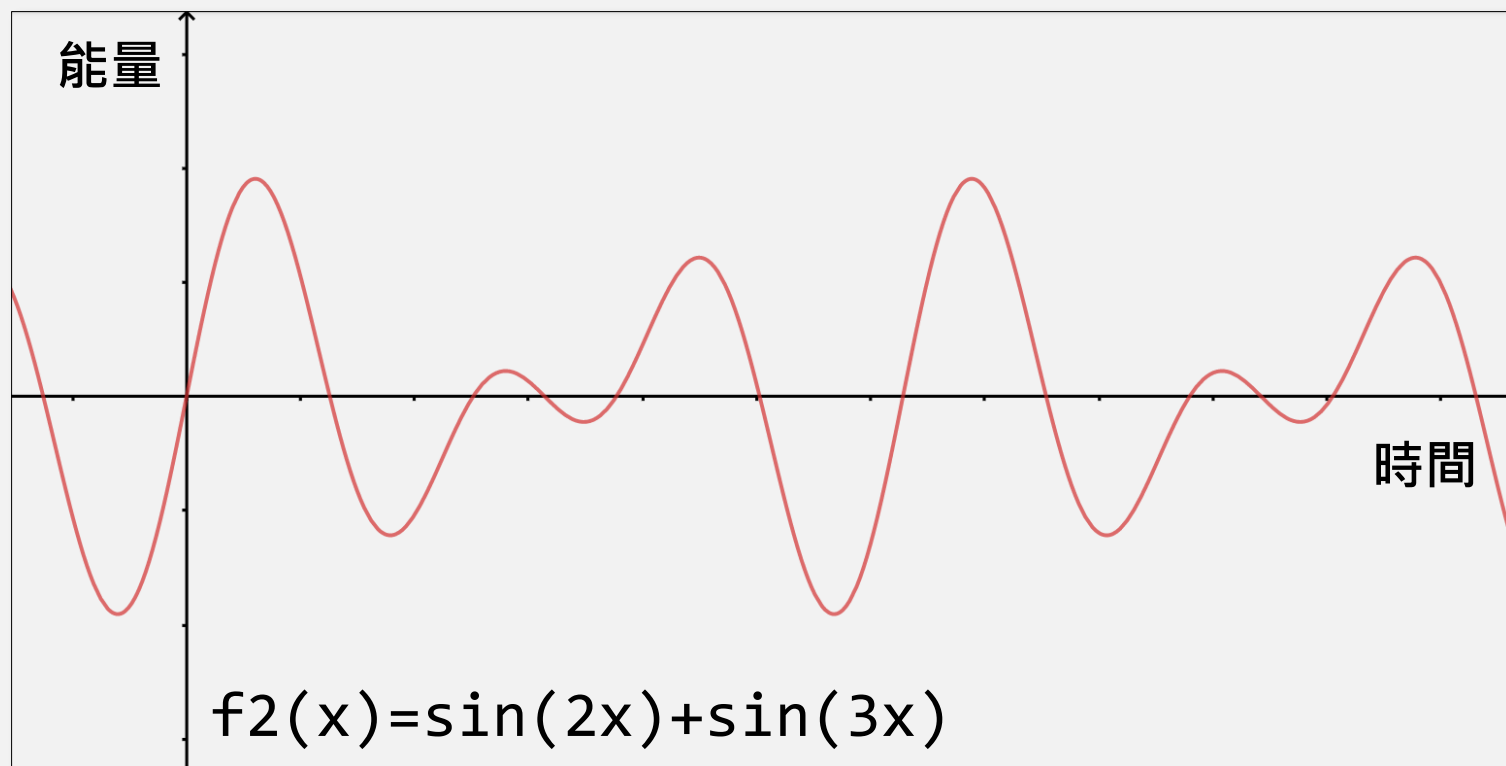
- 週期(period):  
震動一次所需的時間  
0.001 sec / 次
- 頻率(frequency):  
每秒震動的次數  
1000Hz = 1kHz
- 振幅(amplitude):  
聲音的能量大小。可以用音壓轉換得到 (SP to SPL)。

# 1. Introduce: 聲音基本特性

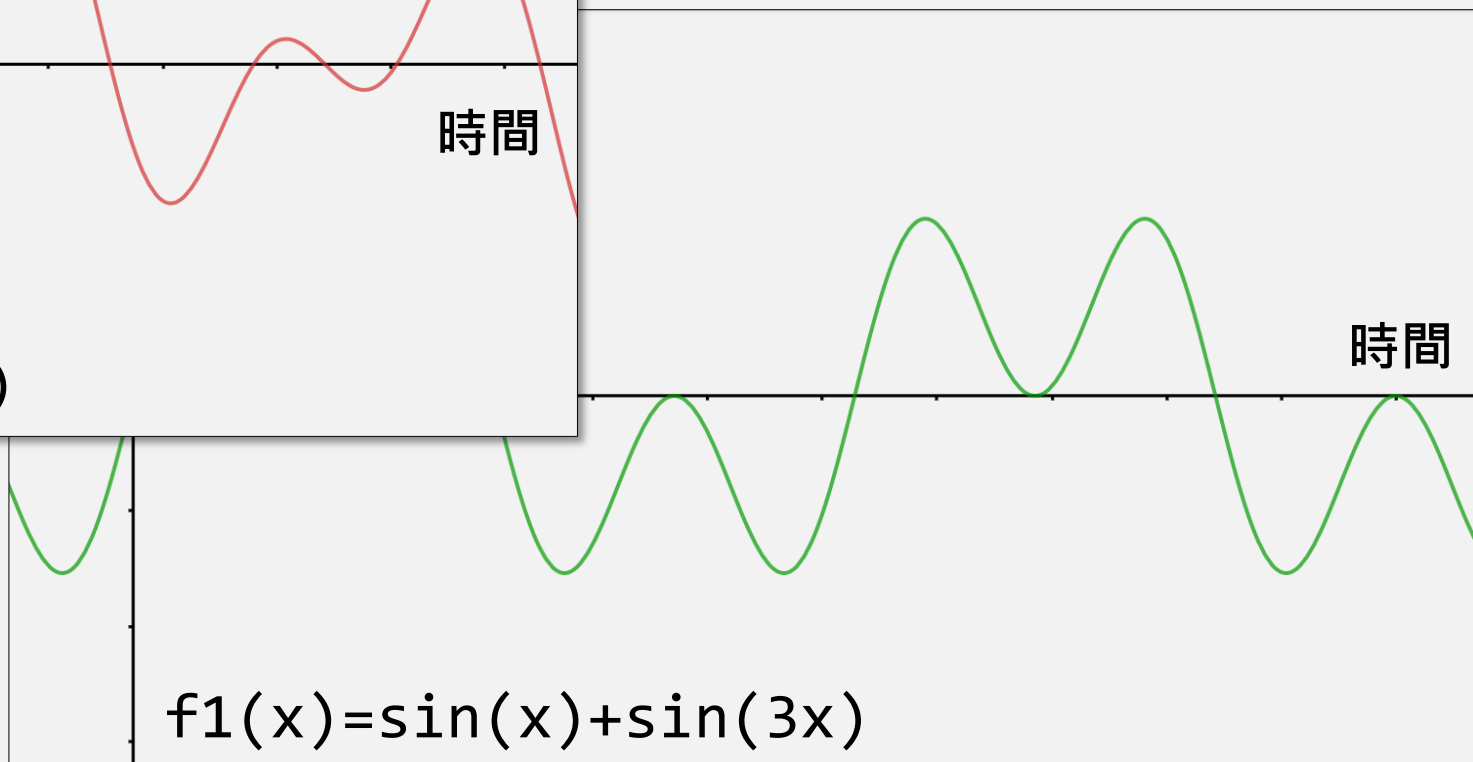


- 週期(period):  
震動一次所需的時間  
 $0.001 \text{ sec} / \text{次}$
- 頻率(frequency):  
每秒震動的次數  
 $1000\text{Hz} = 1\text{kHz}$
- 振幅(amplitude):  
聲音的能量大小。可以用音壓轉換得到  
(SP to SPL)。

