

코딩은 자동화다.

C 언어, R, 파이썬 등이 언어...

모든 lng 는 단어가 있다. 단어를 어떻게 combine 하면 커뮤니케이션이 되나.

컴퓨터 lng 와 사람의 lng 는 똑같다.

단어는 뭘까? 단어는 그 속에 의미를 포함하고 있다. 의미는 정보가 들어있는 거다.

정보를 담는 그릇 같은걸 생각하면 단어가 정보를 담는 그릇.

단어라는 그릇이 하나 있다.

정보로서의 사과를 담으면 그 단어는 사과가 되고 물을 담으면 물이 된다.

computer language 에 단어에 해당하는 부분이 변수 variable 이다.

정보를 담는 그릇이 변수로서 필요한거고 숫자도, 문자도 담을 수 있다.

(사람과 기계 간의 커뮤니케이션) 정보를 가지고 기계한테 커뮤니케이션을 하는데 문법이 필요하다.

기계와의 문법은 생각보다 그렇게 어렵지 않다.

변수라는 그 그릇에 정보를 넣는 것. 정보를 assign 하는 것. variable assignment. (1)

conditioning 에 대한 문법이 필요하다. (~할 때 이렇게 해주세요..) If conditioning 문법.

(2)

자동화의 가장 중요한 것 중 하나가 여러 번 반복하는 것. 그것은 for 라는 것을 쓴다.

for loop (3)

가장 중요한 컴퓨터 프로그래밍의 공통점.

어떤 입력을 넣었을 때 내가 원하는 출력이 나오게 함수. (1,2,3 을 이용해서 입력과 출력으로 packaging 하는게 함수)

함수 속에도 함수가 들어갈 수 있다.

재사용이 가능, 반복적 사용 가능.

<주피터 노트북>

In [] 부분 선택 해서 파란색 되면

a - 위에 셀 만들어짐

b - 아래에 셀 만들어짐

x - 셀 삭제

셀 바이 셀로 실행이 된다.

print(a) 해서 1 이 나오는건 cell 이 아니다. 이건 출력 결과일뿐.

셀 올라가서 다른 입력값 넣었던 셀 run 하고 print 하면 결과값 또 바뀜.

실행하는 단축키- shift + enter

제일 마지막에 변수명 하나를 치면 out 값으로 변수 값을 보여준다. 하나는 보여준다.
print 안해도..

싱글 quote, 더블 quote 상관 없다.

한 줄에 다 쓰고 싶으면 ; 이거 이용해서 한줄로 쓸 수도 있다.

대괄호 대신에 괄호를 써도 됨.

리스트랑 tuple 은 이름만 다르다.

괄호 쓰면 tuple,

대괄호는 리스트.

tuple 이 조금 더 보안에 강하다.

```
a = {'a': 'apple', 'b': 'banana'}
```

딕셔너리.

표제어.

coma보면 리스트에 몇 개 들어있는지 알 수 있다.

리스트랑 달리 curly bracket 을 썼다.

중괄호를 써야지 dictionary 이고 몇 개 들어갈지 부분은 콤마로.

표제어랑 설명.

fall2019 / variables -> string

변수 + 대괄호 (인덱스값)

variable[0]

a = 123; print(a[0]) >> 오류 난다. a 는 숫자 하나.

하지만,

a = [12,13]; print(a[0]) >> 12

```
a = [1,'2', [3, '4']]; print(type(a)); print(a[0]); print(a[1]); print(a[2])
>>
<class 'list'>
1 (숫자로서)
2 (문자로서)
[3, '4']
```

tuple 도 똑같음.

딕셔너리에서 인덱스 할 때는 숫자를 안쓰고
pair 부분의 앞부분을 쓴다.

```
s = 'abcdef'
print(s[0], s[5], s[-1], s[-6])
print(s[1:3], s[1:], s[:3], s[:])
#시험문제 단골
```

```
n = [100, 200, 300]
print(n[0], n[2], n[-1], n[-3])
print(n[1:2], n[1:], n[:2], n[:])
>>
100 300 300 100
[200] [200, 300] [100, 200] [100, 200, 300]
(형식 잘 봐)
```

nlp : natural language processing (자연어처리, 텍스트 분석)

rindex: 오른쪽부터 찾음. the last index.

```
s = ' '.join(tokens)          # combine the words of the text into a string using s as
the glue
s
' ' 이용해서 tokens 안에 있는 것을 join 하라.
```

function - print, len(), type() 등

앞에다가 #을 붙여서 노트를 남겨라.

code -> markdown 해도 메모가 된다. (#붙이면 크기 커짐)

모든 lng 는 for 과 if 를 쓴다.

for 함수:

in 뒤에 있는 것을 하나하나 가져와서 i 에 넣어서 실행한다.

range 라는건 뒤 괄호에 만약 4 가 들어가면 0 부터 3 까지의 리스트를 만들어준다. 4

라는건 4 개의 인덱스를 만들어줘라 라는 뜻.

```
>>
```

```
a = [1,2,3,4]
```

```
for i in range(4):
```

```
    print(a[i])
```

```
>>
```

```
1
```

```
2
```

```
3
```

```
4
```

```
>>
```

```
a = [1,2,3,4]
```

```
for i in range(len(a)):
```

```
    print(a[i])
```

```
>>
```

```
1
```

```
2
```

```
3
```

```
4
```

enumerate 번호를 매기는 것.

```
>>>
```

```
a = ['red', 'green', 'blue', 'purple']
```

```
b = [0.2, 0.3, 0.1, 0.4]
```

```
for i, s in enumerate(a):
```

```
    print(i,s)
```

```
>>>
```

```
0 red
```

1 green
2 blue
3 purple

어떤 format 으로 하고 싶을 때, 내가 넣고 싶은 변수 자리에 중괄호{} 넣어서 만든다.
format 뒤 괄호 안에 들어 있는 것들이 중괄호 안으로 들어간다.

```
>>>
a = ['red', 'green', 'blue', 'purple']
b = [0.2, 0.3, 0.1, 0.4]

for i, s in enumerate(a):
    print("{}: {}".format(s, b[i]*100))
>>>
red: 20.0%
green: 30.0%
blue: 10.0%
purple: 40.0%
```

zip 은 그 자체나 print 함수로 출력되진 않고 list(zip()) 해야한다.

zip 함수 : 동일한 개수로 이루어진 자료형을 묶어주는 함수라고
점프투파이썬에 나온다.
조금 더 내 방식으로 설명하면 **각기 다른 개수로 되어있는 자료형일지라도
동일한 개수로 쌍을 지어 묶어준다** 라고 설명하고 싶다.

