**The code for the Q1.(a).**

import numpy as np

import math

from math import sqrt

def f(m): return (-0.2)\*math.pow(0.99,m)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)+f(13)+f(14)+f(15)+f(16)+f(17)+f(18)+f(19)+f(20)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q1.(a).**

-28.058508665372997

**The code for Q1.(b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-0.1+g(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(b) Middle Degree of seasonality**

79.08657961029601

**The code for Q1.(b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.727

    if m==2:

        return 0.381

    if m==3:

        return 1.303

    if m==4:

        return 1.280

    if m==5:

        return 1.188

    if m==6:

        return 1.119

    if m==7:

        return 1.073

    if m==8:

        return 1.073

    if m==9:

        return 1.073

    if m==10:

        return 1.004

    if m==11:

        return 0.935

    if m==12:

        return 0.843

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-0.1+g(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(b) High Degree of seasonality**

79.01796578459108

**The code for Q1.(b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.932

    if m==2:

        return 0.845

    if m==3:

        return 1.076

    if m==4:

        return 1.070

    if m==5:

        return 1.047

    if m==6:

        return 1.030

    if m==7:

        return 1.018

    if m==8:

        return 1.018

    if m==9:

        return 1.018

    if m==10:

        return 1.001

    if m==11:

        return 0.984

    if m==12:

        return 0.961

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-0.1+g(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(b) Low Degree of seasonality**

79.10665812922323

**Q1.(c).**

**If the seasonal consumption risk is considered, the welfare is high. If the nonseasonal consumption risk is considered, the welfare is low, under zero. So the seasonal consumption risk has significant influence to the welfare.**

**The code for Q1.(d) η=2 (a)**

import numpy as np

import math

from math import sqrt

def f(m): return (-1)\*math.pow(0.99,m-1)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for Q1.(d) η=2 (a)**

-88.4182993237999

**The code for Q1.(d) η=2 (b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-(math.exp(g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=2 (b) Middle Degree of seasonality**

-32.62872967157005

**The code for Q1.(d) η=2 (b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.727

    if m==2:

        return 0.381

    if m==3:

        return 1.303

    if m==4:

        return 1.280

    if m==5:

        return 1.188

    if m==6:

        return 1.119

    if m==7:

        return 1.073

    if m==8:

        return 1.073

    if m==9:

        return 1.073

    if m==10:

        return 1.004

    if m==11:

        return 0.935

    if m==12:

        return 0.843

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-(math.exp(g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=2 (b) High Degree of seasonality**

-33.51459481357202

**The code for Q1.(d) η=2 (b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.932

    if m==2:

        return 0.845

    if m==3:

        return 1.076

    if m==4:

        return 1.070

    if m==5:

        return 1.047

    if m==6:

        return 1.030

    if m==7:

        return 1.018

    if m==8:

        return 1.018

    if m==9:

        return 1.018

    if m==10:

        return 1.001

    if m==11:

        return 0.984

    if m==12:

        return 0.961

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-(math.exp(g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=2 (b) Low Degree of seasonality**

-32.42094583463844

**Discussion**

**If the seasonal consumption risk is considered, the welfare is high. If the nonseasonal consumption risk is considered, the welfare is low, under zero. So the seasonal consumption risk has significant influence to the welfare.**

**The code for Q1.(d) η=4 (a)**

import numpy as np

import math

from math import sqrt

def f(m): return (1/3)\*math.pow(0.99,m-1)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for Q1.(d) η=4 (a)**

29.47276644126663

**The code for Q1.(d) η=4 (b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-1/3\*(math.exp(-3\*g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=4 (b) Middle Degree of seasonality**

-1.583626388269792

**The code for Q1.(d) η=4 (b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.727

    if m==2:

        return 0.381

    if m==3:

        return 1.303

    if m==4:

        return 1.280

    if m==5:

        return 1.188

    if m==6:

        return 1.119

    if m==7:

        return 1.073

    if m==8:

        return 1.073

    if m==9:

        return 1.073

    if m==10:

        return 1.004

    if m==11:

        return 0.935

    if m==12:

        return 0.843

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-1/3\*(math.exp(-3\*g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=4 (b) High Degree of seasonality**

-2.0955838861054845

**The code for Q1.(d) η=4 (b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.932

    if m==2:

        return 0.845

    if m==3:

        return 1.076

    if m==4:

        return 1.070

    if m==5:

        return 1.047

    if m==6:

        return 1.030

    if m==7:

        return 1.018

    if m==8:

        return 1.018

    if m==9:

        return 1.018

    if m==10:

        return 1.001

    if m==11:

        return 0.984

    if m==12:

        return 0.961

print (g(2))

def f(m): return math.pow(0.99,m-1)\*(-1/3\*(math.exp(-3\*g(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.(d) η=4 (b) Low Degree of seasonality**

-1.4882861634725864

**Discussion**

**This condition is opposite with the previous two. If the seasonal consumption risk is considered, the welfare is low. If the nonseasonal consumption risk is considered, the welfare is high. So the nonseasonal consumption risk has significant influence to the welfare.**