# 1. Introduce: 聲音基本特性

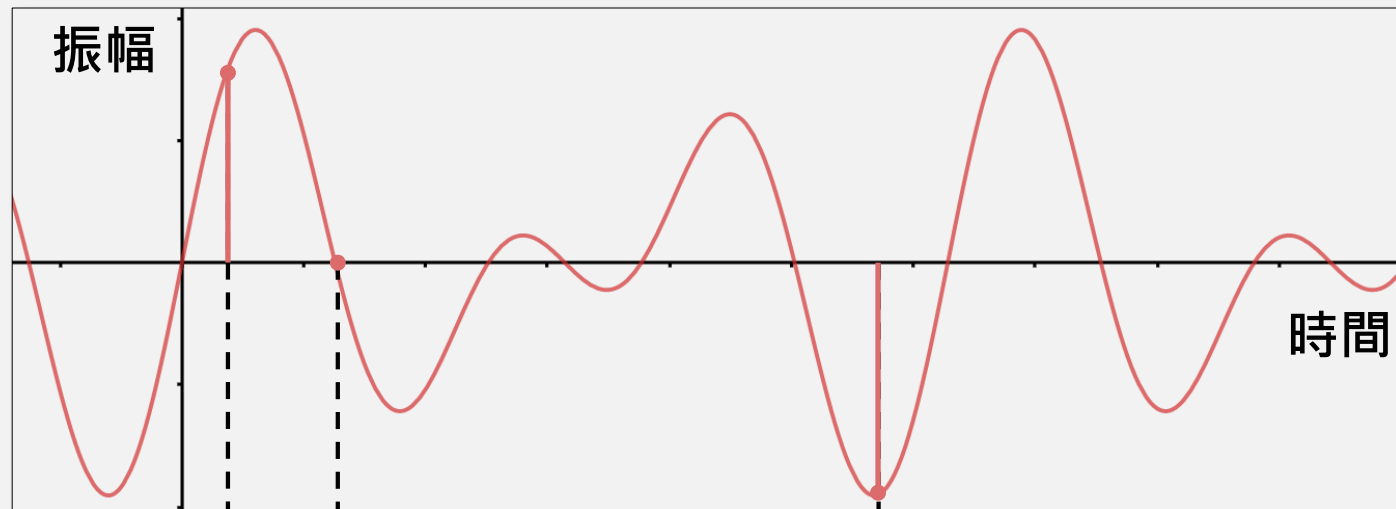


- 音色 (timbre):  
不同的頻率的組成，形成  
不同的波形，形成不同的  
音色。

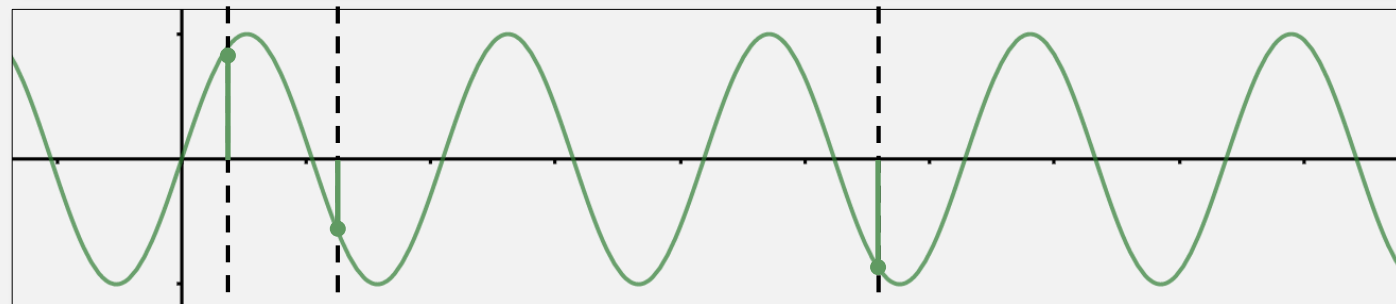




# 1. Introduce: 時間域、頻率域

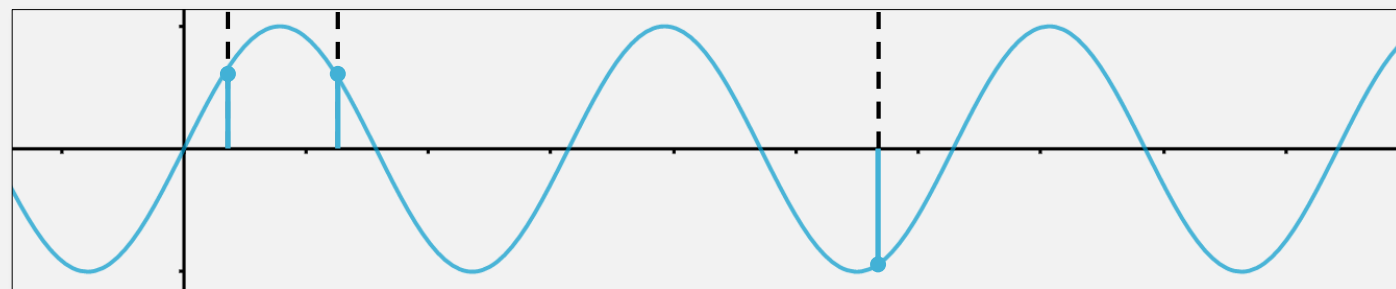


$$f(x) = \sin(2x) + \sin(3x)$$



$$h2(x) = \sin(3x)$$

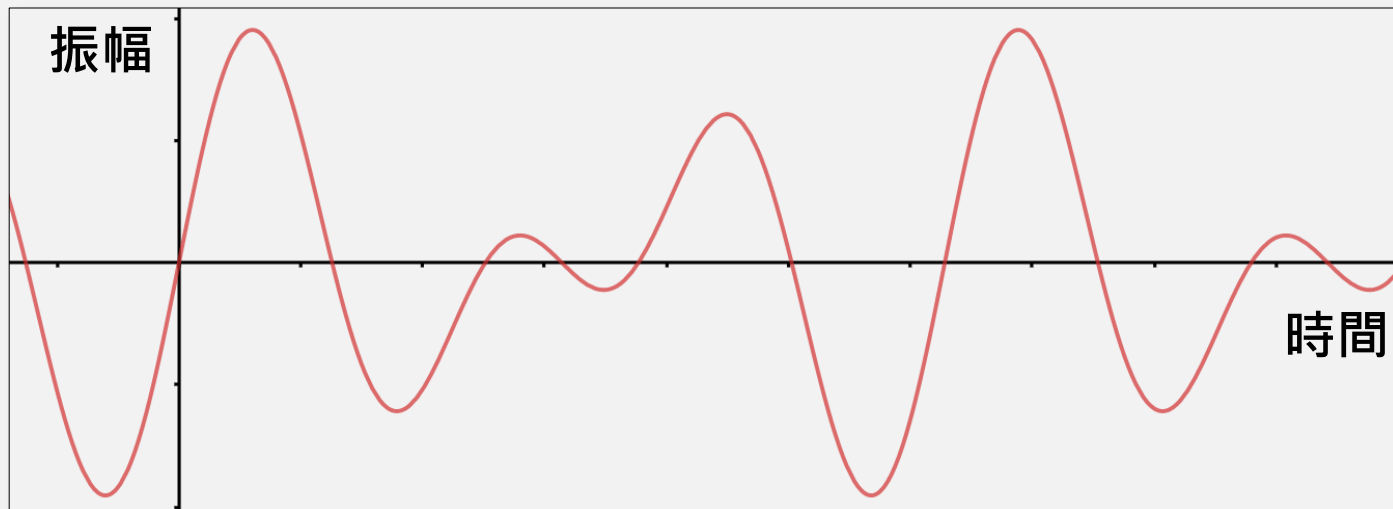
振幅為1



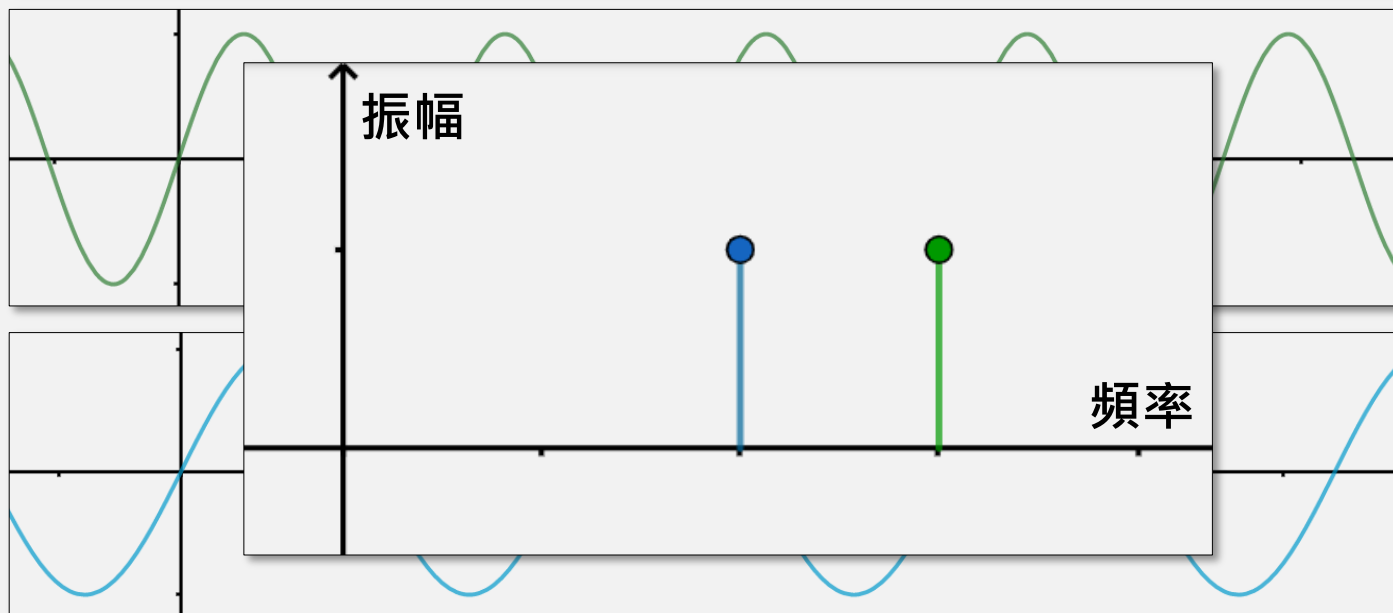
$$h1(x) = \sin(2x)$$

振幅為1

# 1. Introduce: 時間域、頻率域



$$f(x) = \sin(2x) + \sin(3x)$$