If 함수

if 에서 통과가 안되면 아예 결과값 안나옴.

syntax 11 번까지 중간고사.

< numpy >

패키지 안에 패키지 만들 수 있다.

numpy.A.D.~

이런식으로 상위개념부터 쓸 수 있다. (패키지 안에 패키지...)

```
import numpy
```

한 후에 그냥 numpy.A.D.f 이런식으로 f 함수 쓸 수 있다.

from 이란 것을 쓸 수도 있다.

from numpy import A 이런식으로 ..

그 다음엔 이제 A.D.f 이렇게 접근 가능.

```
from numpy import A.D
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

matplotlib 안에 pyplot 있다.

(from matplotlib import pyplot as plt 라고 해도 똑같다. - 시험 단골 질문)

```
data = np.random.normal(0,1, 100)
```

```
print(data)
```

```
plt.hist(data, bins=10)
```

```
plt.show()
```

#정규분표

종 모양의 정규분표 만들기 위해선 min 0,

실행 안하고 싶을 때 %를 앞에 붙여라.

histogram - bins(바구니 몇개로 할지 정하는 것)

데이터값을 바구니에 넣는다. 그렇게 그래프를 그리는 것.

csv 파일 github 에 올려놓음.

데이터 줄 때도 csv 로 준다.

delimiter: 콤마로 나누자

skiprows: 처음에 x,y 써져있는 row 는 생략해라

savetxt 는 저장.

66 처럼 비교하려면 shape 와 개수 다 똑같아야함.

67 에 a 자체가 numpy. numpy 가 만들어낸 산물.

함수니까 괄호해야하고, 자기자신이니깐 괄호 안에 안써도 됨. `a.sum()`
`np.sum(a)`랑 똑같다.
numpy 가 만들어냈기 때문에 저렇게 쓸 수 있는 것이다.

