```
'''
Created on Oct 3, 2017

@author: bob
'''
import numpy as np
X=np.array([0,0,1,0,0,1,1,1]).reshape(4,2
)
#=========================================
=========================================
print(X)
#=========================================
=========================================
#=========================================
=========================================
# digress = np.arange(24)
# print(digress)
# print(np.shape(digress))
# #dig2=np.reshape(digress,(4,3,2))
# dig2=digress.reshape(4,3,2)
# print(dig2)
# print(dig2[1,2,1])
#=========================================
=========================================
#W= np.array([3,2]).reshape(2,1)
```

```
W= np.random.randn(2,1)#right wayfor
matrix mult
#W= np.random.randn(2) #wrong way but
gives an answer unfortunately
print (W.shape,W)

outInt= np.dot(X,W)
print("out without function\n",outInt)
b=1
Y=np.array([0,1,1,1]).reshape(4,1)
def cost(Wv,bv):
    costArray= ((np.dot(X,Wv)+bv)-Y)**2 #
taking square here to eventually take sum
of squares
    return costArray.sum()
def costWrapper(Wwrongv):
    b1=Wwrongv.pop()
    Wright1=
np.array(Wwrongv).reshape(2,1)
    return cost(Wright1,b1)


outInt1= cost(W,b)
print("\n\n out from function\n
",outInt1)
```

```
Wwrong= [3,2,5]
Wr=np.array([3,2]).reshape(2,1)
br=5
print("this is cost from
Wrapper",costWrapper(Wwrong))
print("ref call or value? ",Wwrong)
print("\n this is cost from function \n",
cost(Wr,br))
```

```
exit()
b=Wwrong.pop()
print(Wwrong,b)
#print(Wwrong.shape)
Wright= np.array(Wwrong).reshape(2,1)
print(Wright,Wright.shape)
TestW=  np.random.randn(2)
TestW= TestW.tolist()
print(TestW.shape,TestW)
```