$$h_2(x) = \sin(3x)$$

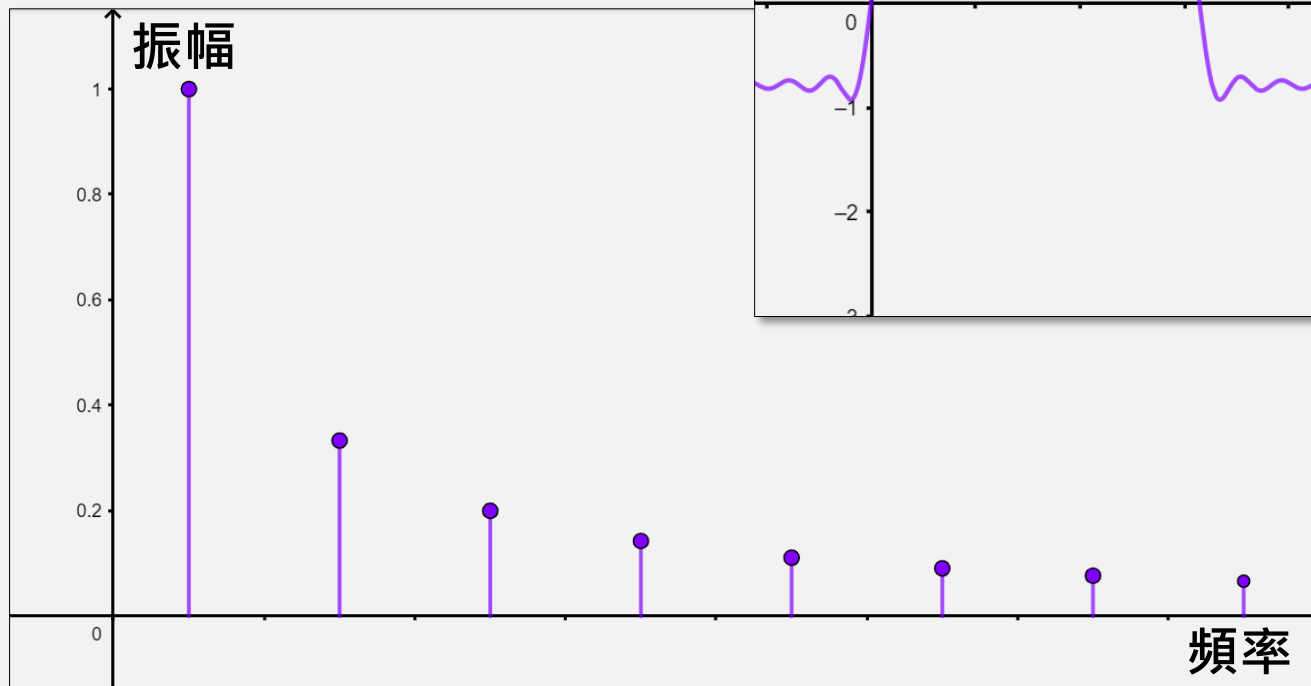
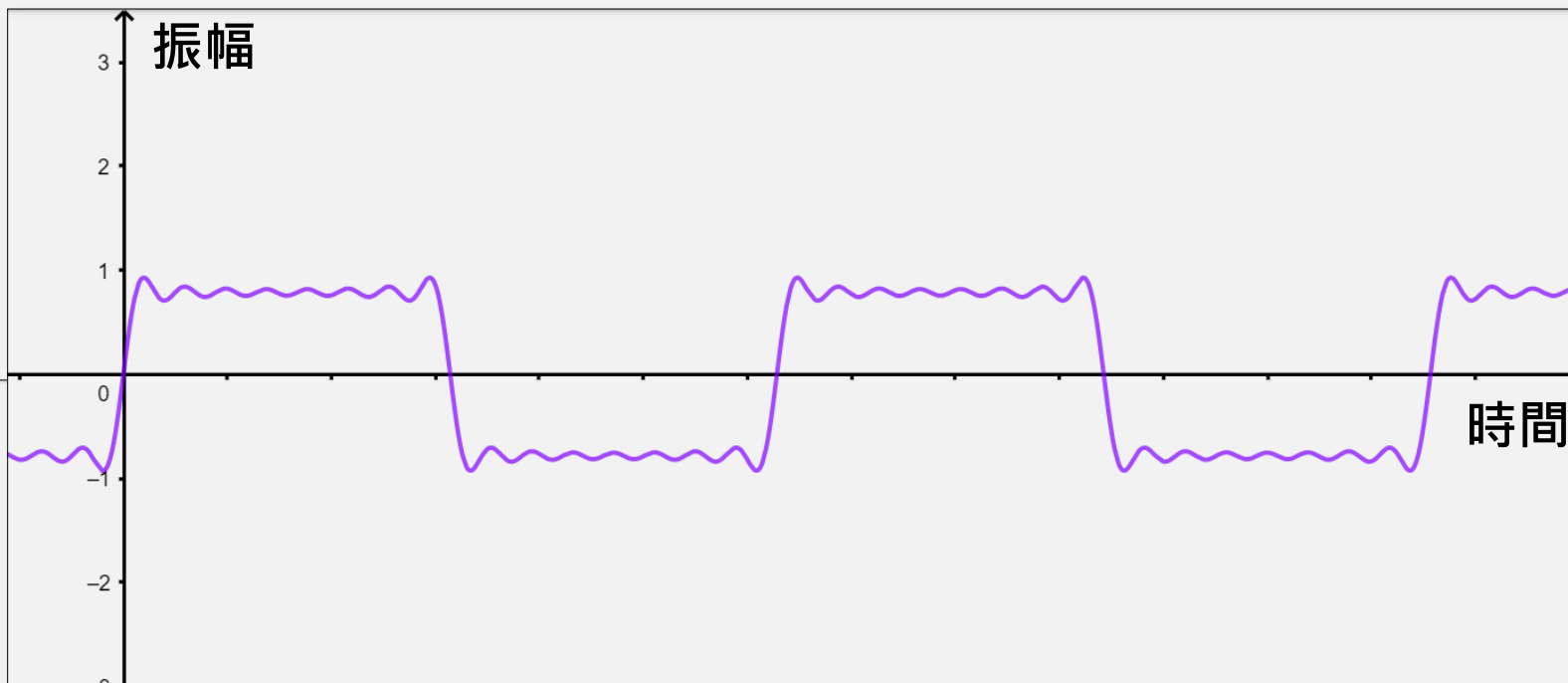
振幅為1

$$h_1(x) = \sin(2x)$$

振幅為1

# 1. Introduce: 時間域、頻率域

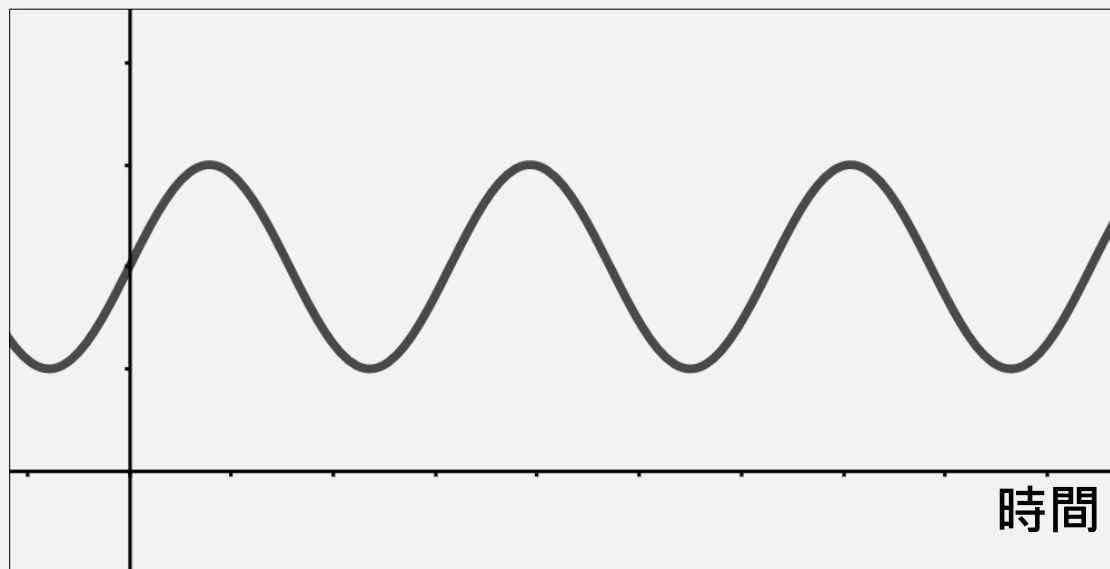
傅立葉轉換



逆傅立葉轉換

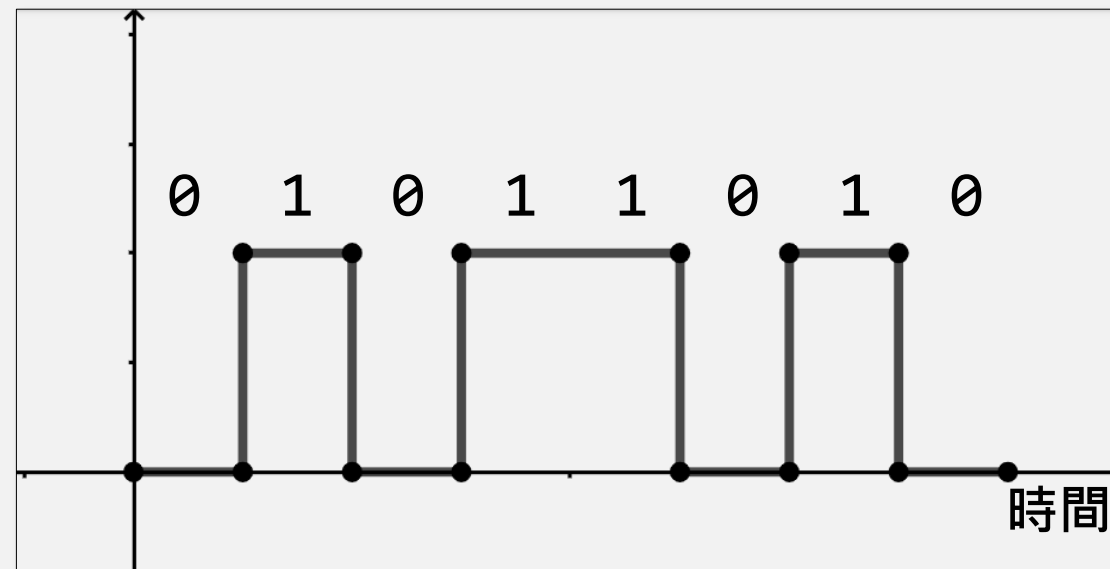
# 1. Introduce: 數位訊號與音訊檔案

類比(Analog signal)



「連續的訊號」稱為類比訊號。  
大自然裡一切的訊號，包括我們聽到的聲音、看到的影像，都屬於類比訊號

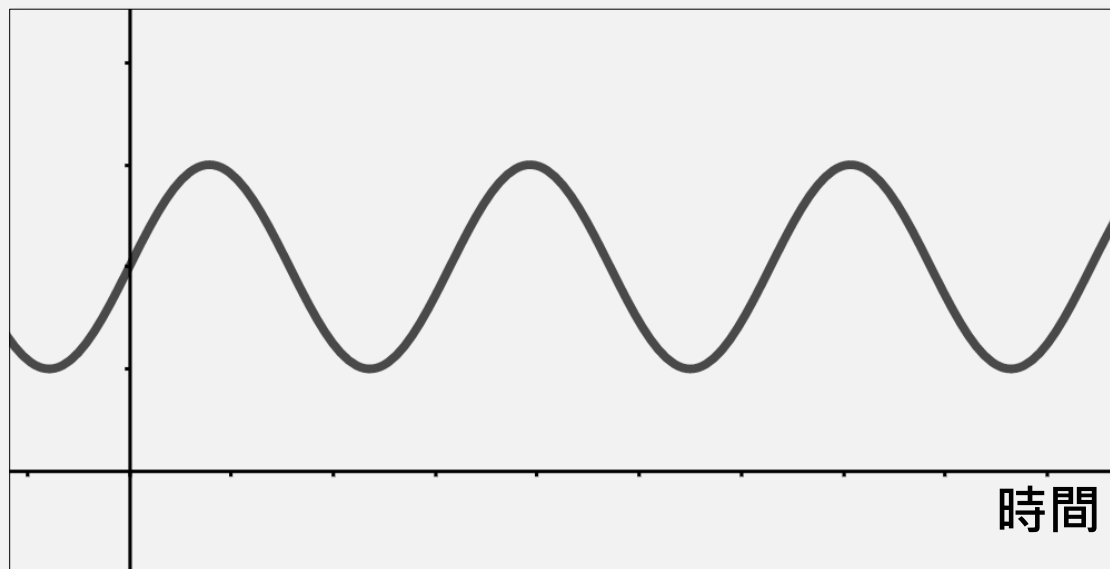
數位(Digital signal)



