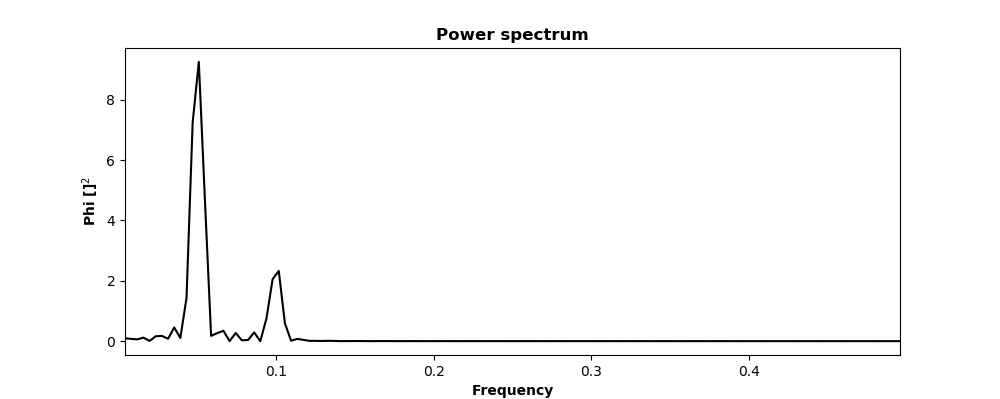
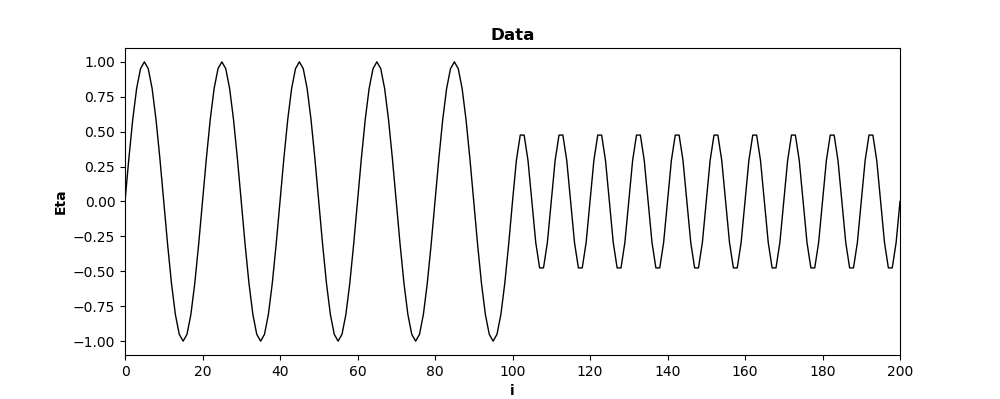
**객관적자료분석 HW#8**

