Source code in Mathematica

1. Standard method of cash-flow calculation (Policy-by-policy)

In[1]:=

```
PVofCF[typ_, pocdat_, vstvek_, pohl_, n_, valDate_,
scen_, PC_, Prem_, f0_, Count_] :=
Module[{polYear, age, polYearVect, periodnew,
        MortExpVect, q, qexp, LapseVect, invIncome,
        discfact, riskpr, AlphaVect, BetaVect,
        GammaVect, konst, distrInvIncome, rate,
        SurrPaid, CommVector, ExpensesVector,
        inforceBoY, inforceEoY, numDeath, numLapses,
        numLapsesPom, numMatur, FondBoYReal,
        PremiumReal, AlphaReal, BetaReal,
        GammaReal, PCReal, riskprReal,
        distrInvIncomeReal, FondEoYReal, SurrPaidReal,
        DeathPaidReal, MaturPaidReal, CommReal,
        ExpensesReal, CFvector, CFdisc, fundBoY,
        fundEoY, discrate, PremPom, PremVector},
polYear = QuantityMagnitude[DateDifference[pocdat,
valDate, "Year"]] + 1;
age = vstvek + QuantityMagnitude[DateDifference[pocdat,
valDate, "Year"]];
polYearVect = Table[polYear + i, {i, 0, n - polYear}];
periodnew = n - polYear + 1;
MortExpVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] < Length[mortexp[[typ]]],
MortExpVect[[i]] =
mortexp[[typ,polYearVect[[i]]]],
MortExpVect[[i]] = Last[mortexp[[typ]]]]];
If[pohl == 0, q = Table[qmale[[i]], {i, age + 1,
        age + periodnew}],
q = Table[qfemale[[i]], {i, age + 1,
        age + periodnew}]];
qexp = q*MortExpVect;
LapseVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] < Length[lapsevect[[typ]]],
LapseVect[[i]] =
lapsevect[[typ,polYearVect[[i]]]],
LapseVect[[i]] = Last[lapsevect[[typ]]]]];
invIncome = Scenarios[[scen]];
discrate = DiscountScenarios[[scen]];
discfact = Table[Product[1/(1 + discrate[[i]]),
{i, 1, k}], {k, 1, periodnew}];
PremPom = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[typ == 2,
```

```
If[polYearVect[[i]] == 1,
PremPom[[i]] = 1,
PremPom[[i]] = 0],
PremPom[[i]] = 1]];
PremVector = Prem*PremPom;
riskpr = Table[(PC*q[[i]])/(1 + TIR[[typ]]),
{i, 1, Length[q]}];
AlphaVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == 1,
AlphaVect[[i]] = AlphaP[[typ]]*Prem +
AlphaSA[[typ]]*PC, AlphaVect[[i]] = 0]];
BetaVect = ConstantArray[Beta[[typ]]*PC, periodnew];
GammaVect = ConstantArray[Gamma[[typ]]*Prem, periodnew];
konst = PremVector - AlphaVect - BetaVect - GammaVect -
riskpr;
rate = Map[Max[TIR[[typ]], #] &, invIncome -
InvestmentMargin];
fundBoY = Join[{f0}, ConstantArray[{}, periodnew - 1]];
For[i = 1, i <= periodnew - 1, i++,
fundBoY[[i + 1]] = (fundBoY[[i]] + konst[[i]])*
(rate[[i]] + 1)];
fundEoY = Join[Rest[fundBoY], {(fundBoY[[periodnew]] +
        konst[[periodnew]])*(rate[[periodnew]] + 1)}];
distrInvIncome = (fundBoY + konst)*Table[rate[[i]],
{i, 1, periodnew}];
SurrPaid = (1 - SurrFee[[typ]])*fundEoY;
CommVector = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == 1,
CommVector[[i]] = InitCommP[[typ]]*Prem +
InitCommSA[[typ]]*PC,
CommVector[[i]] = RenCommP[[typ]]*Prem +
RenCommSA[[typ]]*PC]];
ExpensesVector = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == 1,
ExpensesVector[[i]] = InitExpFix[[typ]] +
InitExpP[[typ]]*Prem +
RenExpFix[[typ]]*
(1 + InflRateFixExp)^(i - 1) +
RenExpP[[typ]]*Prem,
ExpensesVector[[i]] = RenExpFix[[typ]]*
(1 + InflRateFixExp)^(i - 1) +
RenExpP[[typ]]*Prem]];
inforceBoY = Join[{Count}, ConstantArray[{},
periodnew - 1]];
For[i = 1, i <= periodnew - 1, i++,
```

```
inforceBoY[[i + 1]] = inforceBoY[[i]]*
(1 - qexp[[i]])*
(1 - If[polYearVect[[i]] > 2,
LapseVect[[i]], 0])];
numDeath = inforceBoY*qexp;
numLapsesPom = (inforceBoY - numDeath)*LapseVect;
numLapses = numLapsesPom;
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] > 2, numLapses[[i]] =
numLapsesPom[[i]], numLapses[[i]] = 0]];
numMatur = ConstantArray[0, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == n,
numMatur[[i]] = inforceBoY[[i]] -
numDeath[[i]] - numLapses[[i]],
numMatur[[i]] = 0]];
inforceEoY = inforceBoY - numDeath -
numLapses - numMatur;
FondBoYReal = inforceBoY*fundBoY;
PremiumReal = PremVector*inforceBoY;
AlphaReal = AlphaVect*inforceBoY;
BetaReal = BetaVect*inforceBoY;
GammaReal = GammaVect*inforceBoY;
PCReal = PC*inforceBoY;
riskprReal = riskpr*inforceBoY;
distrInvIncomeReal = distrInvIncome*inforceBoY;
FondEoYReal = fundEoY*inforceEoY;
SurrPaidReal = SurrPaid*numLapses;
DeathPaidReal = (fundEoY + PC)*numDeath;
MaturPaidReal = (fundEoY + PC)*numMatur;
CommReal = CommVector*inforceBoY;
ExpensesReal = ExpensesVector*inforceBoY;
CFvector = PremiumReal - SurrPaidReal - DeathPaidReal -
MaturPaidReal - CommReal - ExpensesReal;
CFdisc = CFvector*discfact;
Total[CFdisc]]
```