「不連續的訊號」稱為數位訊號。  
將連續的類比訊號變成 0 與 1 兩種不連續的訊號，如：電腦運算時，只有低電壓(0V)與高電壓(1V)

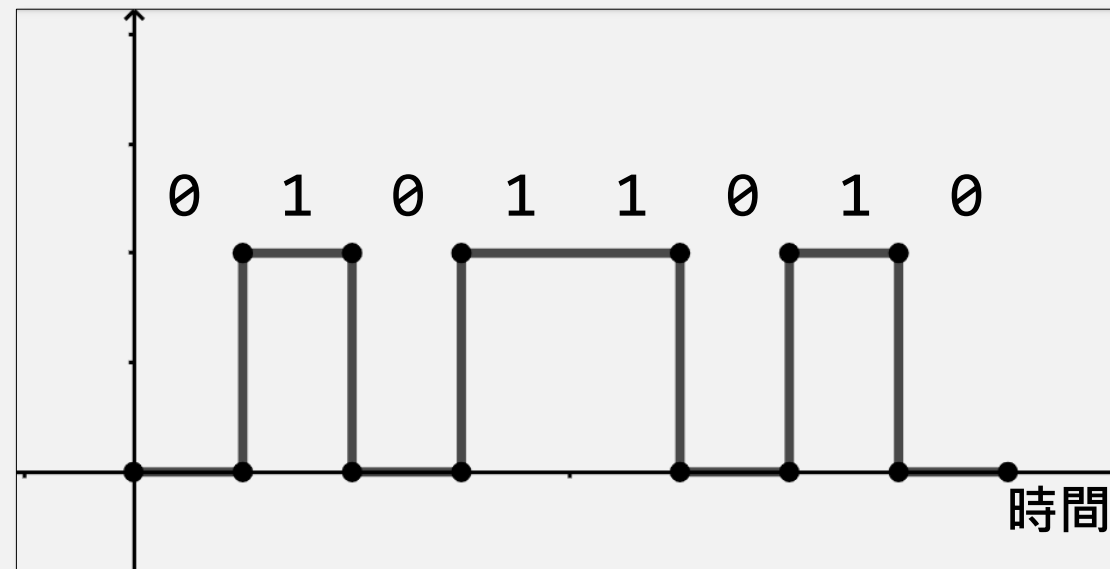
# 1. Introduce: 數位訊號與音訊檔案

類比(Analog signal)



「連續的訊號」稱為類比訊號。  
大自然裡一切的訊號，包括我們聽到的聲音、看到的影像，都屬於類比訊號

數位(Digital signal)



「不連續的訊號」稱為數位訊號。  
將連續的類比訊號變成 0 與 1 兩種不連續的訊號，如：電腦運算時，只有低電壓(0V)與高電壓(1V)

# 1. Introduce: 數位訊號與音訊檔案

## 有無壓縮((un)compressed)

---

常見的無壓縮檔案格式為\*.wav，而壓縮檔案格式為\*.mp3。是對位元速率做壓縮。

## 取樣頻率(SampleRate)

---

取樣頻率越高，數位訊號的波形越接近類比訊號的波形，聲音的品質也越好。而在做取樣時，必須遵守{奈奎斯特頻率}。

## 解析度(BitsPerSample)

---

存取訊號資料點所用的位元數，位元數越大，每個資料點的數值就越精確，也會花比較大的硬體儲存空間。

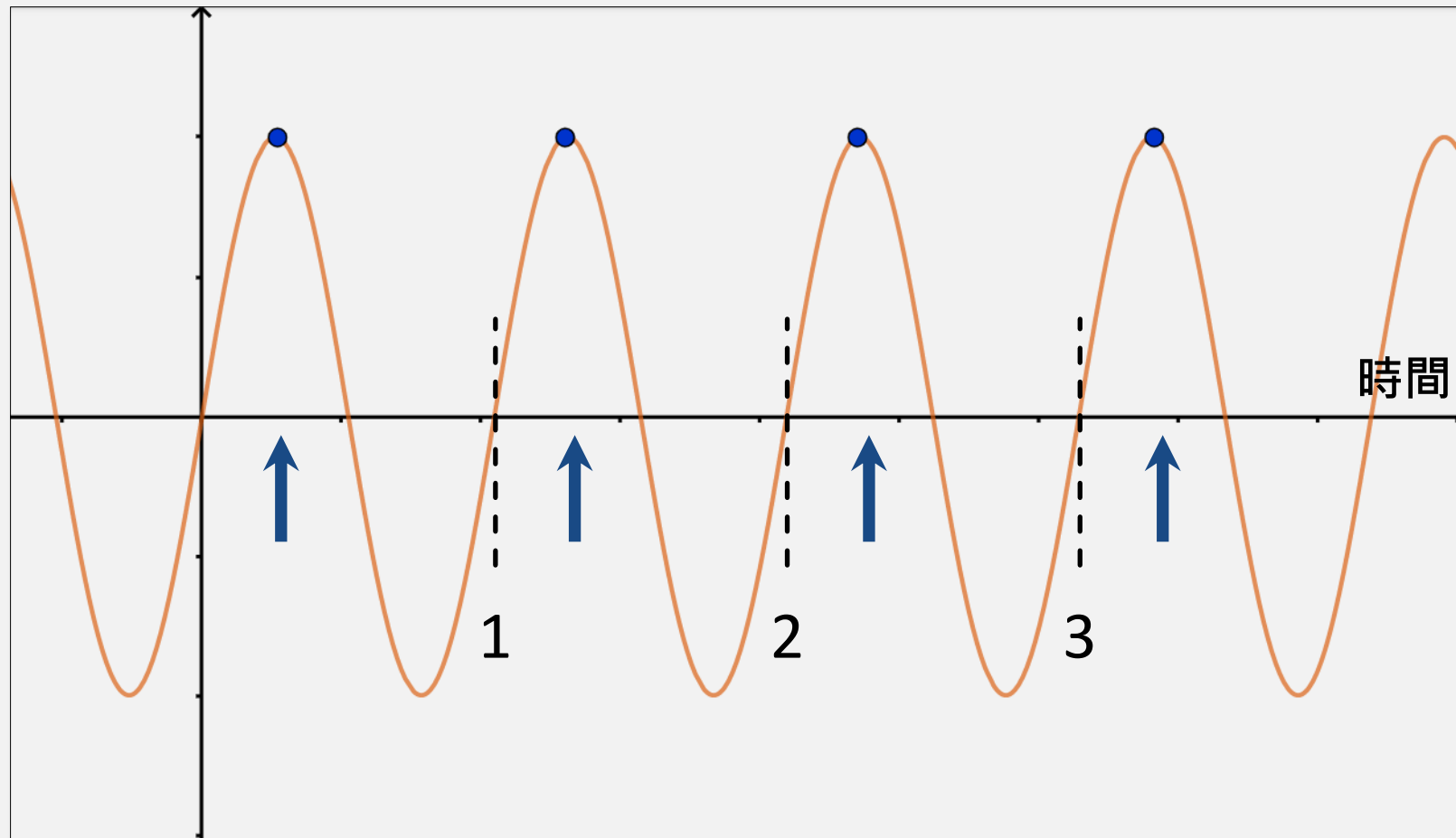
## 聲道(NumChannels)

---

聲音的來源數，如單聲道(左右聲音相通)、雙聲道(左右聲音有差異，聽起來產生立體感)。

# 1. Introduce: 數位訊號與音訊檔案

採樣頻率(SampleRate)



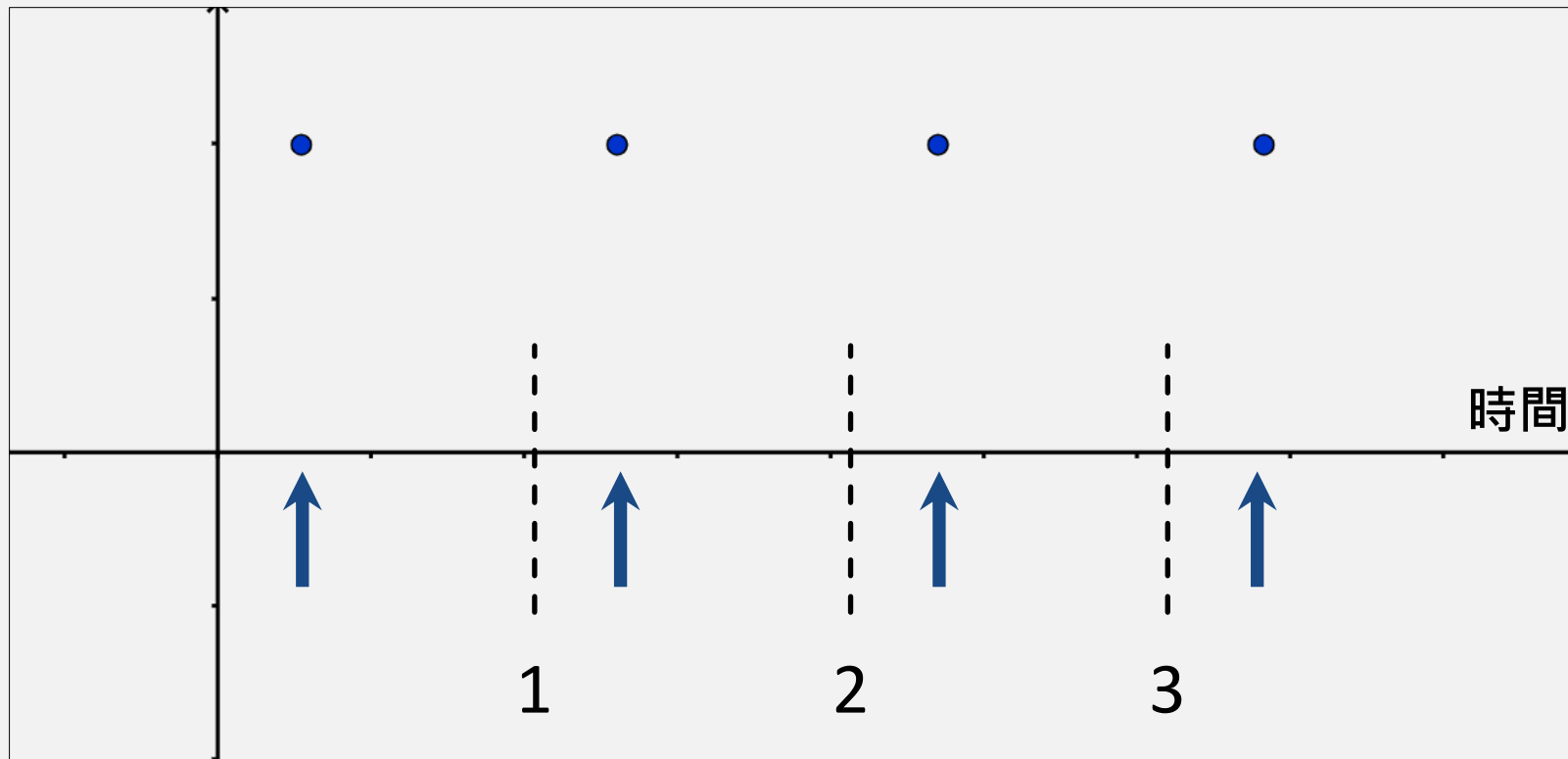
- 聲音頻率:1Hz
- 採樣頻率:1Hz

這樣採樣出的數據是無效的!!  
因為無法得知原本的波形

# 1. Introduce: 數位訊號與音訊檔案

採樣頻率(SampleRate)