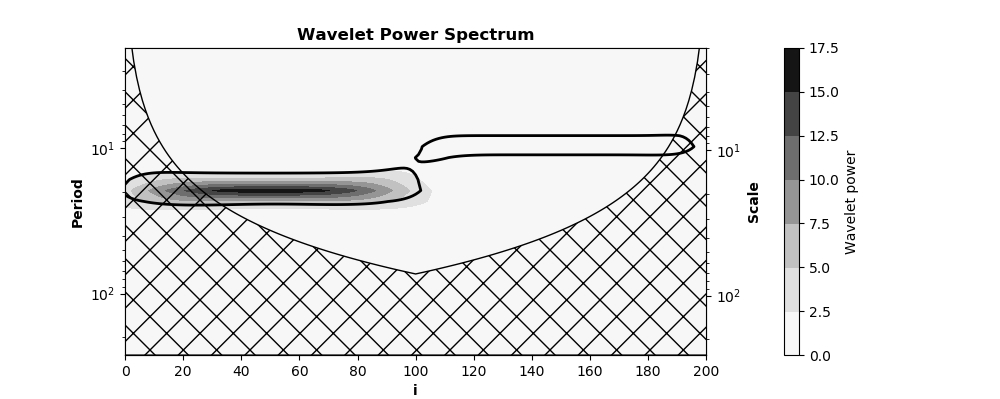
2017-29008 백관구

**1.**

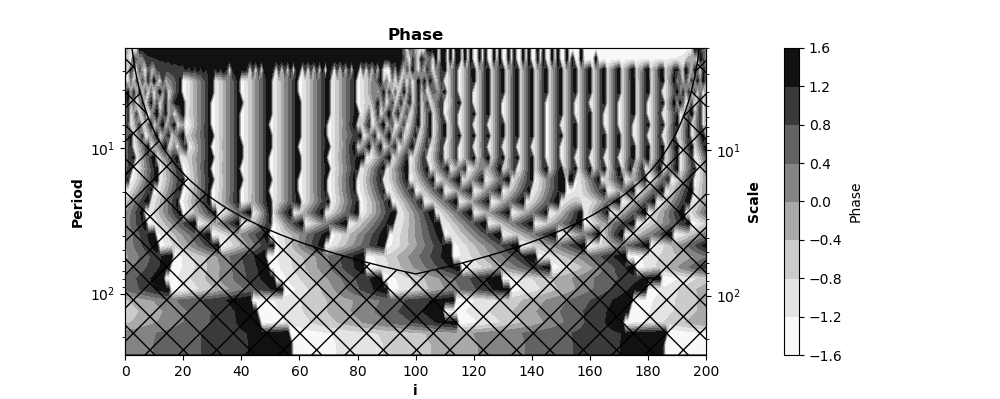
문제에서 주어진 함수를 나타내면 아래의 그림과 같다. i가 0부터 100까지는 주기가 20이고 100 이후로는 주기가 10인 함수로 나타난다. 진폭도 1/2로 줄어든다. Wavelet 분석에 앞서 파워 스펙트럼 결과를 볼 필요가 있다. 주기가 강한 자료의 특성을 잘 나타내는 결과가 확인되며, 진동수 0.05, 0.1 (주기 20, 10)에서 강한 에너지가 잡힌다. 또한 에너지의 크기에 차이가 있다. 이러한 결과에서는 시간에 따라 변하는 자료의 특성(주기)을 완전히 파악하기 어렵다. 즉 파워 스펙트럼 분석은 stationary한 자료를 분석하는 데에 유리하다. 반면 wavelet 분석은 시간에 따라 자료의 특성이 변화하는 때에도 적용할 수 있다.



Morlet wavelet 분석을 위해 으로 설정하고 분석 결과의 신뢰도를 파악하기 위해 1-lag autocorrelation을 계산하여 95% 유의수준에서의 결과를 굵은 실선으로 나타내었다. 파라미터는 매트랩 코드에서 주어진 것과 동일하게 설정하였다. 그리고 cone of influence을 그물 모양으로 나타내었다. 이 영역에서의 결과는 해석에 포함하지 않는 것이 일반적이다. 결과를 분석하면 i가 100보다 작은 곳에서는 20 주기, i가 100보다 큰 곳에서는 10 주기의 에너지가 강하게 나타난다. 특히 주기 20의 에너지가 i가 100 이하일 때 더 큰 에너지를 가진다.