```
array([[1,2,3],  
       [4,5,6],  
       [7,8,9]])
```

첫번째 차원은 1,2,3 / 4,5,6 .. 이런 행.
차원은 행 -> 열 이렇게 간다.

puretone (sin, cos wave)

sin 이나 cosin 처럼 생긴 곡선을 sinusoidal 이라고 말한다.

sinusoidal function 을 만들어 내는것을 phasor 라고 한다.

sin function, cos function 도 phasor 가 된다.

sin 하고 cos 에 들어가는 입력값이 뭐지? sin 에 90 도 넣고 싶으면 $\sin(90)$ 이 아니라
radian 값으로 넣어줘야 한다.

파이는 숫자값. 3.141592... 유리수가 아니라 무리수. 유리수는 분수로 표현되면서
반복되는게 있는 것.

0 부터 숫자가 늘어난다고 생각할 때, 2π 이면 6.28....

0 ~~~~~ 2π (degree 를 이렇게 표현하는 것을 radian)

0 도 ~~~~~ 360 도 (degree)

파이는 180 도

0 ~~~ 180 ~~~ 360 도 (degree)

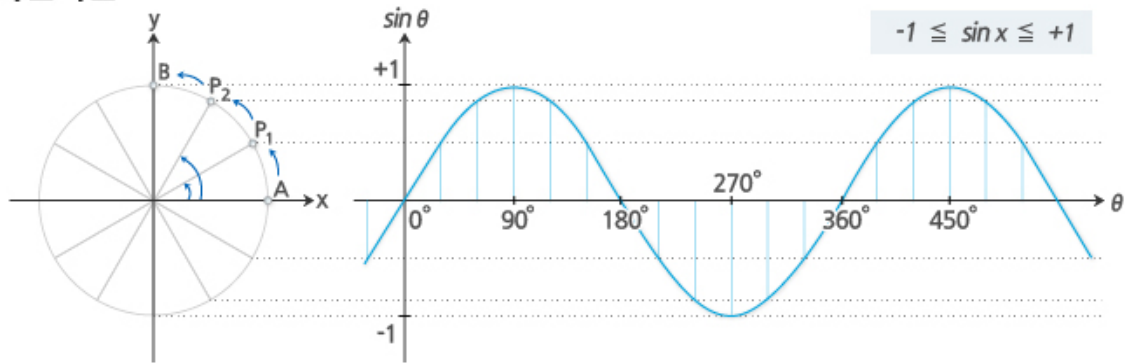
0 ~~~ π ~~~ 2π

sin, cos 에 들어가는 입력값은 radian.

2π 라 4π 는 똑같다. 반복되는 것은 똑같은 값이 나온다.

<phase>

사인곡선



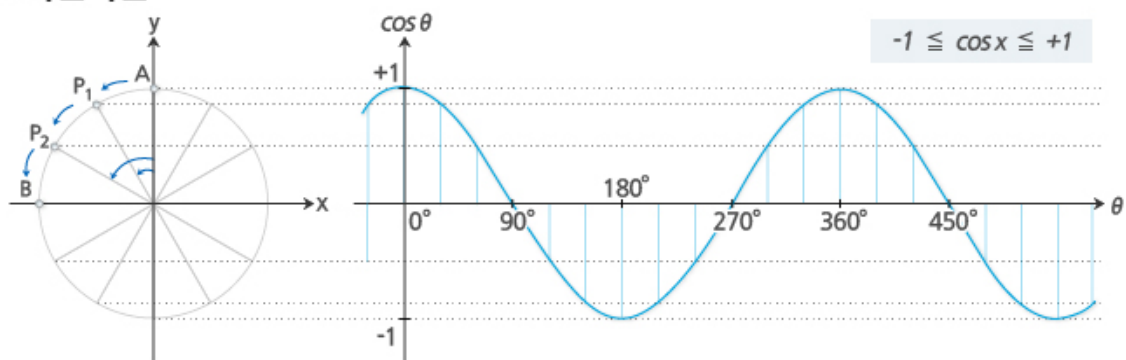
$$\sin(0 \text{ 도}) = 0$$

$$\sin(90 \text{ 도}) = 1$$

$$\sin(180 \text{ 도}) = 0$$

$$\sin(360 \text{ 도}) = 0$$

코사인곡선



$$\cos(0 \text{ 도}) = 1$$

$$\cos(90 \text{ 도} = \pi/2) = 0$$

$$\cos(180 \text{ 도}) = -1$$

$$\cos(360 \text{ 도}) = 1$$

100 파이까지 몇번 반복할까. 50 번! 왜냐하면 2 파이가 50 번 있으니까.

radian 값을 어떻게 적나? - 세타(각도 이야기함). radian 값 쓸 때 많이 씀.

$\cos(\text{세타})$

만약 세타가 $3/2$ 파이면 값은 0 이 된다.

$\sin(\text{세타})$

오일러공식: $e^{i\theta} = \cos(\theta) + i\sin(\theta)$

(e 는 상수값. e 는 2.71... 무리수, 자연로그의 밑)

(i 는 imaginary 의 약자 - 실수의 반댓말. 허수. 허수 i: 루트 -1)

세타값에 따라 달라진다.

$f(\theta) = e^{i\theta}$ 새로운 phasor 만들어냄.

(3.14 를 넣으면 $\rightarrow e^{3.14i}$ 숫자값 나옴)

복소수는 모든 수를 포함한다. 가장 큰 범위.

$a + bi$

($e^{i\theta}$ 도 $(a+bi)$ 로 표현 가능하다.

$e^{i\theta} = \cos(\theta) + i\sin(\theta)$

세타 = 0 일때 1

세타 = $\pi/2$ 일때 i

세타 = π 일때 -1

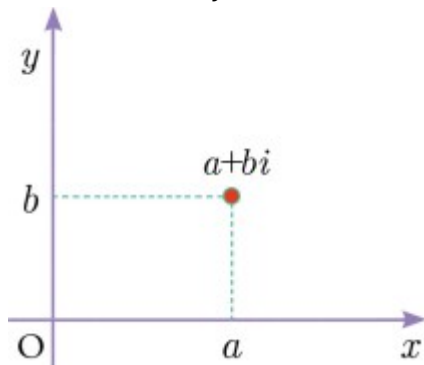
세타 = $3\pi/2$ 일때 -i (4 가지가 반복)

세타 = 2π 일때 1

복소수는 어떻게 표현할까?

complex plane 복소평면에서.

x 축에는 a. y 축에는 b.



>

그럼 다시 세타 넣은 값 가져와서

$e^{i\theta} = \cos(\theta) + i\sin(\theta)$

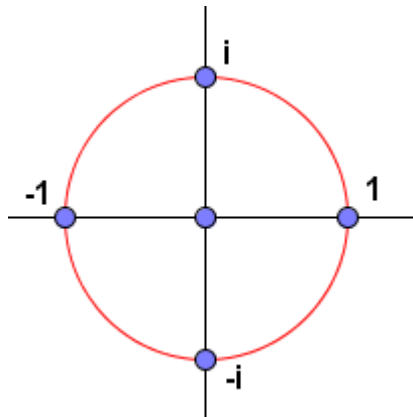
세타 = 0 일때 $1 \gg (1,0)$

세타 = $\pi/2$ 일때 $i \gg (0,1)$

세타 = π 일때 $-1 \gg (-1, 0)$

세타 = $3/2 \cdot \pi$ 일때 $-i$ (4 가지가 반복) $\gg (0, -1)$

이걸 복소평면에 찍어보면 이렇게 원이 된다.



벡터는 숫자열.

몇 콤마 몇 (,) 이것도 다 벡터.

벡터값으로 표현된다.

projection. 빛을 비추듯 보는 것.

실수만 보겠다. 하면 위에서 쳐다보면 실수에서만 왔다갔다.

실수만 보면 $\cos(\text{세타})$

허수만 보면 $\sin(\text{세타})$

그래프에서 \cos 은 1 부터 시작. \sin 은 0 부터 시작.

