1. Find a combination of filters for removing the wide peak at 0 while retaining the narrow peaks at 0.3, 0.7 and 0.8 in the data generated by this script:

```
from numpy import *
import matplotlib as mpl
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
from matplotlib import colors
import pylab

def gaussian(x,x0,s):
     return exp(-(x-x0)**2/(2*s**2))

points=1000
x = linspace(-1,1,points)
ys = zeros(points)
ys+=gaussian(x,0,0.1)
ys+=0.75*gaussian(x,0.3,0.01)
ys+=1.2*gaussian(x,0.7,0.01)
ys+=0.9*gaussian(x,0.8,0.01)
yn=0.2*random.normal(size=len(x))
ysn=ys+yn

fig, (ax1) = plt.subplots(1,figsize=(6,6))
ax1.plot(x,ys,c='r')
ax1.set_ylim([1.2*min(ys),1.2*max(ys)])
ax1.set_xlim([min(x),max(x)])
fig.savefig('peaks',dpi=300,bbox_inches='tight')
fig.clf()

fig, (ax1) = plt.subplots(1,figsize=(6,6))
ax1.plot(x,yn,c='black')
ax1.set_ylim([1.2*min(ysn),1.2*max(ysn)])
ax1.set_xlim([min(x),max(x)])
fig.savefig('noise',dpi=300,bbox_inches='tight')
fig.clf()

fig, (ax1) = plt.subplots(1,figsize=(6,6))
ax1.plot(x,ysn,c='black')
ax1.set_ylim([1.2*min(ysn),1.2*max(ysn)])
ax1.set_xlim([min(x),max(x)])
fig.savefig('peaks_plus_noise',dpi=300,bbox_inches='tight')
fig.clf()
```