<영어음성학 Summary>

<Praat>

- Sampling Frequency는 음질이 얼마나 좋게 하느냐를 결정한다. 음질을 좋게 하려면 수치를 높게 하고, 음질이 좋지 않더라도 파일 크기를 작게 하려면 수치를 낮게 한다.
- 파이썬과 같은 코딩을 통해 praat과 같은 프로그램을 만든다.
- x축은 시간을 의미. 등간격의 시간에 해당하는 수치가 있고 이들을 연결한 것이 소리. 소리는 숫자이다. 시간의 간격을 조절을 할 수 있는데 이에 해당하는 것이 Sampling Frequency이다. 예를 들어, 44100Hz라는 것은 1초를 44100으로 나눈 것을 의미한다.
- Edit에 cut를 누르면 원하는 부분을 자를 수 있다.
- 파란선은 pitch를 뜻하고, 여자이면 높은 pitch 값을, 남자이면 낮은 pitch 값을 갖는다. 노란선은 intensity를 뜻하고, 강도를 의미한다. Pitch가 높으면서 intensity가 낮을 수 있고, pitch가 낮으면서 intensity가 강할 수 있다.
- 아래에 있는 것을 소리의 스펙트럼이라고 한다.

<phonetics>

- English vowels들은 Monophthongs/ Dipthongs로 구분할 수 있다.
- phonology 음운론. Phonetics는 음성학. Phonology는 우리 속에 이루어지는 abstract한 과정. 음성학은 physical하다. 이 수업은 physical한 것에 초점을 맞추는 수업.
- Phonetics는 articulatory, acoustic, auditory phonetics로 나뉠 수 있다.
- Phonotation process를 담당하는 larynx는 voicebox라고도 불리는데, 여기서 성대 닫혀 있으면 voiced가 되고, 열려 있으면 voiceless가 된다.
- Oro-nasal process는 velum이 lowered 혹은 raised 되었는가에 따른다. Velum이 lowered 되면 m, n, ng, 모음과 같은 소리가 나고 나머지 음들은 velum이 raised되어 nasal tract이 막혀있게 된다.
- Articulatory process를 담당하는 부위는(constrictor) lips, tongue tip, tongue body가 있다. 각각의 constriction이 위치는 어디서 나는지(CL), 어느 정도로 나는지(CD)를 구분해야 한다. CL에 따라 음들을 lips, tongue body, tongue lips로 구분할 수 있고, CD에 따라 소리들을 constriction이 강하게 일어나는 순서대로 stops, fricatives, approximants, vowels 순으로 구분할 수 있다.

<Spectrum>

- Sine wave를 결정하는 것은 frequency와 magnitude. X축은 시간, y축은 수치 값(value)를 뜻한다. Sine wave를 x축은 frequency, y축은 amplitude로 나타낸 그래프로 변형할 수 있다.
- 모든 소리 signal들은 여러 간단한 sine wave의 합으로 나태낼 수 있다. 우리가 일상에서 듣는 모든 소리는 complex tone이다. Complex sound의 주기는 frequency가 가장 낮은 wave의 주기와 일치한다.
- Sine wave로 분해하는 과정을 analysis라고 하고, 합치는 과정을 synthesize(합성)이라고 한다.

<Source and Filter>

- Larynx에서 진동하는 것을 source라고 하고, 그 이후의 입 구조를 거치는 과정을 filter라고 한다. 똑 같은 soruce라고 하더라도 어떠한 filter 과정을 거치느냐에 따라 다른 소리가 난다.
- Source의 패턴을 보면 frequency가 배로 늘어날수록 일정하게 decreasing한다.
- 가장 첫번째 주파수를 f0 혹은 fundamental frequency라고 한다. F0에다가 X2, X3 배수하는 것을 harmonics라고 한다.
- Praat의 아래쪽에 볼 수 있는 것이 Spectrogram이라고 한다. x축은 시간, y축은 frequency. 까맣게 되어 있을수록 크기가 세다.
- 소리가 filtered를 거치게 되면 amplitude의 일정한 구조가 깨지고 지그재그처럼 된다. 하지만 배음의 구조는 여전히 유지한다.
- 주파수가 선호하는 곳을 formant, 산맥이라고 부르고 선호하지 않는 곳을 valley라고 부른다.
- Pure tone을 여러 개 만드는데, 주파수는 x2, x3 배수로 늘려가고 amplitude는 조금씩 낮춰가면서 만든 다음 synthesize하게 되면 complex tone이 만들어지게 된다. Combine to stereo는 합하기 전 병렬한 상태이고, mono는 하나로 합친 상태이다
- 가장 첫번째 formant를 f1, 두번째 formant를 f2라고 하고, spectrogram에 나타난 f1, f2의 위치를 확인할 수 있다. 그리고 각각의 영어 모음을 발음했을 때 f1과 f2의 위치가 각각의 모음에 따라 다르다. F1과 f2의 수치를 각각 y축과 x에 위치 시켜놓는다면 이는 바로 모음을 발음할 때 우리 혀의 위치를 결정하게 된다. F1은 혀의 높낮이, f2는 혀의 앞뒤 위치를 결정한다.