허수만 본다고 하면 오른쪽에서 쳐다봐서 위아래로 왔다갔다.

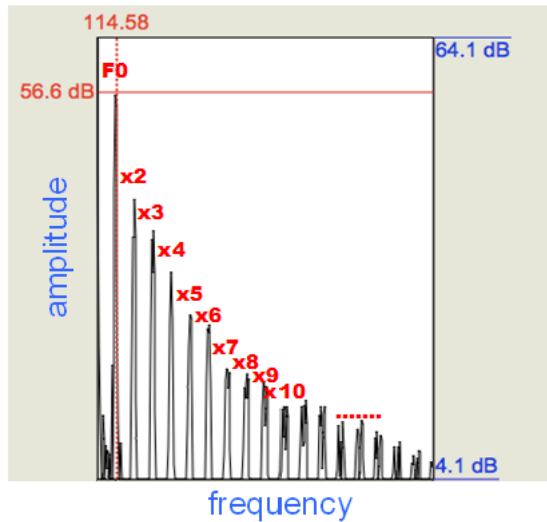
frequency 는 1 초에 몇번 왔다갔다 하는가.

시간의 개념이 들어갈까? 그냥 복소평면에서 왔다갔다 할 땐 안들어있다.

각도 개념 뿐 아니라 초 개념도 넣어줘야 진정한 소리가 완성된다.

소리라는 실체는 반드시 시간의 개념이 들어있어야 한다.

frequency: 진동수 (1 초 동안 진동한 횟수. 단위는 hz. 주파수)



gradually decreasing

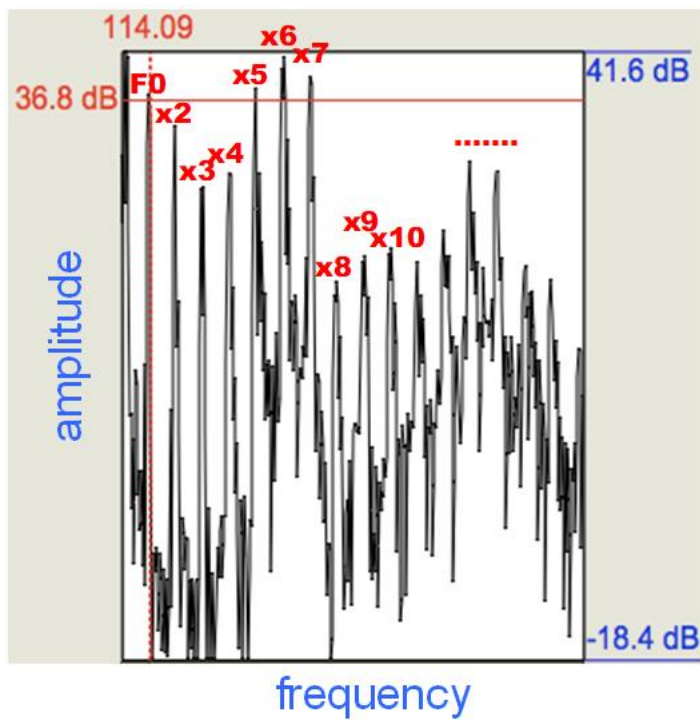
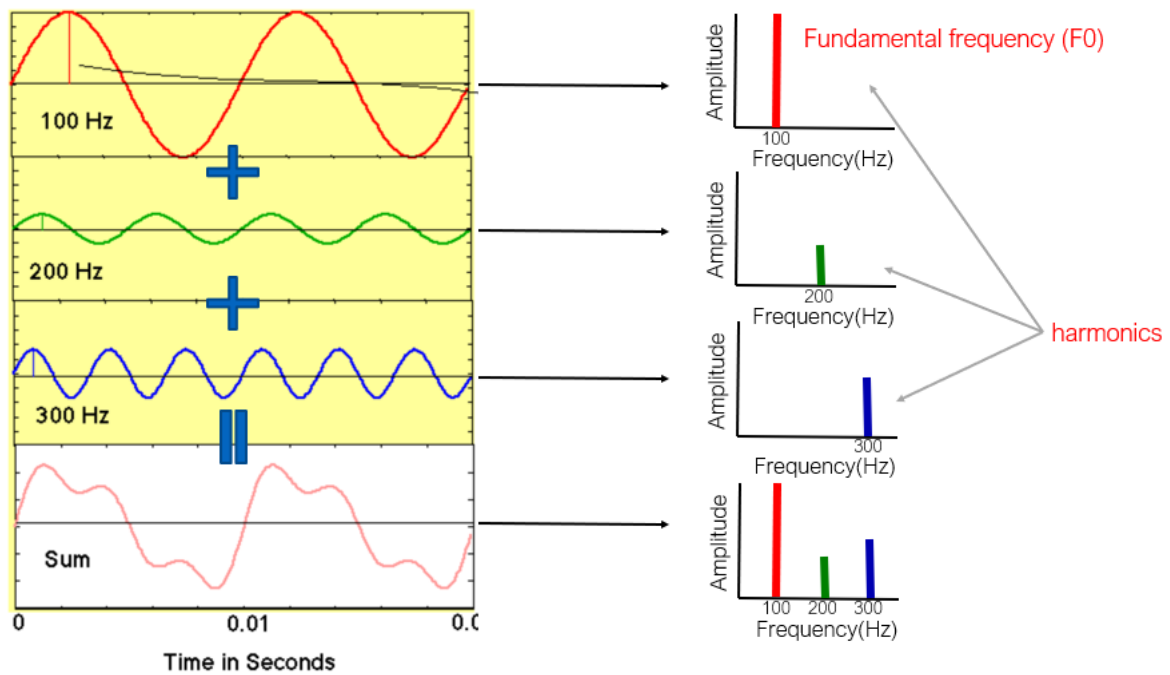
f_0 = fundamental frequency = pitch = the number of vocal vibration in a second....

(다 더해지면 frequency 제일 낮은거 따르기 때문)

(frequency 가 $x2$, $x3$... harmonics 관계.)

(우리 목소리 pitch 는 f_0 과 일치)

(여자가 남자보다 frequency 더 높게 시작해서 더 듣성듣성)

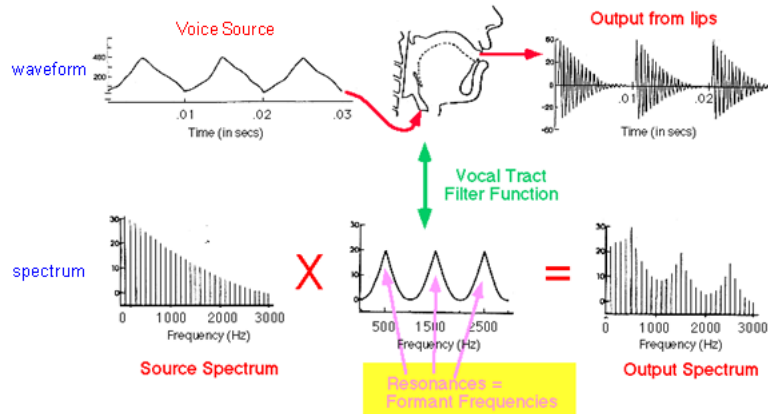


filtered by vocal tract 되면 이렇게 배음의 구조는 깨지지 않았지만, amplitude 의 패턴은 깨진다.

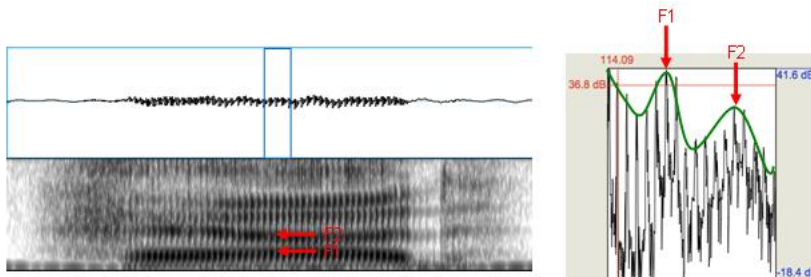
정미110
정미111
정미112
정미113

Source-filter theory

(from larynx) (by vocal tract)



Spectrogram



- Airplane view of temporal concatenation of spectrum!
- Dark band: mountains = **Formants**

f1 과 f2 만 있으면 모음 구분 가능.

f0 과 f1,f2(formant) 는 상관이 없다.

sampling rate 이 100hz 라고 생각해라.

우리가 표현할 수 있는 숫자의 개수를 1 초에 100 개라고 생각.

이걸 가지고 우리가 1hz frequency 를 표현할 수 있을까 없을까
답은 있다. 한번 왔다갔다 하면 된다.

2hz 도 가능하다. 두번 왔다 갔다.

hz 를 계속 올려서 10000hz 가 가능할까 (1 초동안 만번을 왔다갔다 하게)
1 초에 만번 왔다갔다 하려고 하는데 주어진 숫자는 100 개밖에 없다. 안된다.
sampling rate 이 1 초에 충분히 있어야 그만큼의 주파수를 표현할 수 있다.
우리가 갖고 있는 숫자가 너무 적다.

sampling rate 이 그만큼 있어야 그만큼 표현 가능.

|-----|

주어진 숫자 10 개

Sampling rate: 10hz

Frequency: 100hz

많아봤자 5 번 왔다갔다 할 수 있음. 5 번 이상 안됨.

주어진거의 반밖에 maximum 이 안된다.

nyquist frequency = sampling rate / 2

cd 음질은 sr = 44100hz

이것의 nyquist frequency 는 22050 hz (표현할 수 있는 frequency 의 맥시멈)

왜 cd 음질을 44100 로 잡았을까.

사람이 들을 수 있는 가청주파수가 20000hz 이다.

유선전화기의 sampling rate 는 8000hz.

4000hz 가 nyquist frequency.

4000 위로 누구인지의 정보가 더 있기 때문에 유선전화기로는 누구인지 구분하기
어려울 수도 있다.

핸드폰 16000hz

wave 가 여러 개 쌓여서 성대에서 나는 소리. 차곡차곡 더한다.

sampling rate 의 half 까지 더하면 pulse train 나온다. (x 축 time, y 축은 value) - wave
form

frequency 도메인으로 보면 스펙트로그램(x 축은 타임, y 축은 frequency) (어떤 frequency
성분들이 많은지)

여기서 한 슬라이스만 자른다고 생각.

스펙트럼 - 한점의 시간 (x 축 frequency, y 축 amplitude.) 스펙트럼을 시간으로 쪽
나열한게 스펙트로그램
formant 라고 하는게 산맥처럼 만듦.

1. amplitude 를 점점 낮아지게.
2. 산맥 처음 부터 하나하나 만들어주기. (사람 목소리처럼 됨)

wave form

소리를 frequency 로 보면 spectrogram