奈奎斯特頻率



- 聲音頻率:1Hz
- 採樣頻率:1Hz

這樣採樣出的數據是無效的!!  
因為無法得知原本的波形



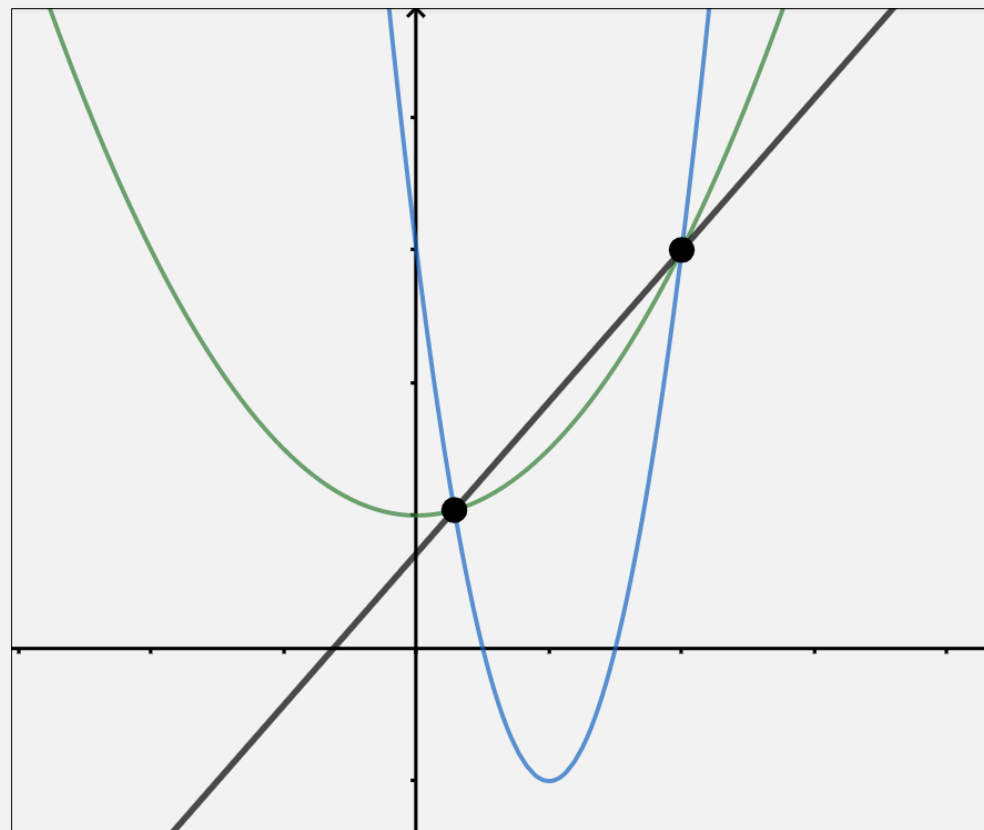
# 1. Introduce: 數位訊號與音訊檔案

有點類似

- 給定兩個點，可以決定一直線 = 給定 $(1 + 1)$ 個點，可以決定一次函數。  
但會有無數個二次函式通過
- 給定 $(2+1)$ 個點，可以決定二次函式。

## {拉格朗日差值多項式}

對於給定的若  $n+1$  個點，對應於它們的次數不超過  $n$  的拉格朗日多項式  $L$  只有一個。如果引入次數更高的多項式，則有無窮個。



# 1. Introduce: 數位訊號與音訊檔案

在聲音分析的規則是，

假設CD音訊訊號的採樣頻率為10000 Hz，那麼它的{奈奎斯特頻率}就是5000 Hz，這是CD音訊數據所能表現的最高頻率。



{奈奎斯特頻率 Nyquist frequency}

頻率在取樣時至少要大於原訊號頻率的2倍，才可以得到有意義的訊號，也能還原成原本的訊號。

## 2. manipulate: 讀取、解讀、寫入

- 範例資料: [Audio Tone Files](#)、內建數據tico(小鳥叫聲)
- R 可以對聲音操作的套件(常見): tuneR、audio、seewave、soundgen  
基本操作 進階使用
- 詳細可以參考 Github repo 下的 ./code/code4demo.Rmd
- 新手入門我推薦 seewave 套件，整合了 tuneR 和 audio，而且文件寫得不錯，有在更新(2020-05)。

	Input	Output	Mono/Stereo	Play	Object
tuneR	readWave, readMP3	writeWave	mono, stereo	play	Wave
audio	load.wave, record	save.wave	mono, stereo	play, pause, resume, rewind	audioSample
seewave	-	export, savewav	-	listen	vector, matrix, ts, mts, Wave, audioSample

## 2. manipulate: 讀取、解讀、匯出

### 讀取

```
readWave(filename = "file.wav",  
          from = 0,  
          to = 2,  
          units = "seconds")
```

→ 可以指定讀取時間範圍。

→ "sample" (取樣點個數)、  
"seconds" (秒)、  
"hour" (小時)

## 2. manipulate: 讀取、解讀、匯出

### 解讀

```
Wave Object
  Number of Samples:      88200
  Duration (seconds):     2
  Samplingrate (Hertz):   44100
  Channels (Mono/Stereo): Mono
  PCM (integer format):   TRUE
  Bit (8/16/24/32/64):    16
```

→ 資料格式其實寫得很清楚!!

## 2. manipulate: 讀取、解讀、匯出

### 裁切長度

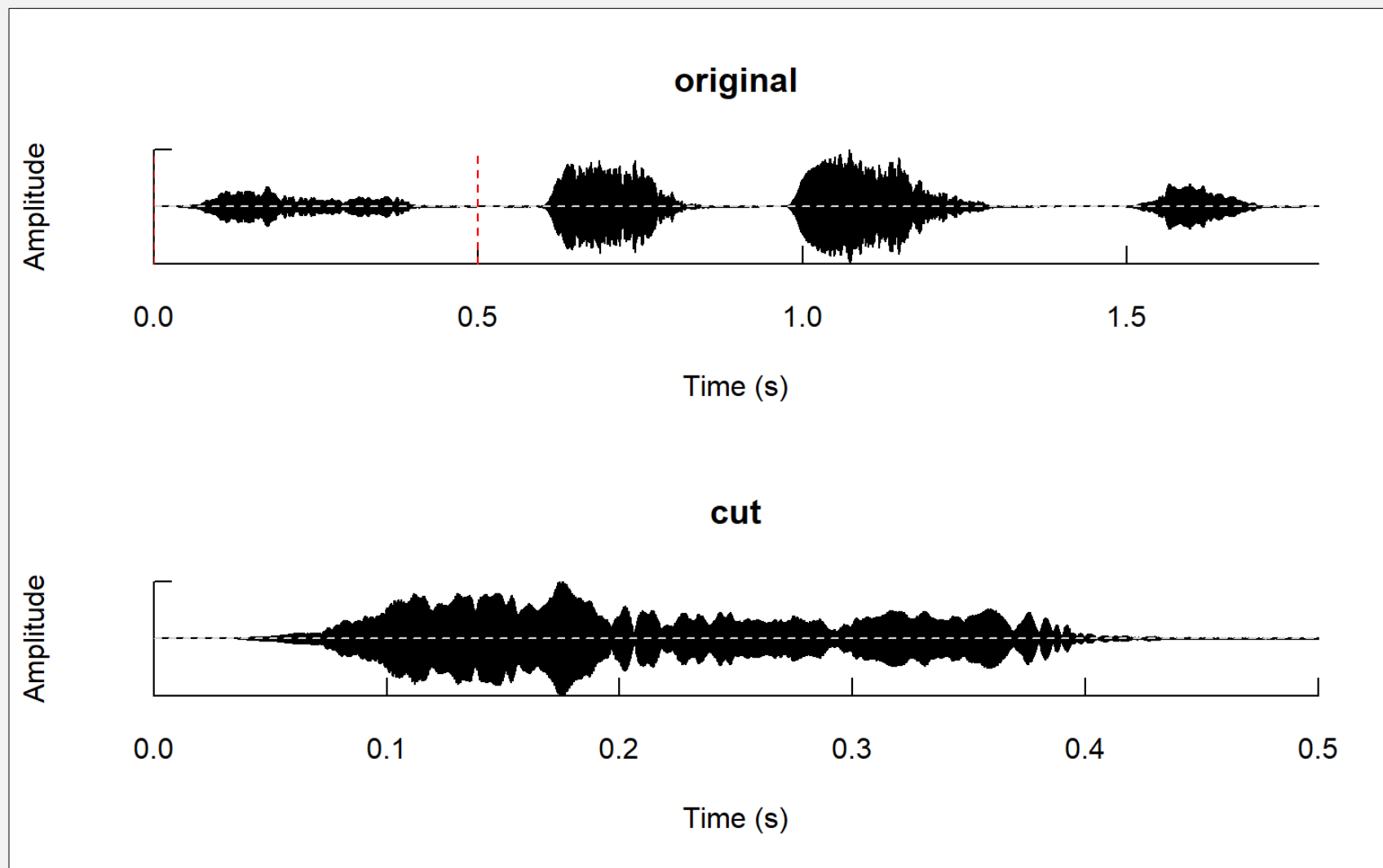
```
ticofirst <- cutw(wave = tico,  
                  f = 22050,  
                  from = 0,  
                  to = 0.5,  
                  plot = T,  
                  output = "Wave")
```