<Python>

- 변수와 정보

- 모든 컴퓨터 언어는 단어와 문법으로 이루어져있다.
- 정보를 담는 단어에 해당하는 것이 변수(variable)이다. 변수라는 그릇에다가 정보를 담는(assign)다. 숫자, 문자 등을 담을 수 있다.
- 모든 컴퓨터 문법은 if로 시작하는 Conditioning(~하면 ~해라)이 필요하다. If conditioning
- 여러 번 반복을 실행하는 for문법이 필요하다. Ex) 10번 반복해라
- 어떤 것을 입력하면 내가 원하는 것이 나오는, 내부적으로는 복잡한 함수. 입력과 출력을 패키징하는 것을 함수라고 한다. Ex) 자동차라는 함수. 나의 움직임은 입력이 되고 차가 움직이는 것은 출력.
- = 표시는 절대 같다는 의미가 아니라 오른쪽에 있는 정보를 왼쪽 variable에다가 담는다는 의미.
- Variable에 들어가는 정보의 종류는 숫자와 문자.
- 왼쪽을 클릭하고 a를 클릭하면 윗칸에다 셀 하나가 추가되고, b를 클릭하면 밑칸에다 셀 하나가 추가된다. X를 누르면 셀이 삭제된다.
- 쉬프트 + 엔터는 실행(running)의 단축키
- 여러 셀에 적을 필요 없이 세미콜론';'을 붙여 한 셀에다가 붙여 써도 된다. 혹은 한 셀에다가 엔터를 치는 것도 가능하다.
- variable에다가 하나의 정보를 assign한 다음에, 그 밑에다가 똑같은 variable에 다른 정보를 넣으면 그 정보로 replace 된다. Ex) a = 1을 assign하고 다음 cell에다가 a = 2라고 assign을 하면 a에는 2가 들어가게 된다. 그 후에 a = 1을 다시 run하게 되면 a에는 1이 들어가게 된다.
- 영어 알파벳을 그냥 쓰게 되면 그것은 무조건 변수를 의미하게 된다. 따라서 b = love 를 입력하면 오류가 난다. 만약 love = 2라고 입력하면 love라는 variable에 2를 넣는다는 것을 의미한다. 문자는 반드시 '' 혹은 "" / single quote, double quote 안에 입력해야 한다.
- variable에다가 variable을 넣는 것도 가능하다. Ex) love = 2; b = love; print(b) 하면 2가 나온다.
- #을 붙이면 뒤에 무엇을 쓰더라도 실행이 되지 않기 때문에 노트를 남길 수 있다. 혹은 셀의 type을 code에서 markdown으로 변경하면 그 셀을 주석처럼 사용할 수 있다.

- List, dictionary, tuple

- a라는 변수에 여러 정보를 담으려면 대괄호 [ ]안에다가 다 넣으면 된다. list 안에는 문자, 숫자, 리스트가 들어갈 수 있다. 각각의 사이사이에는 콤마로 구분한다.

- 대괄호[ ] list대신 괄호( )를 써도 되는데 이것의 타입은 tuple이다. Tuple은 list는 동일하지만 tuple이 보안에 더 강하다

- 중괄호{ }는 dictionary이다. 여러 개의 정보는 콤마로 구분, 표제어와 설명으로 구성, 표제어와 설명 사이에는 :(페어, 콜론)로 구분, ex) a = {'a': 'apple', 'b': 'banana'}

- 함수

- Print도 누가 만들어 놓은 함수이다. 좋고 유용한 함수들을 이미 만들어 놓은 것이 아나콘다, 파이썬이다. 함수를 쓰는 법은 괄호에다가 변수들을 입력 해주면 된다. Print(), print(a) 라고 해도 되지만 그냥 셀의 마지막에 a라고 하면(꼭 마지막에 해야 됨) 변수 a를 print out해준다.

- Type()은 괄호 안에 들어가는 variable이 어떠한 유형인가를 밝히는 함수. 안에 들어가는 변수가 리스트인지, 숫자인지 문자인지 변수의 타입을 알려준다. Int(숫자), list(리스트), str(스트링, 문자), Float(실수. Ex1.2), dict(딕셔너리)

- Float는 variable을 float로 바꾸어 주는 함수.
  ex) a = 1.2; a = int(a); print(a)
  a는 float(소수점 숫자)였는데 int함수 하니까 소수점이 날라가서 1이 된다.
  Ex) a = 1; a = float(a); print(type(a)); <class 'float'>.
  Variable a는 int였는데 이것을 float로 바꾸어주고 print하니까 float가 나온다.

- Int는 variable을 int로 바꾸어주는 함수.
  ex) a = 1; a = float(a); print(a)
  a는 int(소수점 없는 숫자)였는데 float함수 하니까 소수점이 붙어서 1.0이 된다.
  Ex) a = 1.2; a = int(a); print(type(a)).
  Variable a는 float였는데 이것을 int로 바꾸어주고 print하니까 int가 나온다.

