1. A patient takes vitamin pills daily. Each day he must have at least 240 IU of vitamin A, 5 mg of vitamin B1, and 120 mg of vitamin C. He can choose between pill 1, which contains 120 IU of vitamin A, 1 mg of vitamin B1, and 15 mg of vitamin C; and pill 2, which contains 30 IU of vitamin A, 1 mg of vitamin B1, and 40 mg of vitamin C. Pill 1 costs $0.10, and pill 2 costs $0.20. **Using Python**, determine how many of each pill he should take daily in order to minimize cost and determine the minimum cost. Also, determine the amount of surplus of each type of vitamin, if any.

import numpy as np

import matplotlib.pyplot as plt

# 120x + 30y >= 240 (0,8), (2,0)

# x + y >= 5 (0,5), (5,0)

# 15x + 40y >= 120 (0,3), (8,0)

# .1x + .2y

A = np.array([[120,30],

[1 ,1 ],

[15 ,40]])

b = np.array([240,5,120])

zs = np.array([.1, .2])

# 120x + 30y >= 240

l1 = {'m':-120./30, 'b':240.0/30}

# x + y >= 5

l2 = {'m':-1, 'b':5}

# 15x + 40y >= 120

l3 = {'m':-15./40, 'b':120./40}

lines = [l1,l2,l3]

def get\_points(line):

p1 = (0, line['b'])

p2 = (float(-line['b'])/line['m'], 0)

return p1, p2

def intersection(line1, line2):

"""Finds the intersection of two lines.

The lines are dictionaries of the form l = {'m':2, 'b':6}

for a line y = 2x + 6"""

m1 = line1['m']

m2 = line2['m']

b1 = line1['b']

b2 = line2['b']

m1 -= m2

b2 -= b1

x = float(b2)/m1

y = line1['m']\*x + line1['b']

return x,y

def is\_feasable(A, x, b):

A = np.array(A)

x = np.array(x)

b = np.array(b)

return np.all(A.dot(x)>=b)

corner\_points = []

for line in lines[:-1]:

for line2 in lines[1:]:

if line != line2:

point = intersection(line, line2)

# our point point needs to made of integers

# search the integer points around this for feasable solutions.

xs = [np.floor(point[0]), np.ceil(point[0])]

ys = [np.floor(point[1]), np.ceil(point[1])]

for x in xs:

for y in ys:

if is\_feasable(A,np.array([x,y]), b):

corner\_points.append((x,y))

for l in lines:

p1, p2 = get\_points(l)

corner\_points.append(p1)

corner\_points.append(p2)

\_min = 10000

solution = {}

print '\n'

for point in corner\_points:

x = np.array(point)

if is\_feasable(A, x, b):

amounts = A.dot(x)

temp\_min = zs.dot(x)

# print 'point: {}'.format(point)

# print 'for amounts: {}'.format(amounts)

# print 'for a total cost of: {}'.format(temp\_min)

# print '\n'

if temp\_min < \_min:

\_min = temp\_min

solution['cost'] = temp\_min

solution['points'] = point

solution['surplus'] = amounts-b

print 'The optimal solution is {} units of pill 1 and {} units of pill 2.'.format(solution['points'][0],\

solution['points'][1] )

print 'The cost is minimized at {}'.format(solution['cost'])

print 'The surplus for each vitamin is: vitamin a {}, vitamin b {}, vitamin c {}'.format(solution['surplus'][0],\ solution['surplus'][1],\ solution['surplus'][2])

print '\n'\*3

OUTPUT:

The optimal solution is 3.0 units of pill 1 and 2.0 units of pill 2.

The cost is minimized at 0.7

The surplus for each vitamin is: vitamin a 180.0, vitamin b 0.0, vitamin c 5.0

1. Two stores sell a certain product. Store A has 38% of the sales, 4% of which are defective. Store B has 62% of the sales, 1% of which are defective. The difference in defective items is due to different levels of pre-sales checking of the product. A person receives a defective item of this product as a gift. What is the probability it came from store B?

NotA = StoreAPrecent \* StoreANotDefect = .38\*(1-.04)

A = StoreAPrecent \* StoreAtDefect = .38\*(.04)

NotB = StoreBPrecent \* StoreBDefect = .62\*(.01)

B = StoreBPrecent \* StoreBNotDefect = .62\*(1-.01)

P(B|d) = P(B ^ d)/P(d) = B/A+B = 0.0152/(.0062+.0152) = .7102 = 71.02%

1. The following is a graph of a third degree polynomial with leading coefficient 1. Determine the function depicted in the graph. **Using Python**, recreate the graph of the original function, (𝑥), as well as the graph of its derivative.

1. The average number of vehicles waiting in line to enter a parking ramp can be modeled by the function

(𝑥) = 𝑥2 /2(1 − 𝑥)

where x is a quantity between 0 and 1 known as the traffic intensity. Find the rate of change of the number of vehicles in line with respect to the traffic intensity for x = 0.3.

7. The population of Asian Carp in Lake Michigan is described by the logistic equation where G (t) is the population after t years. Find the point in time at which the growth rate of this population begins to decline. 12000/(1+19e^-1.2t)

My method to solve this problem was to find the second derivative of the function and set to equal to zero. The second derivative

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.0 | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 | 1.8 | 1.9 | 2.0 |
| 1785.0 | 1975.1 | 2181.2 | 2403.6 | 2642.5 | 2898.1 | 3170.0 | 3457.7 | 3760.4 | 4076.9 | 4405.9 |
|  | 190.1 | 206.0 | 222.4 | 239.0 | 255.5 | 271.9 | 287.7 | 302.7 | 316.6 | 328.9 |
|  |  | 15.9 | 16.3 | 16.6 | 16.6 | 16.3 | 15.8 | 15.0 | 13.9 | 12.4 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 2.7 | 2.8 | 2.9 | 3.0 | 3.1 | 3.2 |
| 4745.5 | 5093.7 | 5448.2 | 5806.7 | 6166.6 | 6525.3 | 6880.3 | 7229.0 | 7569.4 | 7899.2 | 8216.7 | 8520.4 |
| 339.6 | 348.2 | 354.6 | 358.5 | 359.9 | 358.7 | 354.9 | 348.8 | 340.3 | 329.8 | 317.5 | 303.8 |
| 10.6 | 8.6 | 6.4 | 3.9 | 1.4 | -1.2 | -3.7 | -6.2 | -8.5 | -10.5 | -12.3 | -13.8 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2.40 | 2.41 | 2.42 | 2.43 | 2.44 | 2.45 | 2.46 | 2.47 | 2.48 | 2.49 | 2.50 |
| 5806.750 | 5842.719 | 5878.700 | 5914.689 | 5950.684 | 5986.683 | 6022.683 | 6058.681 | 6094.675 | 6130.662 | 6166.640 |
| 35.955 | 35.969 | 35.980 | 35.989 | 35.995 | 35.999 | 36.000 | 35.998 | 35.994 | 35.987 | 35.978 |
| 0.016 | 0.014 | 0.011 | 0.009 | 0.006 | 0.004 | 0.001 | -0.002 | -0.004 | -0.007 | -0.009 |