spectrum (한 점의 시간)

그걸 시간순대로 쪽 보여주는게 스펙트로그램.

gradually decreasing

점점 고주파로 갈 수록 낮아지도록.

<복습>

(3*2) 가 (2*3) 나 (곱하려면 2 가 겹쳐야 한다)

1 3 6 2 0

5 1 3 9 1

6 -1

나 6*가 1 + 나 3*가 3 = 15

결과적으로 3*3

15 29 3

33 19 1

33 3 -1

그냥 나에서 6

3 만 곱하는게 벡터

3*2 (기계) 2*1 (입력) 하면 2 없어지고 결과는 3*1 15

33

33

(출력)

순서를 거꾸로해서 6, 3 을 가에 먼저 곱하게 놓으면 2*1 과 3*2 가 겹치는게 없어서 곱해지지 않는다.

6 3 을 가로로 [6 3] 으로 놓고

[A] (3*2)

1 2

-1 0

3 5

column 쪽에서 space 생각해보면 3*2 에서 3 (칼럼 차원에서의 whole space 는 3 차원)
columnize whole space 는 3 차원.

column space 는 plane.

spanning (길게 확장시키는 것)

2 차원의 plane 나온다. 이게 칼럼 스페이스. 칼럼 벡터를 표현하고 원점과 연결해서 spanning 시키면.

colum space 와 whole space 는 안 똑같다 (전자는 2 차원, 후자는 3 차원. 후자가 더 크다)

나머지 한 차원을 우리는 left null space 라고 부른다.

영어로 orthognal 이 수직, 직각이란 뜻. 이 평면과 수직이 되는 게 뭘까? 그런 선은 하나가 있다. 이 선은 몇 차원일까?

선이니깐 1 차원. 이게 null space.

총 3 차원이 된다.

A 에 컬럼에 하나씩 a, b 를 곱한다.

plane 을 넘어서지 않는다.

벡터의 합은 평행사변형 해서 만든 점. 그렇게 해서 모든 가능한 점을 찍으면 결국엔 아까 삼각형을 span 한 것과 동일하다.

row whole space 는 2 차원.

row vector space 는 2 차원.

column 벡터 independent 한개 2 개. 이것들을 우리는 rank 라고 부른다.

row 벡터들은 절대로 independent 하지 않다. 뭘 곱하거나 더해서 (linear combination)을 통해 서로가 될 수 있다.

column 으로 생각하든, row 로 생각하든 independent 한 rank 의 개수는 항상 똑같다.
rank 는 두개로 똑같다. (independent 한 개수)

3*2 (whole space - 2 차원)

빼기

2 2 (column/row space)

하면

1 0 (이게 null space)

plane 이 있다고 생각

column space

$xA = 0$

아이겐 벡터

A 벡터에 x 를 곱했을 때 원점과 일직선상에 있게하는 모든 벡터를 아이겐 벡터라고 한다.

$Av = \lambda v$

(아이겐벡터) 어떤 벡터 v 를 곱했을때 (란다는 어떤 상수)

transformation 된게 transformation 안된거의 상수배밖에 안된다. (그냥 확장)

< inner (dot) product >

아무리 차원이 높더라도 두 벡터가 있으면 원점과 삼각형 이룰 수 밖에 없다.
(2 차원 평면)

$a = [1 \ 2 \ 3] \ (1 \times 3)$

$b = [2 \ 4 \ 7] \ (1 \times 3)$

$a * b$ 이 둘은 곱하기가 안된다.

곱하려면 b 를 $3*1$ 로 만들자. Transpose 하자. $a*b^T$

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 2 \\ 4 \\ 7 \end{bmatrix}$

이렇게 하면 31 하나 나옴. ($1*1$) (이런걸 inner product 라고 하고) ($a.b$ 라고 쓰면된다)

$a^T * b$ 해도 된다.

$(3*1) (1*3) \rightarrow 3*3$ 만든다 (이런걸 outer product 라고 한다 - 우린 안배울거다)

$a.b = |a| * \cos \text{세타} * |b|$

각도 세타 모르면 어떻게 하나?

$a.b / |a|*|b| = \cos \text{세타}$

$a.b$ 는 31. $|a|$ 는 루트 제곱들, $|b|$ 도.

\cos 세타의 값을 아니깐 세타도 알게 된다.

세타가 90 도면 \cos 세타는 0 .

inner product 도 0 이 될 수 밖에 없다.

a 라는 사람과 b 라는 사람이 있는데,