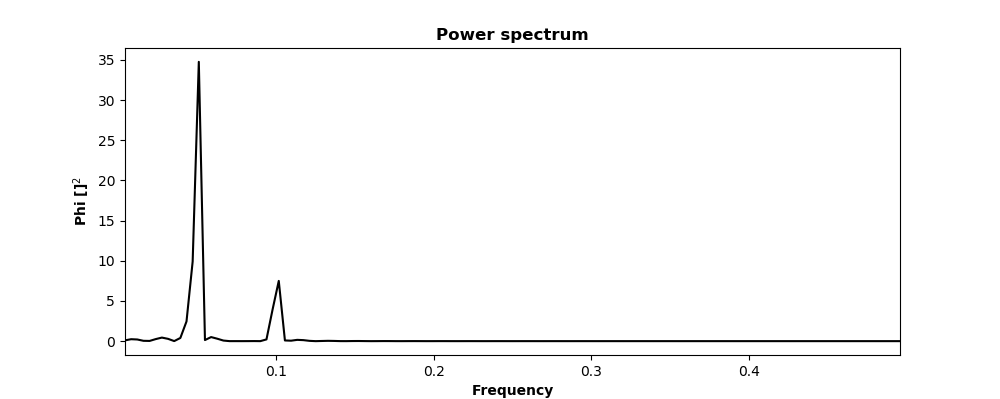


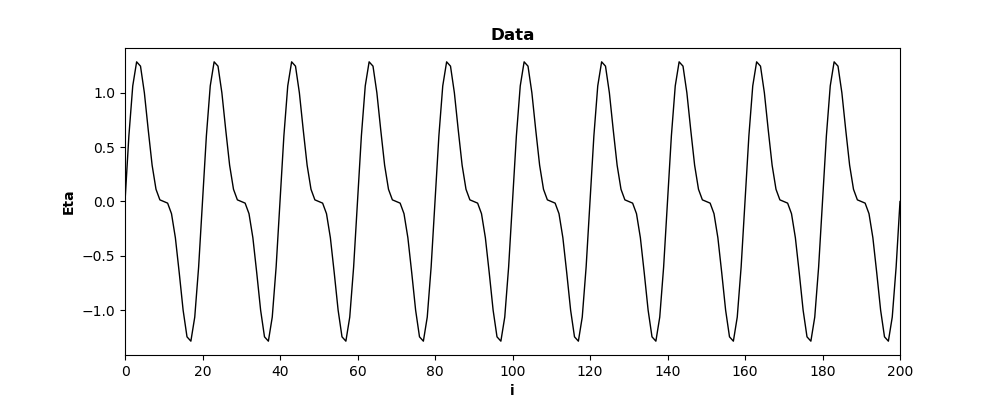
Phase는 wavelet transform의 실수, 허수 부분으로 계산한다. 마찬가지로 i가 100보다 작을 때는 20일 주기, 100보다 클 때는 10일 주기로 phase가 바뀌는 것을 볼 수 있다.



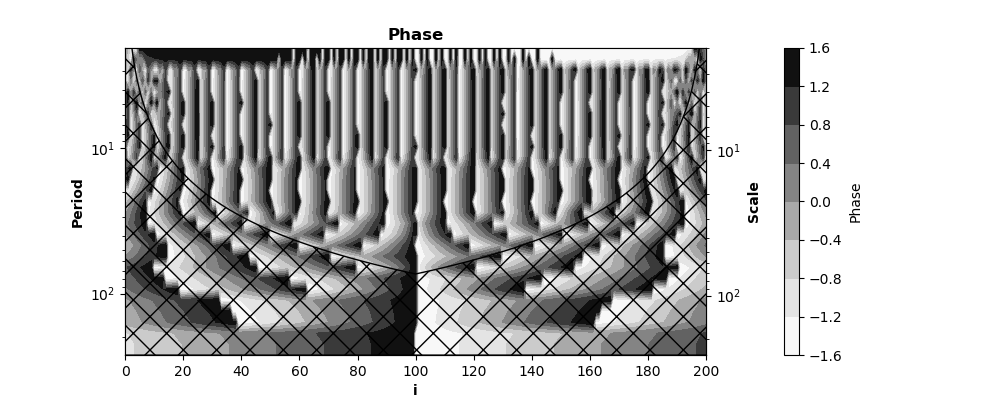
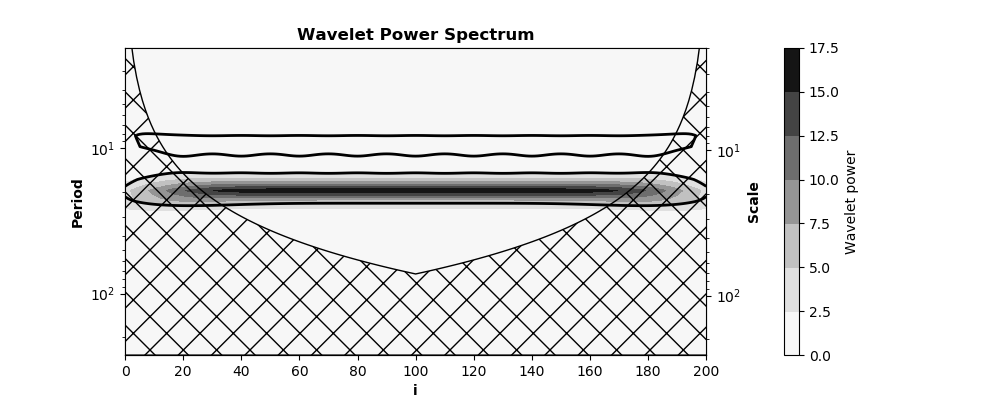
**2.**

