“Embedded Value”

import pandas as pd

import numpy as np

import xlrd

import xlsxwriter

import time

from datetime import datetime

start\_time = time.time()

valuation\_date = pd.to\_datetime('31-12-2018')

print(' ')

print('Reading Data...')

print(' ')

evdata = pd.read\_csv("ev\_data.csv")

wakala\_b1 = pd.read\_csv("wakala\_bonus.csv")

inv\_b2 = pd.read\_csv("inv\_bonus.csv")

annuity\_table = pd.read\_csv("AnnuityFactorPlan1.csv", index\_col='Rem\_Term')

bonus\_1 = pd.read\_csv("bonus1.csv", index\_col = 'Year')

path = "Embedded\_Value.xlsx"

writer = pd.ExcelWriter(path, engine='xlsxwriter')

print(' ')

print('Data Fetched..')

print(' ')

mortality = pd.read\_csv("Mortality\_Rates.csv", index\_col='Age')

alloc\_table = pd.read\_csv("Allocation\_Table.csv", index\_col='Year')

def main():

#Other Assumptions

Past\_Unit\_Growth\_Rate\_Assumption = 0.07

Annual\_Unit\_Growth\_Rate = 0.12

Annual\_Investment\_Charge = 0.015

Mortality\_RI\_Rates = 0.8

Past\_PTF\_ROR\_Assumption = 0.30

PTF\_GROWTH\_RATE = 0.12

bid\_offer\_charge = 0.05

annual\_imc = 0.015

NET\_CV = []

INV\_INCOME = []

I\_M\_C = []

END\_FUND = []

ACCU\_Year = []

POL\_ID = []

Bid\_Offer = []

ALLOCATED\_PREMIUM = []

START\_FUND = []

PREMIUM = []

MORT\_RATE = []

MORT\_CHARGE = []

MORT\_CHARGE\_PLAN2 = []

prdinfo = pd.read\_csv("D:/Python/PyCharm Projects/EmbeddedValue/prodinfo.csv")

prod\_info = pd.DataFrame(prdinfo)

print(' ')

print('Calculating Embedded Value...')

print(' ')

for index, row in evdata.head(10).iterrows():

print(index,row)

n = 5

product = row['PRDUCT\_NAME']

prod\_code = row['PRODUCT\_CODE']

gender = row['GENDER']

mop = row['POL\_MOP']

pay\_term = row['PAYING\_TERM']

pol\_term = row['POL\_TERM']

pol\_fee = row['POL\_FEE']

ea = row['Entry Age']

sa = row['SA']

ptf\_rate = row['PTF\_RATE']

issue\_date = pd.to\_datetime(row['ISSUANCE\_DT'])

maturity\_age = ea + pol\_term

pol\_row = row['POL\_NO']

basic\_cont = row['BASIC\_CONT']

cov\_multiple = sa/basic\_cont

trans\_date = row['TRANS\_DATE']

dob = row['DOB']

ptf\_amount = row['PTF\_AMT']

total\_cont = row['TOTAL\_CONT']

pol\_status = row['POL\_STATUS']

prem\_paid\_year = row['POL\_PREM\_PAID\_YEAR']

risk\_paid\_year = row['POL\_RISK\_PAID\_YEAR']

cust\_dob = row['CUST\_DOB']

cont\_index = row['Cont\_Indexation']

sob = row['SOB']

total = 0.0

year = 2018

mop\_dict = {'Year': ['Y', 'H', 'Q', 'M'], 'Year\_Value': [1, 2, 4, 12]}

date\_issue = round((issue\_date - valuation\_date).days / 365.25)

mop\_df = pd.DataFrame(mop\_dict, index=list('YHQM'))

bonus\_table = {'Year':['5','6','11','15'], 'bonus\_1':[0,0,0.05,0.05]}

bonus\_table1 = pd.DataFrame(bonus\_table)

p\_code = prod\_code

p\_code = str(p\_code)

main\_cond = prod\_info[(prod\_info.Product\_Code == p\_code) & (prod\_info.SOB\_From.isin([sob]))]

'Policy Fee Deduction'

certfeededtype = main\_cond.iloc[0,9]

cert\_fee = pol\_fee

certificate\_fee = np.where(certfeededtype == 'Internal',0,cert\_fee \* mop\_df.loc[mop,'Year\_Value'])

'Mortality Charge Deduction'

mort\_charge\_type = main\_cond.iloc[0, 13]

in\_built\_rider\_rate = int(main\_cond.iloc[0, 16])

in\_built\_rider\_SAR = int(0)

cert\_annual\_index\_rate = main\_cond.iloc[0, 17]

loading\_on\_retakaful = float(main\_cond.iloc[0, 18])

date\_diff = int(issue\_date.year - valuation\_date.year)

product\_type = main\_cond.iloc[0,7]

'Bonus Percentages of PIA'

bonus\_pia10 = main\_cond.iloc[0, 35]

bonus\_pia15 = main\_cond.iloc[0, 36]

bonus\_pia20 = main\_cond.iloc[0, 37]

bonus\_pia25 = main\_cond.iloc[0, 38]

bonus\_pia30 = main\_cond.iloc[0, 39]

bonus\_pia35 = main\_cond.iloc[0, 40]

bonus\_pia40 = main\_cond.iloc[0, 41]