裁切完成後繪製圖形，可知道裁切的部分為全部音訊的哪個位置。

這些格式都可以。("matrix", "Wave", "Sample", "audioSample" or "ts")

## 2. manipulate: 讀取、解讀、匯出

裁切長度-結果



## 2. manipulate: 讀取、解讀、匯出

匯出聲波

```
savewav(wave = ticofirst,  
        filename = "tico_firstnote.wav")
```



## 2. manipulate: 製造聲音

生成聲音: 生成 sin 波

```
demo_wav <- sine(freq = 5,  
                  duration = 1,  
                  samp.rate = 44100,  
                  bit = 32,  
                  xunit = "time")
```

→ 頻率

→ 長度，隨 xunit 改變。

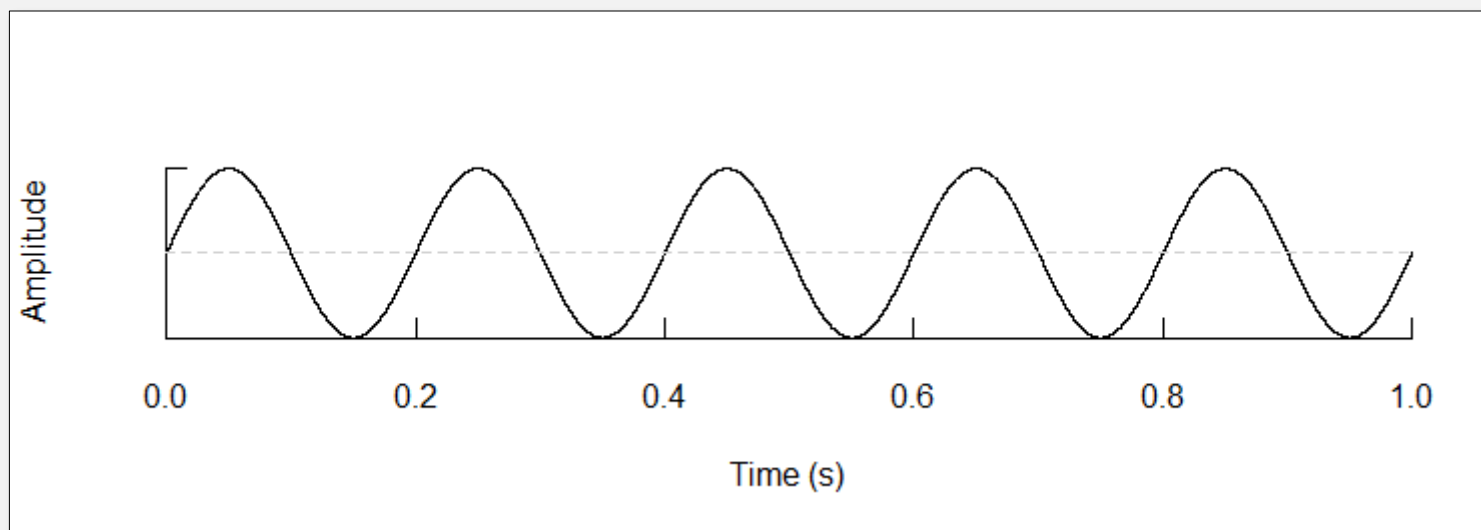
→ 取樣率

→ 解析度

## 2. manipulate: 製造聲音

繪製聲波(時域圖)

```
oscillo(demo_wav)
```



播放

```
listen(demo_wav)
```

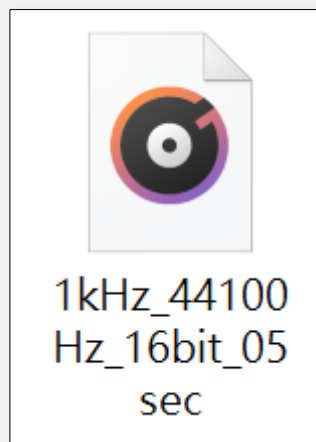
### 3. Extract Features: 基本特性

- 振幅: 利用 range

```
> tico@left %>% range()  
[1] -18596 19125
```



- 取得聲音頻率
  - 檔案有，直接取。
  - 檔案沒有，利用過零率推敲。



```
[1]      0    3219    6372    9396   12232   14817  
[7]  17103  19043  20597  21732  22428  22669  
[13]  22450  21778  20663  19131  17209  14939  
[19]  12367   9544   6528   3379    162  -3058  
[25]  -6217  -9249 -12095 -14694 -16997 -18954  
[31] -20528 -21686 -22404 -22667 -22472 -21822  
[37] -20729 -19216 -17314 -15060 -12502  -9691  
[43]  -6682  -3538   -323   2899   6061   9102  
[49] 11957 14572
```

### 3. Extract Features: 基本特性

```
Wave Object
  Number of Samples:      88200
  Duration (seconds):     2
  Samplingrate (Hertz):   44100
  Channels (Mono/Stereo): Mono
  PCM (integer format):   TRUE
  Bit (8/16/24/32/64):   16
```

一個基本周期約為

$$44 * (1/44100) = 0.0009977$$


$$45 * (1/44100) = 0.0010204$$

和真實的 0.001秒/次 差不多

```
> wav_head100 <- demo_wav@left %>% head(100)
```

```
> wav_head100 <- tmp01 <- ifelse(tmp>=0, 1, -1)
```

```
> which(diff(tmp01)!=0)
23 45 67 89
```



### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

what it means  
它的功用為何

分解聲音的頻率，得到{各頻率}，  
以及其{大小}。

why it's intuitive  
如何直觀理解

$$\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \underline{e^{-2\pi i f t}} dt$$

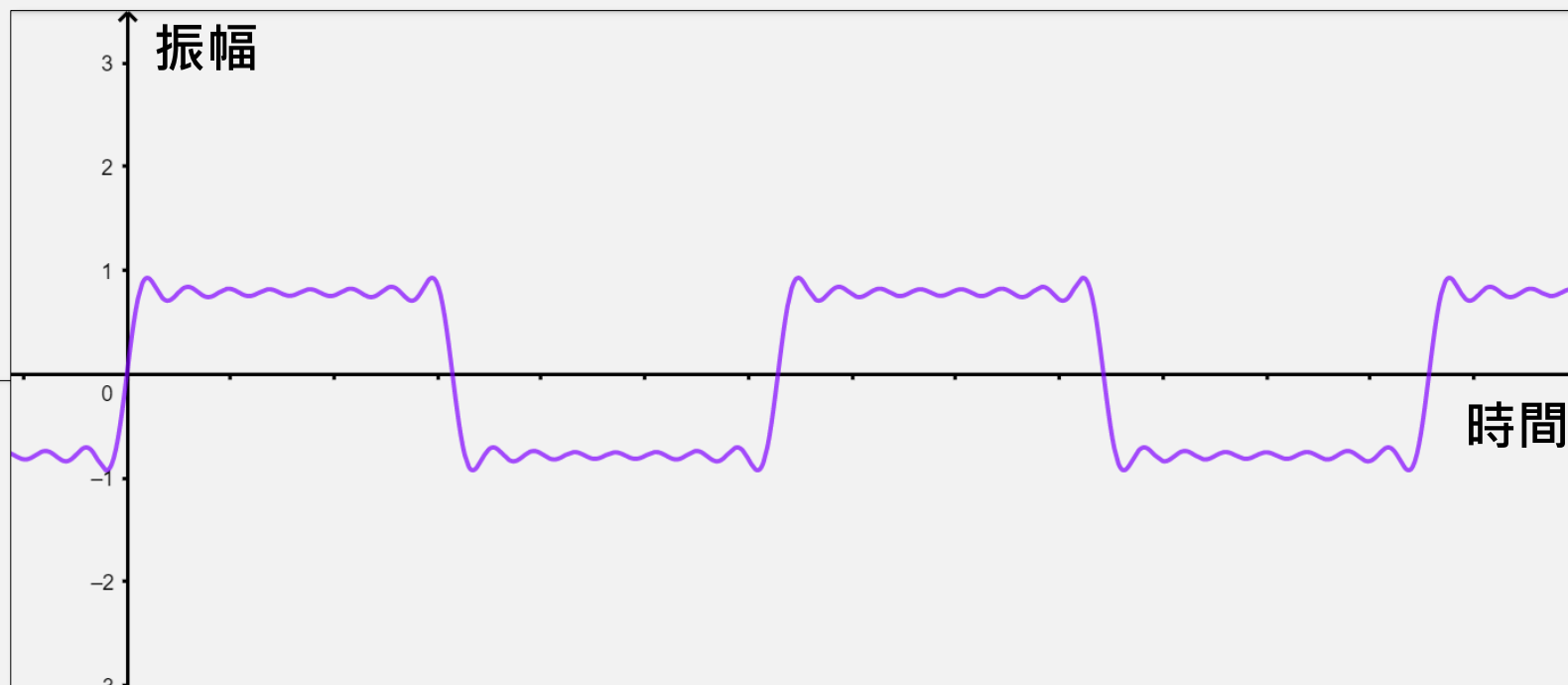
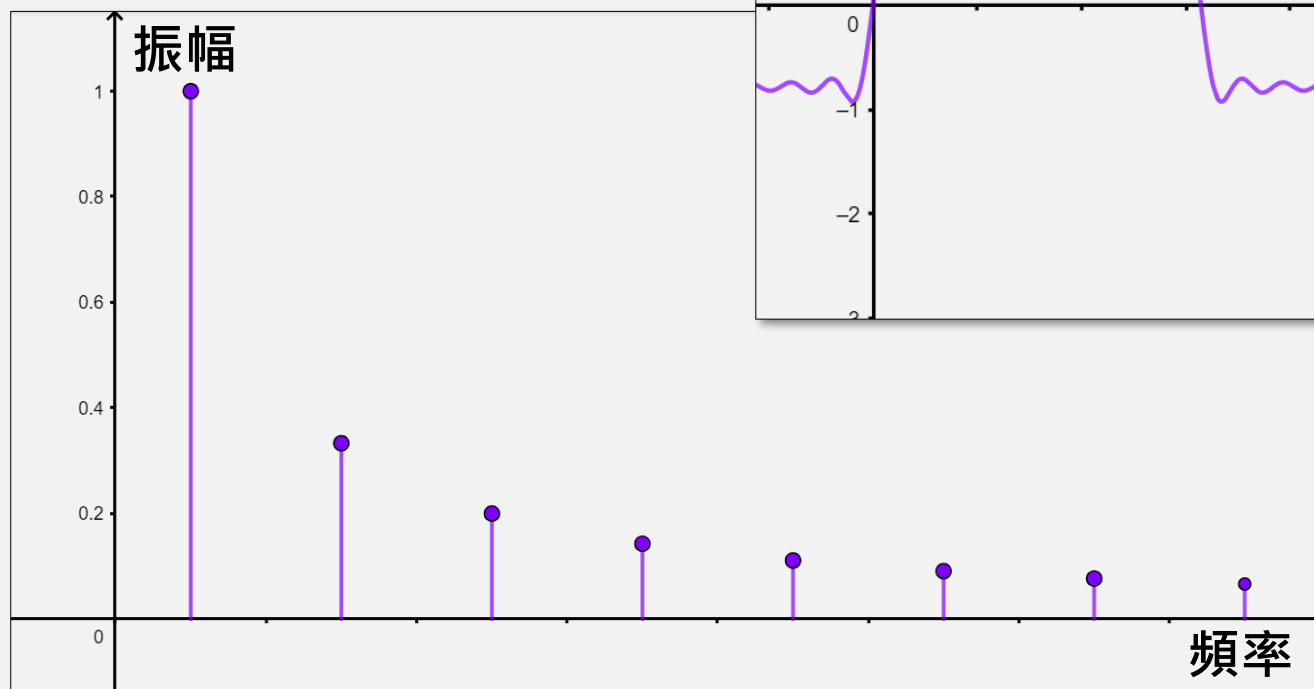
why it works  
為何可以這樣做