**The code for the Q1.2.(a).**

import numpy as np

import math

from math import sqrt

def f(m): return (-0.2)\*math.pow(0.99,m)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)+f(13)+f(14)+f(15)+f(16)+f(17)+f(18)+f(19)+f(20)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q1.2.(a).**

-28.058508665372997

**The code for Q1.2.(b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.085

    if m==2:

        return 0.068

    if m==3:

        return 0.290

    if m==4:

        return 0.283

    if m==5:

        return 0.273

    if m==6:

        return 0.273

    if m==7:

        return 0.239

    if m==8:

        return 0.205

    if m==9:

        return 0.188

    if m==10:

        return 0.188

    if m==11:

        return 0.171

    if m==12:

        return 0.137

print (a(1))

def f(m): return -0.1+g(m)-1/2\*(a(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(b) Middle Degree of seasonality**

74.30186924698219

**The code for Q1.2.(b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.171

    if m==2:

        return 0.137

    if m==3:

        return 0.580

    if m==4:

        return 0.567

    if m==5:

        return 0.546

    if m==6:

        return 0.546

    if m==7:

        return 0.478

    if m==8:

        return 0.410

    if m==9:

        return 0.376

    if m==10:

        return 0.376

    if m==11:

        return 0.341

    if m==12:

        return 0.273

print (a(1))

def f(m): return -0.1+g(m)-1/2\*(a(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(b) High Degree of seasonality**

65.0112334766455

**The code for Q1.2.(b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.043

    if m==2:

        return 0.034

    if m==3:

        return 0.145

    if m==4:

        return 0.142

    if m==5:

        return 0.137

    if m==6:

        return 0.137

    if m==7:

        return 0.119

    if m==8:

        return 0.102

    if m==9:

        return 0.094

    if m==10:

        return 0.094

    if m==11:

        return 0.085

    if m==12:

        return 0.068

print (a(1))

def f(m): return -0.1+g(m)-1/2\*(a(m))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(b) Low Degree of seasonality**

78.94525238917458

**Q1.2.(c).**

**If the seasonal consumption risk is considered, the welfare is high. If the nonseasonal consumption risk is considered, the welfare is low, under zero. So the seasonal consumption risk has significant influence to the welfare.**

**The code for Q1.2.(d) η=2 (a)**

import numpy as np

import math

from math import sqrt

def f(m): return (-1)\*math.pow(0.99,m-1)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for Q1.2.(d) η=2 (a)**

-88.4182993237999

**The code for Q1.2.(d) η=2 (b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.085

    if m==2:

        return 0.068

    if m==3:

        return 0.290

    if m==4:

        return 0.283

    if m==5:

        return 0.273

    if m==6:

        return 0.273

    if m==7:

        return 0.239

    if m==8:

        return 0.205

    if m==9:

        return 0.188

    if m==10:

        return 0.188

    if m==11:

        return 0.171

    if m==12:

        return 0.137

print (a(1))

def f(m): return -(math.exp(-(g(m))+a(m)))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=2 (b) Middle Degree of seasonality**

-41.802950904320994

**The code for Q1.2.(d) η=2 (b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.171

    if m==2:

        return 0.137

    if m==3:

        return 0.580

    if m==4:

        return 0.567

    if m==5:

        return 0.546

    if m==6:

        return 0.546

    if m==7:

        return 0.478

    if m==8:

        return 0.410

    if m==9:

        return 0.376

    if m==10:

        return 0.376

    if m==11:

        return 0.341

    if m==12:

        return 0.273

print (a(1))

def f(m): return -(math.exp(-(g(m))+a(m)))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=2 (b) High Degree of seasonality**

-51.03489785147121

**The code for Q1.2.(d) η=2 (b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.043

    if m==2:

        return 0.034

    if m==3:

        return 0.145

    if m==4:

        return 0.142

    if m==5:

        return 0.137

    if m==6:

        return 0.137

    if m==7:

        return 0.119

    if m==8:

        return 0.102

    if m==9:

        return 0.094

    if m==10:

        return 0.094

    if m==11:

        return 0.085

    if m==12:

        return 0.068

print (a(1))

def f(m): return -(math.exp(-(g(m))+a(m)))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=2 (b) Low Degree of seasonality**

-37.912176349972796

**Discussion**

**If the seasonal consumption risk is considered, the welfare is high. If the nonseasonal consumption risk is considered, the welfare is low, under zero. So the seasonal consumption risk has significant influence to the welfare.**

**The code for Q1.(d) η=4 (a)**

import numpy as np

import math

from math import sqrt

def f(m): return (1/3)\*math.pow(0.99,m-1)

def g(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print(g(f))

def h(t): return math.pow(0.99,12\*t)\*(g(f))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for Q1.(d) η=4 (a)**

29.47276644126663

**The code for Q1.2.(d) η=4 (b) Middle Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.085

    if m==2:

        return 0.068

    if m==3:

        return 0.290

    if m==4:

        return 0.283

    if m==5:

        return 0.273

    if m==6:

        return 0.273

    if m==7:

        return 0.239

    if m==8:

        return 0.205

    if m==9:

        return 0.188

    if m==10:

        return 0.188

    if m==11:

        return 0.171

    if m==12:

        return 0.137

print (a(1))

def f(m): return -(math.exp(-3\*(g(m))-1/2\*(a(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=4 (b) Middle Degree of seasonality**