주어진 함수는 주기 20, 10을 갖는 사인 함수가 다른 규모로 합쳐진 형태로 자료를 나타내면 아래와 같다. 20 간격으로 주기적으로 진동하는 형태를 갖는다. 1번 문제와 다르게 시간에 따라 파동의 형태가 바뀌지는 않는다. 이 경우 파워 스펙트럼으로 분석한 결과 진동수 0.05, 0.1 (주기 20, 10)에서 강한 에너지가 감지되며, 20 주기의 에너지가 더 강하게 나타난다. 이는 사인 함수의 진폭의 차이로부터 유도된 것으로 보인다. 여기서 주의할 점은 1번 문제와 함수의 형태가 다름에도 불구하고 파워 스펙트럼 분석 결과는 동일하게 나왔다는 점이다. 따라서 wavelet 분석을 통해 시간에 따라 변화하는 자료의 특성을 파악할 필요가 있다.





마찬가지로 Morlet wavelet 분석 결과, 문제 1번과는 다르게 자료의 모든 기간에서 20 주기의 강한 에너지가 나타나며, 파워 스펙트럼 분석과 동일하게 20 주기의 에너지가 더 강하게 나타난다. Phase 분석에서는 10, 20 주기로 phase가 바뀌는 것을 확인할 수 있다.



# -\*- coding: utf-8 -\*-

"""

Created on Thu May 16 14:24:20 2019

@author: WHITE

"""

#%% IMPORT MODULES

import numpy as np

import matplotlib.pyplot as plt

import pycwt as wavelet

#%% HW8

dt = 1

i = np.arange(0, 200 + dt, dt)

eta = np.zeros(len(i), dtype = float)

# HW8-1

for ii in i:

if ii >= 0 and ii <= 100:

eta[ii] = np.sin(2 \* np.pi \* ii / 20)

else:

eta[ii] = np.sin(2 \* np.pi \* ii / 10) / 2

# HW8-2

"""

for ii in i:

eta[ii] = np.sin(2 \* np.pi \* ii / 20) + np.sin(2 \* np.pi \* ii / 10) / 2

"""

fig, sub = plt.subplots(figsize = (10, 4))

sub.plot(i, eta, ls = "-", c = "k", lw = 1)

sub.set\_xlim(min(i), max(i))

sub.set\_xticks(np.arange(0, 200 + 20, 20))

sub.set\_xlabel("i", fontdict = {"weight": "bold"})

sub.set\_ylabel("Eta", fontdict = {"weight": "bold"})

sub.set\_title("Data", fontdict = {"weight": "bold"})

mother = wavelet.Morlet(f0 = 6)

alpha, \_, \_ = wavelet.ar1(eta)