1. 正交函數

2.  $\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \underline{\cos(-2\pi f t) + i \sin(-2\pi f t)} dt$

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

傅立葉轉換



分解聲音的頻率，得到{各頻率}，  
以及其{大小}。

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

why it's intuitive  
如何直觀理解

$$\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \underline{e^{-2\pi i f t}} dt$$

#### 先備知識

- $e^{\theta i} = \cos(\theta) + i \sin(\theta)$  在複平面上逆時針旋轉  $\theta$  度。  
證明就是兩個複數相乘後，利用三角函數的和角公式即可。

Example:  $e^{\frac{\pi}{2}i} = \cos\left(\frac{\pi}{2}\right) + i \sin\left(\frac{\pi}{2}\right) = 0 + i$ ，可以旋轉90度。

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

why it works  
為何可以這樣做

1. 正交函數

2.  $\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \cos(-2\pi ft) + i \sin(-2\pi ft) dt$

#### 先備知識

- 所有的週期波都可以由  $\cos(n\theta), \sin(n\theta)$  合成。 $(n = 0, 1, 2, \dots)$
- 基底？直角坐標？ $(1, 0)$ 、 $(0, 1)$ ，就是一組基底。
- 想要把聲波分解，就要找到能夠組成波形的基底。

#### {正交函數}

兩個函數相乘後，對一個週期 $(0 \sim 2\pi)$ 做積分 = 0，表示兩函數成正交。  
可以用和差化積證明。




### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

- 只要  $n \neq m$  ,  $\sin(nx)$  就和  $\sin(mx)$  正交。
- 對任意  $n, m$  ,  $\sin(nx)$  就和  $\cos(mx)$  正交。

$$\begin{aligned} g(t) &= a_0 \cancel{\cos(0x)}^1 + a_1 \cos(1x) + a_2 \cos(2x) + \dots \\ &\quad + \cancel{b_0 \sin(0x)}^0 + b_1 \sin(1x) + b_2 \sin(2x) + \dots \\ &= a_0 + a_1 \cos(1x) + a_2 \cos(2x) + \dots \\ &\quad + b_1 \sin(1x) + b_2 \sin(2x) + \dots \end{aligned}$$

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

- 只要  $n \neq m$  ,  $\sin(nx)$  和  $\sin(mx)$  正交 ,  $\cos(nx)$  和  $\cos(mx)$  正交。
- 對任意  $n, m$  ,  $\sin(nx)$  和  $\cos(mx)$  正交。

$$g(t) = a_0 + \mathbf{a_1} \cos(1x) + a_2 \cos(2x) + \dots \\ + b_1 \sin(1x) + b_2 \sin(2x) + \dots$$


如果我想要求  $a_1$

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

- 只要  $n \neq m$  ,  $\sin(nx)$  和  $\sin(mx)$  正交 ,  $\cos(nx)$  和  $\cos(mx)$  正交。
- 對任意  $n, m$  ,  $\sin(nx)$  和  $\cos(mx)$  正交。

$$g(t) = \left\{ \begin{aligned} &a_0 + \mathbf{a_1} \cos(1x) + a_2 \cos(2x) + \cdots \\ &\quad + b_1 \sin(1x) + b_2 \sin(2x) + \cdots \end{aligned} \right\} \times \cos(x) , \text{再積分}(0 \sim 2\pi)$$

如果我要求  $a_1$

### 3. Extract Features: 傅立葉轉換 (Fourier transform, FT)

what it means  
它的功用為何

分解聲音的頻率，得到{各頻率}，  
以及其{大小}。

why it's intuitive  
如何直觀理解

$$\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \underline{e^{-2\pi i f t}} dt$$

why it works  
為何可以這樣做

1. 正交函數

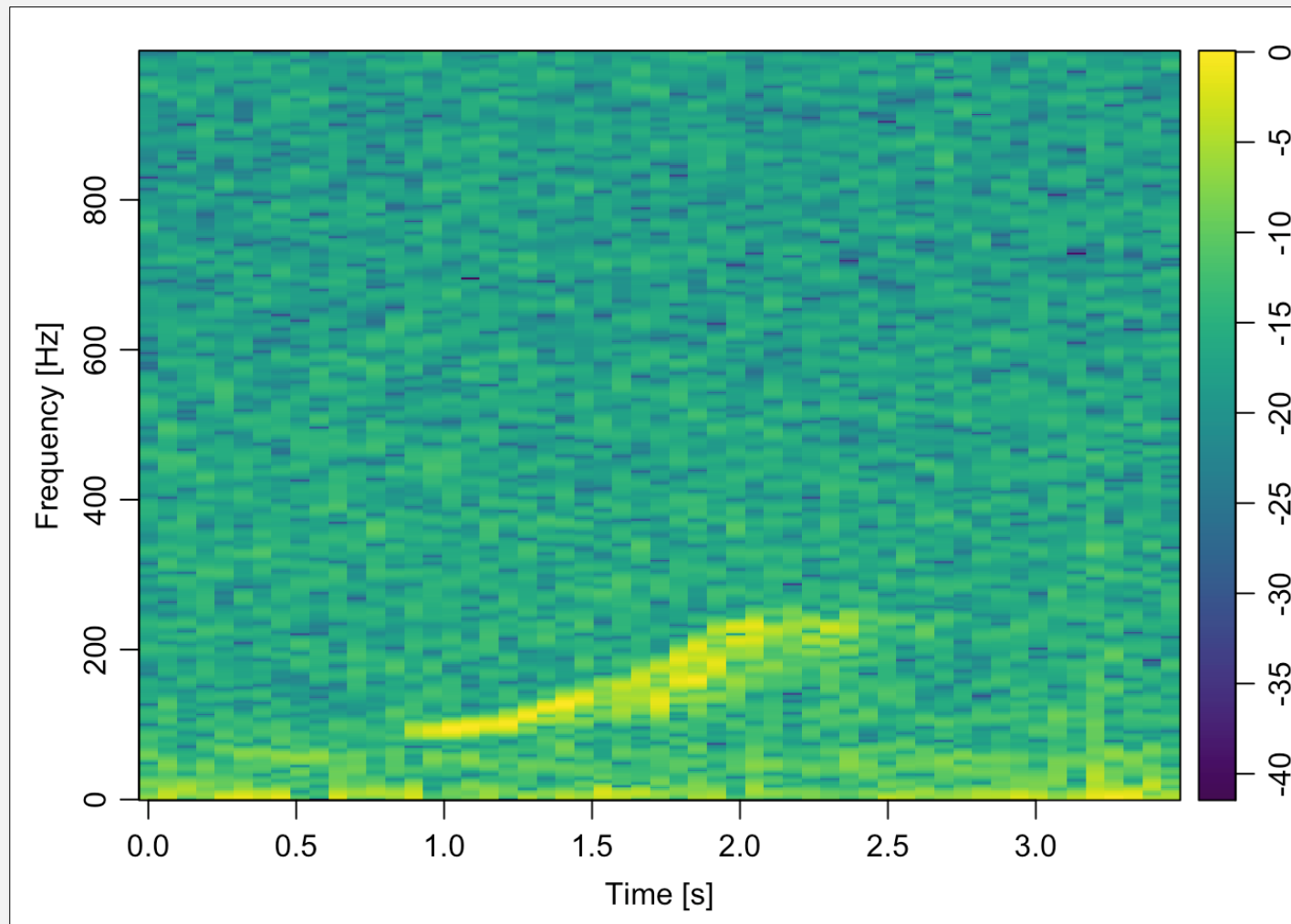
2.  $\hat{g}(f) = \int_{-\infty}^{\infty} g(t) \underline{\cos(-2\pi f t) + i \sin(-2\pi f t)} dt$

## 4. Visualization(?)

### Spectrogram 頻譜圖

不是不分享阿!!  
而是要繪製這個圖，應該還要知道短時距分析，比如

- 矩形窗(Rectangular Window)
- 漢明窗(Hamming Window)
- 漢尼窗(Hanning Window)