a 의 국어 영어 수학 점수가 b 의 점수.

\cos 세타가 \cos similarity. (correlation r 과 비슷. 두 벡터가 얼마나 비슷한지 비교)

두개의 벡터를 주고, 코사인 시밀러리티를 구해라.

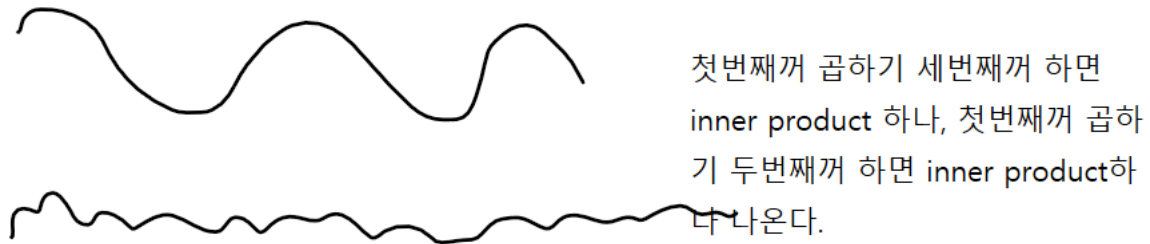
~~ 이런 그래프에 점 100개가 있다.

점 100개로 똑같은 그래프 만들 수 있다.



점 100개 이용

이것도 점 100 개, 밑에거도 점 100 개 이용



100hz 부터 해서 100hz, 200hz, ... 10000 hz 까지. (sin wave 만드는데 phase 라고 한다)
sin wave 를 똑같은 간격으로 해서 다 inner product 해버린다.

wave 개수만큼 숫자가 만들어진다.

100hz 성분은 어느정도 있구나.

알 수 있다.

스펙트로그램은 어느 부분이 진하고 어느부분이 약하다. (inner product 값과 똑같다)
타임슬라이스 하나에서.

이게 진행되면서 스펙트로그램 만든다.

스펙트로그램에선 100hz 가 제일 밑에 있다.

a,b 는 똑같은 그래프.

c 는 좀더 촘촘한 (2 배 빠른) 그래프.

a,b 는 (떨어질때 같이 떨어지고 올라갈때 같이 올라가고... 이럼 값이 크다)

a,c 는 (어떨때는 불일치. 그러므로 값이 훨씬 작을 수밖에)

어떤 복잡한 웨이브에 어떤 것을 그냥 inner product 하면
똑같은 성분 있으면 반응 크게 한다.

frequency 는 완전 똑같은데, 90 도 이동한 그래프끼리는 0 이 나온다. 맹점!
phase shift.

a 라는 벡터와 b 라는 벡터. (90 도가 되어야 inner product 0)
벡터 공간에서의 90 도와 그래프에서의 90 도가 같아져 버린다.

너무 민감하기 때문에

sin, cos phasor 말고 complex phasor 써서 inner product 할거다.

그렇게 하면 phase 에 대한 민감도 해결 가능.

a 와 b 의 길이가 고정되어 있다면,

앵글값(세타)에 따라 닷 프로젝트 값이 결정한다. 앵글이 0에 가까울수록 닷 프로젝트 크기가 커진다.

90도가 될 수록 최소(0)가 된다.

닷프로덕트: $\|b\| \|a\| \cos \theta$ (a,b 고정되어 있으면 90도면 최소, 0도면 최대)

wave 도 벡터.

어떤 웨이브 속에 어떤 사인 성분이 많은가 아는게 중요.

spectral analysis

어떤 frequency 성분이 많은가.

아무리 복잡한 시그널도 웨이브들의 합. fourier 프리에. 조금 조금한 웨이브들이 얼마씩 들어있는지 중요. 다 들어있긴 다 들어있는데, 몇 곱하는지가 중요.

그걸 알아낼때 inner product 의 기법을 쓰는것. inner product 하면 그 값으로 얼마 들어있는지 알 수 있다.

두가지 phasor 가 있다.

complex phasor ($e^{i\theta}$)

웨이브 조금 이동했다고 0이 되어버렸다. target wave 에 대해 여러가지 웨이브 만들어서 probe 한다.

target 을 조금씩 shift 했다고 해서 phasal sensitivity 가 너무 크다. 아이디어 좋았지만 쓸 수 없다. 그래서 complex phasor 써야함.

(sin 이나 cos 으로 만든거) -> 실수값 나와서 plotting 가능하다

웨이브-sin,cos / 스프링-complex phasor

우리는 그래서 sensitive 하지 않은 complex phasor 를 쓸거다. ($e^{i\theta}$ 로 만든거)

곱하기 할 수 있지만, 콤플렉스 넘버 벡터니깐 허수를 포함하는 complex number 가 나온다. -> plotting 불가능 (실수가 아니라서) (그래서 절대값 씌운다. $|a+bi|$) 절댓값은 그래프에서 원점에서부터 점까지의 길이이다.

inner product 할 땐 두개의 dimension 똑같아야 함.

<Numpy>

Numpy 도 중요하고

reshape 은 아주 중요한 function, reshape 에 대한 이해 해야함

NumPy I/O 가 어떤 역할 하는지 알아라

random 관련된건 모두 중요하다

파이썬에서 모듈 어떻게 불러와서 쓰는지두 중요하다

as from import 이런것들

matrix 이런거 안한건 안해도 된다.

matrix 의 차원이란 것은 1×4 이런거
이런걸 벡터라고 부르고
 3×4 라고 하면 직사각형