dj = 0.25

s0 = 2 \* dt

J = 7 / dj

wave, scales, freqs, coi, fft, fftfreqs = wavelet.cwt(signal = eta, dt = dt,\

dj = dj, s0 = s0, J = J,\

wavelet = mother)

power = np.abs(wave) \*\* 2

fft\_power = np.abs(fft) \*\* 2

period = 1 / freqs

signif, fft\_theor = wavelet.significance(signal = 1.0, dt = dt,\

scales = scales, alpha = alpha,\

significance\_level = 0.95,\

wavelet = mother)

sig95 = np.ones([1, len(eta)]) \* signif[:, None]

sig95 = power / sig95

phase = np.zeros(np.shape(wave), dtype = float)

for jj in range(len(period)):

for ii in range(len(i)):

real = wave[jj, ii].real

imag = wave[jj, ii].imag

phase[jj, ii] = np.arctan(imag / real)

fig, sub = plt.subplots(figsize = (10, 4))

sub1 = sub.twinx()

contf = sub.contourf(i, period, power, cmap = "Greys", zorder = 7)

cont = sub.contour(i, period, sig95,\

[-99, 1], colors = "k", linewidths = 2, zorder = 9)

sub.set\_yscale("log")

sub1.set\_yscale("log")

sub.fill(np.concatenate([i, i[-1 :] + dt, i[-1 :] + dt,\

i[: 1] - dt, i[: 1] - dt]),\

np.concatenate([coi, [1e-9], period[-1 :],

period[-1 :], [1e-9]]),

"None", hatch = "x", edgecolor = "k", zorder = 11)

sub.set\_xlim(min(i), max(i))

sub.set\_ylim(min(period), max(period))

sub1.set\_ylim(min(scales), max(scales))

sub.invert\_yaxis()

sub1.invert\_yaxis()

sub.set\_xticks(np.arange(0, 200 + 20, 20))

sub.set\_xlabel("i", fontdict = {"weight": "bold"})

sub.set\_ylabel("Period", fontdict = {"weight": "bold"})

sub1.set\_ylabel("Scale", fontdict = {"weight": "bold"})

sub.set\_title("Wavelet Power Spectrum", fontdict = {"weight": "bold"})

fig.colorbar(contf, pad = 0.1, label = "Wavelet power")

fig, sub = plt.subplots(figsize = (10, 4))

sub1 = sub.twinx()

contf = sub.contourf(i, period, phase, cmap = "Greys", zorder = 7)

sub.set\_yscale("log")

sub1.set\_yscale("log")

sub.fill(np.concatenate([i, i[-1 :] + dt, i[-1 :] + dt,\

i[: 1] - dt, i[: 1] - dt]),\

np.concatenate([coi, [1e-9], period[-1 :],

period[-1 :], [1e-9]]),

"None", hatch = "x", edgecolor = "k", zorder = 11)

sub.set\_xlim(min(i), max(i))

sub.set\_ylim(min(period), max(period))

sub1.set\_ylim(min(scales), max(scales))

sub.invert\_yaxis()

sub1.invert\_yaxis()

sub.set\_xticks(np.arange(0, 200 + 20, 20))

sub.set\_xlabel("i", fontdict = {"weight": "bold"})

sub.set\_ylabel("Period", fontdict = {"weight": "bold"})

sub1.set\_ylabel("Scale", fontdict = {"weight": "bold"})

sub.set\_title("Phase", fontdict = {"weight": "bold"})

fig.colorbar(contf, pad = 0.1, label = "Phase")

fig, sub = plt.subplots(figsize = (10, 4))

sub.plot(fftfreqs, fft\_power, c = "k", ls = "-")

sub.set\_xlim(min(fftfreqs), max(fftfreqs))

sub.set\_xlabel("Frequency", fontdict = {"weight": "bold"})

sub.set\_ylabel(r"Phi []$^2$", fontdict = {"weight": "bold"})

sub.set\_title("Power spectrum", fontdict = {"weight": "bold"})