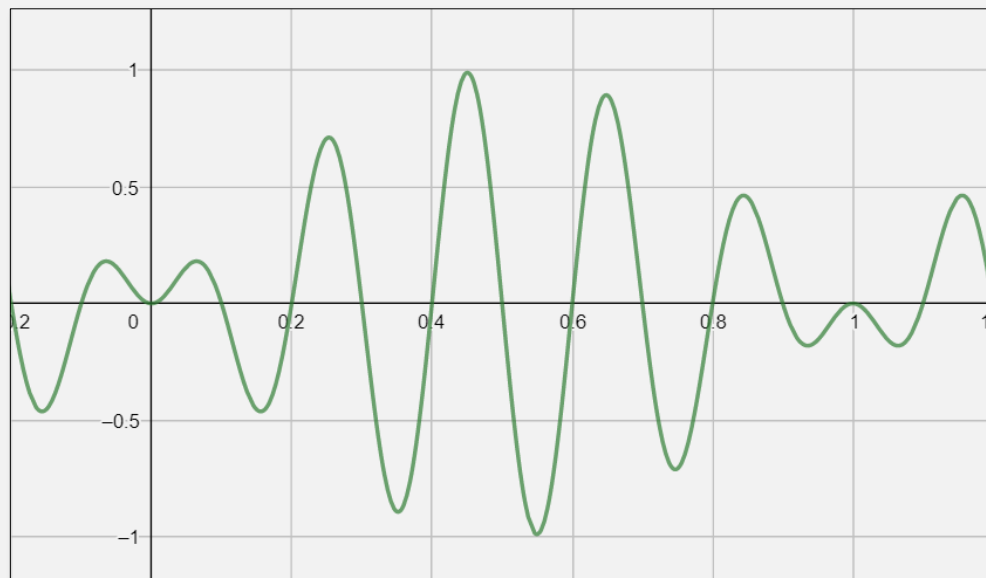
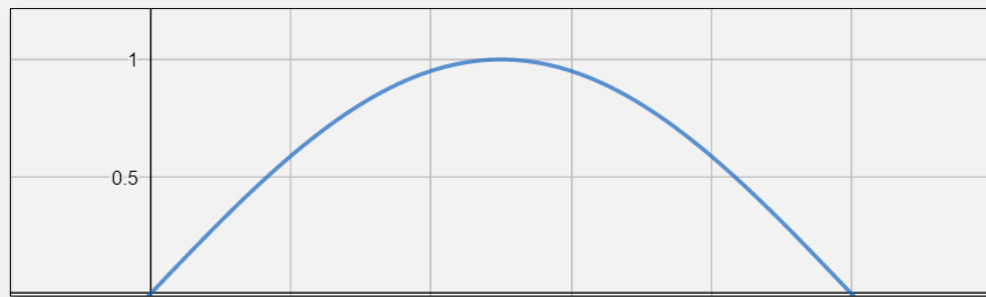
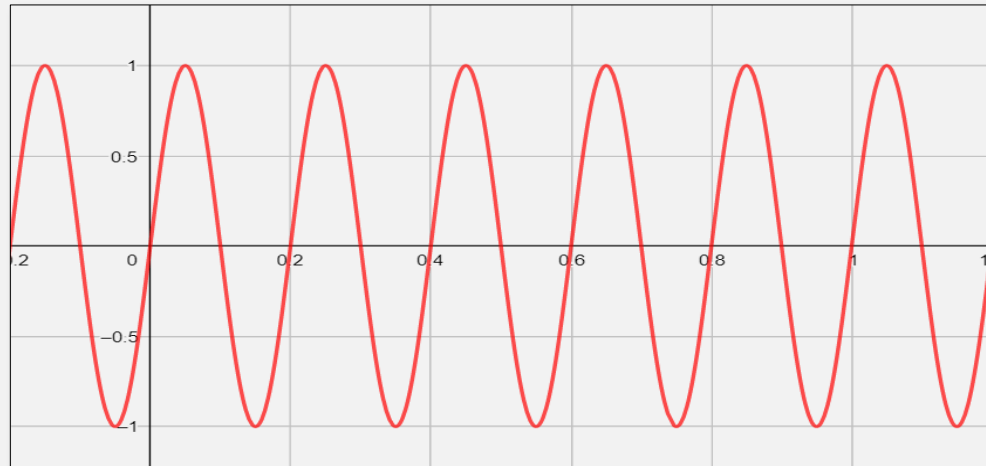


## 4. Visualization(?)

### Spectrogram 頻譜圖

不是不分享阿!!  
而是要繪製這個圖，應該還要知道短時距分析，比如

- 矩形窗(Rectangular Window)
- 漢明窗(Hamming Window)
- 漢尼窗(Hanning Window)



## 5. Some extra

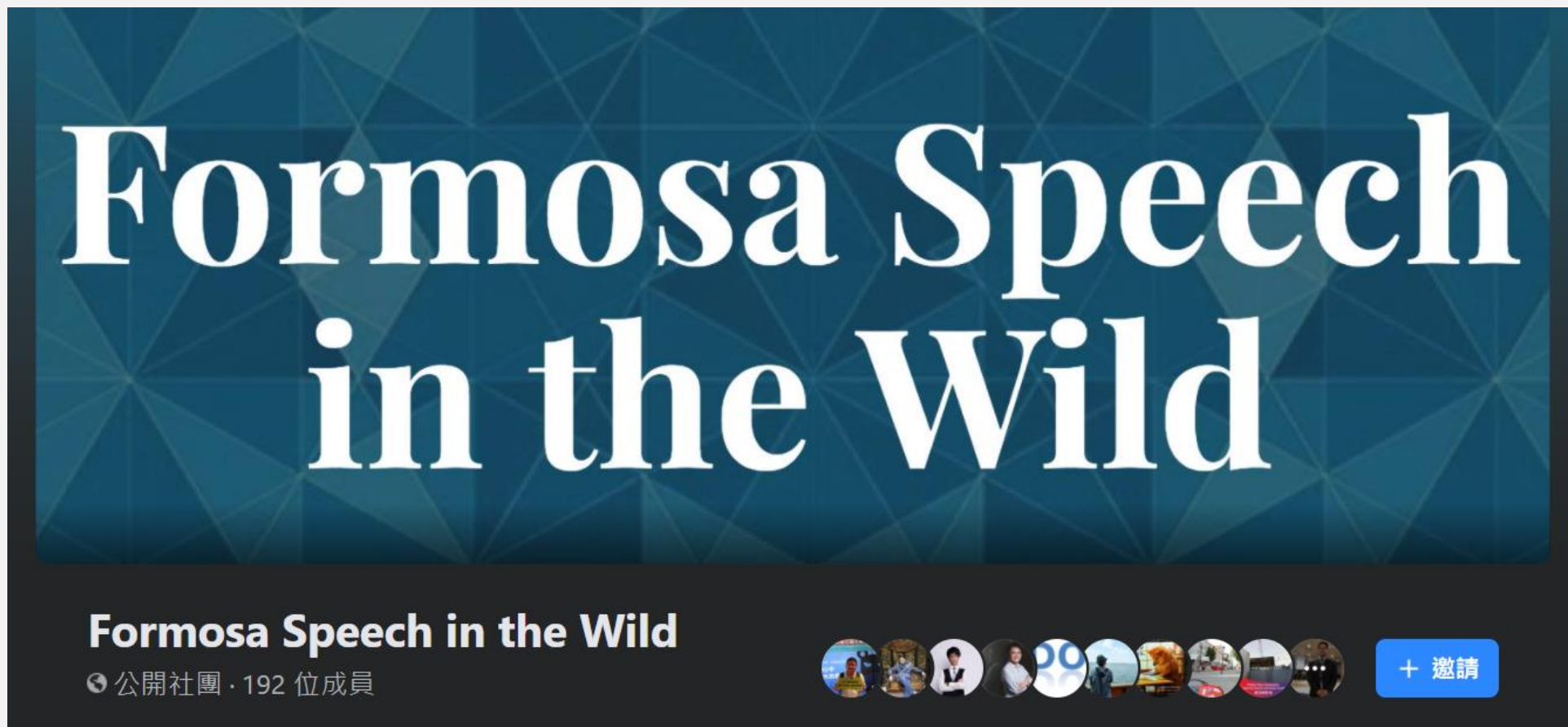
- 訊號轉換
  - z-transform
  - Fourier transform
  - discrete Fourier transform
- 濾波
  - FIR(Finite impulse response)
  - IIR(Infinite Impulse Response)
- 更多的語音處理
  - 短時距分析: 矩形窗、漢明窗、漢尼窗
  - 越零率(Zero-Crossing Rate): 估算此訊號的基本週期長短
  - 自相關函數(Autocorrelation Function)
  - 線性預測編碼(Linear Predict Coding,LPC)
  - 倒頻譜分析、梅爾頻率倒譜(Mel-Frequency Cepstrum)

諧波!!

頻譜洩漏!?!?

## 5. Some extra

FB 社團: Formosa Speech in the Wild





## 5. Some extra

Aidea 競賽: 歌聲轉譜競賽、和弦辨識。

ARTIFICIAL INTELLIGENCE COLLABORATION PLATFORM  
Aldea

About Us Topics Projects Competition Career Login

### 歌聲轉譜競賽

Begin 03/05

Introduction Rules Data

2020/07/28 歌聲轉譜  
請參賽者至AI CUP網站

+ 查看更多

#### Introduction

ARTIFICIAL INTELLIGENCE COLLABORATION PLATFORM  
Aldea

About Us Topics Projects Competition Career Login

### 和弦辨識競賽

Begin 09/14 Register end 12/11 Closed 01/09

Introduction Rules Data Upload Team up Leader board

#### Introduction

隨著時代的進展，聆聽音樂的方式已經從以往的 CD 轉移到各式各樣的音樂線上平台，如國外的 Spotify、Line Music，國內的 KKBOX、Friday Music 等。就 2016 年 IFPI 的報告指出，數位音樂的產值已經正式超過實體音樂產值，而實體音樂的產值正在連年下降中，顯示出趨勢正站在數位音樂這邊。

而數位音樂的發展帶動了許多相關的 AI 智能應用，包括原曲辨識、哼唱選歌、樂曲分類等，其中 Line Music、KKBOX 與 Spotify 都相繼成立機器學習或人工智慧部門，專門透過歌曲內容與使用者的聆聽

#### Registration

55 days	07 hrs	54 mins	31 secs
------------	-----------	------------	------------

Time to team-up deadline

# 6. Conclusion

欸~我主題是什麼??

喔對!! 面對聲音分析，數學背景和統計背景的人應該怎麼切入XD

## 數學

---

- 理解每個公式背後的推論
- 將實際問題轉換成數學問題
- 發展新的理論

## 統計

---

- 學會使用工具
- 知道什麼時候可以使用
- 解釋應用的結果
- 不要忘記80/20，發想新的應用場景

# Thank You

Github(簡報與資源)

<https://reurl.cc/GrN0jW>



Slido 有問題可以問

Code: 1017voice