sum 을 할땐 어떤 방향으로 sum 을 할건가.

5.3 Aggregate

a 가 3×3 .

axis=0, axis=1 이런식으로 지정해줬다.

$a + b$

뭔지 알아야하고

$c = \text{np.arange}(4).reshape([4,1])$ 이런 것도 알아야한다.

<Sound> ppt

벡터의 개수, 벡터의 사이즈 차원 공간에서 한 점으로 찍힌다.
100 차원이어도 한 점에서 찍힘.

vector multiplication

벡터에 상수값 곱하기

기하학적으로 이게 어떤 의미를 가지는가.

기하학적 의미 다 이해해야 함.

$\begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

이런것도 슬쩍 지나갔어도 시험에 나올 수도.

vector spaces

linear combinations

전체의 whole space 에서 얼마나 차지하는가.

column space 라고 할 때

column vector 들의 모든 가능한 linear combination 이

column vector 들이 갖는

spanning = column(vector) space

만약 dependent 하면 아무리 linear combination 하든 될하든

그 선상을 넘어설 수가 없다.

whole space: \mathbb{R}^3 (컬럼 벡터가 갖고 있는 whole space, 로우 벡터가 갖고 있는 whole space 다 있다)

Column space: \mathbb{R}^3

transpose 에 대한 이야기를 했다: column 의 관점에서

2 차원 plane

45pg ppt 중요!

\mathbb{R}^2 를 넘어설 수가 없다.

redundant 한 경우

44pg 는 redundant 한 경우가 아니다.

one matrix.

column space, independent 하니깐 plane

로우스페이스는 2 차원.

독립적인거 rank

남는 부분을 null space.

48pg : 위에꺼 column vector 관점에서의 whole space.

밑에꺼 row vector 관점에서의 whole space.

두개는 완전 다른 두개의 세계. (독립)

위에꺼 한 차원이 남는다.

남는 부분을 null space.

null space 가 무슨 의미 갖는가 그정도만 알고 있으면 된다.

58pg $Ax=b$ 개념 매우 중요. 시험에 나온다.

matrix A 가 곱해져서 출력 벡터 b 가 나온다.

차원이 중요하다. 몇바이몇인지.

x 가 만약 3×1 이면 A 가 뭐 바이 3 밖에 되지 않는다.

transformation matrix = A

transformation 의 형태에 따라서 바뀔 수 있다.

61pg : 왼쪽 matrix 가 기하학적으로 어떤 역할을 하는가.

62pg: 왼쪽 column 1,0 / 오른쪽 column 0,1

78pg Detransformation inverse matrix

움직였던 것을 다시 inverse.

입력으로 갔던걸 다시 출력으로 오게하는

$$A^{-1} \cdot b = x$$

(원래는 $A \cdot x = b$ 였다)

eigenvector 가 제일 마지막으로 중요한걸로 나오는데

제일 중요한건 99pg 개념 이해하는 것

반드시 1,1 일 필요는 없지만 이해 돕기 위해서.

어떤 matrix 에 아이겐벡터가 뭐냐.

matrix 가 있어야 한다.

그 matrix 이 아이겐벡터가 뭐냐.

원점과 입력과 출력이 일직선상에 놓이는 경우가 생긴다.

109pg 오른쪽.

이런 벡터를 다 찾아보면 이 선상에 있는 모든 것들은 우리가

이 조건을 만족하게 된다.

transformation 거치더라도 원점과 자기자신, 결과값이 일직선상에 있다.

$$Av = \lambda v \quad (\lambda = \text{eigenvalues}) \quad (v = \text{eigenvectors})$$

이 식이 의미하는 바를 이해해야 한다.

109pg 에서 v 도 eigenvector 고 저 선상에 있는 모든 v 들은 eigenvector 가 된다. 우리가 벡터란 얘기할 때 방향에 더 관심 갖는다.

eigenvector 는 저 v 하나다 라고 얘기해도 괜찮다.

eigenvalue 가 뭐냐.

eigenvalue 는 여기서 저 상수값에 해당하는 것.

Av 까지 가는 값의 몇배가 되는 ratio.

5 배다. 그러면 란다 값이 5. 아이겐벨류