-4.589108434831149

**The code for Q1.2.(d) η=4 (b) High Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.171

    if m==2:

        return 0.137

    if m==3:

        return 0.580

    if m==4:

        return 0.567

    if m==5:

        return 0.546

    if m==6:

        return 0.546

    if m==7:

        return 0.478

    if m==8:

        return 0.410

    if m==9:

        return 0.376

    if m==10:

        return 0.376

    if m==11:

        return 0.341

    if m==12:

        return 0.273

print (a(1))

def f(m): return -(math.exp(-3\*(g(m))-1/2\*(a(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=4 (b) High Degree of seasonality**

-4.221956879879851

**The code for Q1.2.(d) η=4 (b) Low Degree of seasonality**

import numpy as np

import math

from math import sqrt

def g(m):

    if m==1:

        return 0.863

    if m==2:

        return 0.691

    if m==3:

        return 1.151

    if m==4:

        return 1.140

    if m==5:

        return 1.094

    if m==6:

        return 1.060

    if m==7:

        return 1.037

    if m==8:

        return 1.037

    if m==9:

        return 1.037

    if m==10:

        return 1.002

    if m==11:

        return 0.968

    if m==12:

        return 0.921

print (g(2))

def a(m):

    if m==1:

        return 0.043

    if m==2:

        return 0.034

    if m==3:

        return 0.145

    if m==4:

        return 0.142

    if m==5:

        return 0.137

    if m==6:

        return 0.137

    if m==7:

        return 0.119

    if m==8:

        return 0.102

    if m==9:

        return 0.094

    if m==10:

        return 0.094

    if m==11:

        return 0.085

    if m==12:

        return 0.068

print (a(1))

def f(m): return -(math.exp(-3\*(g(m))-1/2\*(a(m))))

print (f(2))

def h(f): return f(1)+f(2)+f(3)+f(4)+f(5)+f(6)+f(7)+f(8)+f(9)+f(10)+f(11)+f(12)

print (h(f))

def l(t): return math.pow(0.99,12\*t)\*(h(f))

print (l(1))

def w(l): return l(1)+l(2)+l(3)+l(4)+l(5)+l(6)+l(7)+l(8)+l(9)+l(10)+l(11)+l(12)+l(13)+l(14)+l(15)+l(16)+l(17)+l(18)+l(19)+l(20)+l(21)+l(22)+l(23)+l(24)+l(25)+l(26)+l(27)+l(28)+l(29)+l(30)+l(31)+l(32)+l(33)+l(34)+l(35)+l(36)+l(37)+l(38)+l(39)+l(40)

print (w(l))

**The result for Q1.2.(d) η=4 (b) Low Degree of seasonality**

-4.787010092026615

**Discussion**

**If the seasonal consumption risk is considered, the welfare is high. If the nonseasonal consumption risk is considered, the welfare is low, under zero. So the seasonal consumption risk has significant influence to the welfare.**

**The code for the Q2.(a). Isolating the effect of  consumption**

import numpy as np

import math

from math import sqrt

def h(t): return math.pow(0.99,12\*t)\*(-0.1\*12)

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q2.(a). Isolating the effect of  consumption**

-9.338717545031592

**The code for the Q2.(a). Isolating the effect of  leisure**

import numpy as np

import math

from math import sqrt

def h(t): return math.pow(0.99,12\*t)\*(12\*(-28.5\*30/7\*1/2\*math.exp(2)))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q2.(a). Isolating the effect of  leisure**

-42141.91656901566

**The code for the Q2.(b). Isolating the effect of  consumption**

import numpy as np

import math

from math import sqrt

def h(t): return math.pow(0.99,12\*t)\*(-0.1\*12)

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q2.(b). Isolating the effect of  consumption**

-9.338717545031592

**The code for the Q2.(b). Isolating the effect of  leisure**

import numpy as np

import math

from math import sqrt

def h(t): return math.pow(0.99,12\*t)\*(12\*(-28.5\*30/7\*1/2\*math.exp(2)))

def w(h):return h(1)+h(2)+h(3)+h(4)+h(5)+h(6)+h(7)+h(8)+h(8)+h(10)+h(11)+h(12)+h(13)+h(14)+h(15)+h(16)+h(17)+h(18)+h(19)+h(20)+h(21)+h(22)+h(23)+h(24)+h(25)+h(26)+h(27)+h(28)+h(29)+h(30)+h(31)+h(32)+h(33)+h(34)+h(35)+h(36)+h(37)+h(38)+h(39)+h(40)

print(w(h))

**The result for the Q2.(b). Isolating the effect of  leisure**

-42141.91656901566

**The result for the Q2.(c). Isolating the effect of  consumption**

-9.338717545031592

**The result for the Q2.(c). Isolating the effect of  leisure**

-42141.91656901566