- List에서 정보 access하기

- Ex) a = [1, 2] ; b = [3, 4] ; c = a[0]+b[0] ; c 결과는 4가 나온다. 왜냐하면 a[0]이라는 것은 variable a에서 첫 번째 정보를 가져오라는 것을 의미. 이것을 list에 access한다고 한다. a 중에서 어떠한 정보만 가져오라는 의미.

- a = '123'; print(type(a)); print(a[1])
  a = [123, 124] ; print(type(a)); print(a[1])
  a = 123; print(type(a)); print(a[1])
  가장 밑에 것은 에러가 나는데 왜냐하면 int에서 몇 번째 것을 가져오는 것은 불가능

- a = '123'; a = list(a); print(type(a)); print(a); print(a[2])
  str이었던 variable a를 list로 만들기

- a = [1,'2', [3, '4']]; print(type(a)); print(a[0]); print(a[1]); print(a[2])
  list안에 list가 있는 형태. 첫 번째 정보는 숫자. 두 번째는 문자, 세 번째는 리스트에 access한다.

- a ={"a": "apple", "b": "orange", "c": 2014} print(type(a)) print(a["a"])
  dict의 정보에 access할 때는 페어":"에서 앞부분을 index로 사용한다. a["a"]
  a ={1: "apple", 2: "orange", 3: 2014} print(type(a)) print(a[1])
  표제어에 "a"와 같이 str말고도, 1, 2, 3과 같은 int가 들어가도 무관하다. 표제어가 1이기 때문에 똑같이 apple이 뜬다.

- a=[(1,2,3), (3,8,0)] a[0] type((a[0]))
  a의 0번째 것은 (1,2,3)이 나올 것이고, a의 0번째 것의 type은 tuple이 된다.

- s = 'abcdef' print(s[0], s[5], s[-1], s[-6]) print(s[1:3], s[1:], s[:3], s[:])
  <class 'str'>, a f f a, bc bcdef abc abcdef

- 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]

- len(s) length 6

- s[1]+s[3]+s[4:]*10
  bdefefefefefefefefef

- s.upper()
  'ABCDEF'

- String

- s = ' this is a house built this year.₩n'

- result = s.find('house') result 11

- result = s.find('this') result 1
  this가 두 개 있는데 이 함수는 첫 번째 this를 찾아줌

- result = s.rindex('this') 23

- s = s.strip() s 'this is a house built this year'
  스페이스 같은 잡스러운 것을 없애고 순수한 text만 남겨주는 기능

- tokens = s.split(' ') tokens
  일단 s에 들어가 있는 문장은 바로 바로 위의 strip을 거친 문장이다. 스페이스를 기준으로 split을 해주는 기능.
  ['this', 'is', 'a', 'house', 'built', 'this', 'year']

- s = ' '.join(tokens) s
  'this is a house built this year.'
  cf) s = ','.join(tokens) s
  'this,is,a,house,built,this,year'

- s = s.replace('this', 'that')
  'that is a house built that year.'
  모든 this를 that으로 바꾸는 것.

- for loop, if conditioning

- a = [1, 2, 3, 4]
  for i in a:
  print(i)
  a에 있는 것을 하나씩 돌려서 i에 넣어라

- a = [1, 2, 3, 4]
  for i in range(4):
  print(a[i])
  range함수는 0부터 뒤에 있는 숫자 직전까지 리스트를 만들어 준다.

- a = [1, 2, 3, 4]
  for i in range(len(a)):
  print(a[i])
  len(a)는 4가 되니까 range(4)가 되고 이는 0부터 3까지 루프가 돌게 된다.

- a = ['red', 'green', 'blue', 'purple']
  for i in a:

print(i)

각각의 str이 print out된다.

- 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))

  enumerate함수는 i에는 0, 1, 2, 3가 순차적으로 루프에 들어가고, s에는 red, green, blue, purple이 순차적으로 루프에 들어가게 된다.

- a = ["red", "green", "blue", "purple"]

  b = [0.2, 0.3, 0.1, 0.4]

  for p, q in zip(a, b):

  print("{}: {}%".format(p, q*100))

  p, q에 a, b에 담긴 것을 순차적으로 루프에 넣는다.

- a = 1

  if a == 0:

  print(a)

  else:

  print(a+1)

- for i in range(1, 3):

  for j in range(3, 5):

  print(i*j)

  i에 1부터 2까지 넣고, j에 3부터 4까지 넣는다.i의 첫번째 루프에 1이 들어갈 때 j의 첫번째 두번째 루프가 3, 4에 해당하고 i의 두번째 루프에 2가 들어갈 때 j의 첫번째 두번째 루프가 3,4에 해당하니 값은 3,4,6,8이 나온다

- for i in range(1, 3):

  for j in range(3, 5):

  if j >=4:

  print(i*j)

  j가 4보다 크거나 같으면 print out 한다. 값은 4, 8이 나온다.