'Commission'

commission1 = main\_cond.iloc[0, 42]

commission2 = main\_cond.iloc[0, 43]

commission3 = main\_cond.iloc[0, 44]

commission5 = main\_cond.iloc[0, 45]

start\_fund = 0

pia\_year = 0

start\_inv\_income = 0

start\_net\_cv = 0

for i in range(n):

check\_pay\_term = 1

age = ea

pif\_ror = (np.where(date\_diff<0,Past\_Unit\_Growth\_Rate\_Assumption,Annual\_Unit\_Growth\_Rate))

ptf\_ror = (np.where(date\_diff < 0, Past\_PTF\_ROR\_Assumption, PTF\_GROWTH\_RATE))

gender\_cond = str(np.where(gender == "MALE",'Male','Female'))

mort\_rate = np.where(mort\_charge\_type == 'Level Rates', ptf\_rate,np.where(mort\_charge\_type == 'Attained Age',((mortality.loc[age, gender\_cond]) \* (1 + loading\_on\_retakaful)), 0))

alloc\_cond = int(np.where(pia\_year<=6,pia\_year,6))

prod = str(product)

mortality\_charge = np.where(mort\_charge\_deduct == 'Internal',0, ((sa\*mort\_rate/1000)+(in\_built\_rider\_SAR\*in\_built\_rider\_rate/1000))\*check\_pay\_term)

allocated\_premium = (basic\_cont - mortality\_charge - certificate\_fee) \* allocation

mortality\_charge = ((sa\*mort\_rate)/1000)

mop\_cond = mop\_df.loc[mop,'Year\_Value']

certificate\_fee = np.where(certfeededtype == 'Internal',0,(cert\_fee\*(mop\_cond)))

bonus\_two = int(pia\_year+1)

bonus\_two\_table = inv\_b2.loc[bonus\_two+1, 'Investment']

bonus\_one\_ap = allocated\_premium \* bonus\_one\_table

bonus\_two\_pia = start\_fund\*bonus\_two\_table

distributed\_surplus = np.where((pia\_year-1)==(pay\_term),0,0)

bid\_offer = (bid\_offer\_charge \* (allocated\_premium+bonus\_one\_ap+bonus\_two\_pia+distributed\_surplus))

pol\_term\_cond = int(np.where((pol\_term - pia\_year)>40,40,np.where((pol\_term - pia\_year)<=0,0,(pol\_term - pia\_year))))

cov\_term\_cond = annuity\_table.loc[pol\_term\_cond,'Annuity\_factor']

#Basic Sum At Risk

basic\_life\_cond = np.where((bas\_life\_ben) == "SC", sa,

np.where((bas\_life\_ben) == "SC - CV", (sa - (in\_built\_rider\_SAR) - (start\_fund) - (allocated\_premium) \* 0.95),

np.where((bas\_life\_ben)=="Annuity x 2",cov\_term\_cond \* allocated\_premium \* 2,

np.where((bas\_life\_ben)== "Annuity x 1",cov\_term\_cond \* allocated\_premium,0))))

'InBuilt Rider Sum At Risk'

inbuilt\_rider\_SAR\_cond = max([np.where(age>in\_built\_rider\_maturity\_age,0,

np.where(in\_built\_rider=='-Plan2Old',basic\_cont\*5,

np.where(in\_built\_rider=='-Plan2',sa,

np.where(in\_built\_rider=='-Plan1',basic\_cont\*pay\_term,

np.where(in\_built\_rider=='-FIB',cov\_term\_cond\*basic\_cont,0)))))], default=0)

mort\_charge\_deduct\_Plan2 = np.where(mort\_charge\_deduct=="External",(np.where(product == 'PLAN2',((basic\_life\_cond\*mort\_rate/1000)+(in\_built\_rider\_rate\*inbuilt\_rider\_SAR\_cond/1000)),0)),((basic\_life\_cond\*mort\_rate/1000)+(in\_built\_rider\_rate\*inbuilt\_rider\_SAR\_cond/1000)))

# print(' ')

# print('start fund',start\_fund)

# print('allocated premium',allocated\_premium)

# print('bonus one ap', bonus\_one\_ap)

# print('bonus two pia', bonus\_two\_pia)

# print('distributed surplus', distributed\_surplus)

# print('bid offer', bid\_offer)

# print('mort charge Plan2', mort\_charge\_deduct\_Plan2)

# print(' ')

net\_cv = ((start\_fund)+(allocated\_premium)+(bonus\_one\_ap)+(bonus\_two\_pia)+(distributed\_surplus)-(bid\_offer)-(mort\_charge\_deduct\_Plan2))

IMC = (net\_cv+inv\_income+start\_net\_cv+start\_inv\_income)\*0.5\*annual\_imc

start\_inv\_income = (inv\_income)

end\_fund = (inv\_income+net\_cv-IMC)

ACCU\_Year.append(year)

NET\_CV.append(net\_cv)

INV\_INCOME.append(inv\_income)

I\_M\_C.append(IMC)

END\_FUND.append(end\_fund)

POL\_ID.append(pol\_row)

Bid\_Offer.append(bid\_offer)

ALLOCATED\_PREMIUM.append(allocated\_premium)

START\_FUND.append(start\_fund)

PREMIUM.append(basic\_cont)

MORT\_RATE.append(mort\_rate)

MORT\_CHARGE.append(mortality\_charge)

MORT\_CHARGE\_PLAN2.append(mort\_charge\_deduct\_Plan2)

ea += 1

pia\_year += 1

start\_fund = end\_fund

year += 1

date\_diff += 1

start\_net\_cv = net\_cv

# print('date difference is', date\_diff)

print(' ')

print('Updating Results')

print(' ')

main()

print(' ')

print('Completed...')

print(' ')

print("--- Embedded Value Completed in %s seconds ---" % (time.time() - start\_time))