이 matrix 는 2×2

아이겐 벡터 2×2 에선 두개 존재

3×3 에선 세개 존재.

한 방향으로만 존재하는게 아니다.

그 방향에 해당하는 란다 값이 또 따로 있다.

<Sound> jupyter notebook

phasor

parameter 바꾸면 새로운 phasor 생김.

원래 기본적으로 phasor 의 인풋은 theta.

실제 소리를 생각을 할 때 소리를 만들 때 사인웨이브를 만들기도 해야하지만 시간도 할당되어야 한다.

각도값에다가 미리 time 값을 입력해야 나중에 plot 을 하던지 play 를 할 수가 있다.

plot 에 대해선 각각의 axis 가 뭘 의미하는지 정확히 알아야한다.

sin wave 에서는 한 점이 벡터다.

무슨 벡터? 시간 콤마 값이 되는 벡터. (처음 나오는 그림)

밑에서보면 한 점은 3 차원 벡터. (스프링같은 그림)

t, a, b 이렇게 있는.

어떤 차원에서 보느냐에 따라서 하나는 sin, 하나는 cos 이 보여진다.

imaginary 에서 왔다갔다하는건 sin 과 같다.

위에서 보는건 cos 과 일치.

Generate pulse train 이건 중간고사 이전과 연결된다.

저 부분을 보지 않더라도 이해하면서 쓸 수 있도록 준비해라.
무한대로 하면 진짜 줄만 생긴다.

given sampling rate 에 대해서 표현 가능한 최대치가 얼마가 되는가.
어떤 mp3 는 표현 가능한 frequency 는 5000hz 밖에 안된다.
돌고래(5000hz 이상) 의 소리는 녹음이 안된다.
돌고래가 20000hz 의 아주 고주파를 낸다.
그러면 sampling rate 를 40000 이상 높여야 고주파가 잘 잡힌다.
초음파 (high frequency)가 100000hz 라고 치면 sampling rate 이
20 만 정도 있어야한다.
표현 가능하다는게 뭔지 알아야한다.

우리가 만약 pulse train 을 만들면 그 pulse train 의
x 축은 time. 단순한 wave.
이 웨이브를 주파수로 표현할 수 있어야 한다.
어떤 주파수가 많은지 분석. spectrum.
pulse train 의 spectrum 을 보면 harmonics. 제일 베이스 되는 것부터
2 배, 3 배가 되는게 같은 크기만큼 들어가있다.
사람의 목소리는 고주파로 갈수록 점점 약해진다.
제일 저주파는 우리의 피치와 일치. (f_0 . 이것의 자연수배가 된다)

스펙트럼 상에서 산맥을 우리가 잘 꺾어야한다.
고주파로 갈수록 amplitude 를 약하게.
산맥을 어디에 위치시키느냐에 따라서 다른 자음이 나온다 -> 이걸 틀린말
모음이 달라진다.

제일 첫번째 한 작업이 In [126]
resonance 라는 누가 만들어놓은 함수 이용.
RG 는 산맥을 어디에 위치시킬까.
BWG 는 산맥을 얼마나 뚱뚱하게 만들건가. 크면 클수록 완만.
100 이면 아주 완만.
산맥을 0 에다가 만들었다. 그러면 고주파로 갈수록 완만히 꺾는 첫번째 작업을 한다.
이게 성대에서 일어나는 일.
이 이후가 vocal tract 이 filter 역할.
500 에 하나, 1500 에 하나, 2500 에 위치시킨다. (1,2,3 formant)

이게 영어에서 얘기하는 schwa 와 일치한다.
스트레스가 없는 부분을 schwa 라고 부른다.
3500, 4th formant 까지 만들어보았다.

제일 마지막은 입술에서 나가는것.
In 131. 더 크게. 퍼져나가게.

Fourier transform 이 제일 중요.

```
omega = 2*np.pi*n/nFFT # angular velocity  
z = np.exp(omega*1j)**(np.arange(0,nSamp))
```

이 부분이 제일 중요.
어떤 길이의 wave 가 있다고 생각할 때, 그 길이만큼 for loop 를 하는 거다.

complex phasor 를 써서 하는거 중요.

```
amp.append(np.abs(np.dot(s,z)))
```

왜 absolute 썼는지 중요.
complex number 의 절댓값이 뭐냐.
복플렉스 도메인의 a 콤마 b 를 찍고
dot product 가 이번학기에 제일 중요한거.

plot 절댓값을

In 134 : half 는 말이 안된다.
half 에 해당하는걸 spectrum .
이 스펙트럼을 다시 visualize 한걸 스펙트로그램.
이걸 어떻게 스펙트로그램화가 되는가. 이게 중요하다.

preprocessing signal

In 101 어떻게 analyze