In[2]:=
```
cftable = ConstantArray[0, {Length[PolicyType],
        Length[Scenarios]}]
```

In[3]:=
```
cfallscenallMP = ConstantArray[0, Length[Scenarios]]
```

In[4]:=
```
For[m = 1, m <= Length[PolicyType], m++,
For[s = 1, s <= Length[Scenarios], s++,
cftable[[m,s]] = PVofCF[PolicyType[[m]],
IncDates[[m]], EntryAge[[m]], Sex[[m]],
PolicyPeriod[[m]], ValuationDate, s,
SumAssured[[m]], Premiums[[m]],
CVatValDate[[m]], CountPol[[m]]]]]
```

In[5]:=
```
For[w = 1, w <= Length[Scenarios], w++,
cfallscenallMP[[w]] =
Total[Transpose[cftable][[w]]]]
```

2. Analytic function for cash calculation

In[6]:=
```
fcepro1MP[typ_, pocdat_, vstvek_, pohl_, n_, valDate_,
PC_, Prem_, f0_, Count_] :=
Module[{polYear, age, polYearVect, periodnew,
        MortExpVect, q, qexp, LapseVect, PremPom, PremVector
        riskpr, AlphaVect, BetaVect, GammaVect, inforceBoY,
        numDeath, numLapsesPom, numLapses, numMatur, inforce
        CommVector, ExpensesVector, PremiumReal, CommReal,
        ExpensesReal, NetPremium, PayoutSA, Decrems, PayoutC
        SavPrem, PayoutSP, NetPremEq, PayoutSAEq, PayoutCVEq
        PayoutSPEq},
polYear = QuantityMagnitude[DateDifference[pocdat,
valDate, "Year"]] + 1;
age = vstvek + QuantityMagnitude[DateDifference[pocdat,
valDate, "Year"]];
polYearVect = Table[polYear + i, {i, 0, n - polYear}];
periodnew = n - polYear + 1;
MortExpVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] < Length[mortexp[[typ]]],
MortExpVect[[i]] =
mortexp[[typ,polYearVect[[i]]]],
MortExpVect[[i]] = Last[mortexp[[typ]]]]];
If[pohl == 0, q = Table[qmale[[i]],
{i, age + 1, age + periodnew}],
q = Table[qfemale[[i]],
{i, age + 1, age + periodnew}]];
qexp = q*MortExpVect;
LapseVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] < Length[lapsevect[[typ]]],
LapseVect[[i]] =
lapsevect[[typ,polYearVect[[i]]]],
LapseVect[[i]] = Last[lapsevect[[typ]]]]];
PremPom = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[typ == 2, If[polYearVect[[i]] == 1,
PremPom[[i]] = 1, PremPom[[i]] = 0],
PremPom[[i]] = 1]];
PremVector = Prem*PremPom;
riskpr = Table[(PC*q[[i]])/(1 + TIR[[typ]]),
{i, 1, Length[q]}];
AlphaVect = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
```

```
If[polYearVect[[i]] == 1,
AlphaVect[[i]] = AlphaP[[typ]]*Prem +
AlphaSA[[typ]]*PC, AlphaVect[[i]] = 0]];
BetaVect = ConstantArray[Beta[[typ]]*PC, periodnew];
GammaVect = ConstantArray[Gamma[[typ]]*Prem, periodnew];
inforceBoY = Join[{Count}, ConstantArray[{},
periodnew - 1]];
For[i = 1, i <= periodnew - 1, i++,
inforceBoY[[i + 1]] = inforceBoY[[i]]*
(1 - qexp[[i]])*(1 - If[polYearVect[[i]] > 2,
LapseVect[[i]], 0])];
numDeath = inforceBoY*qexp;
numLapsesPom = (inforceBoY - numDeath)*LapseVect;
numLapses = numLapsesPom;
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] > 2,
numLapses[[i]] = numLapsesPom[[i]],
numLapses[[i]] = 0]];
numMatur = ConstantArray[0, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == n,
numMatur[[i]] = inforceBoY[[i]] -
numDeath[[i]] -
numLapses[[i]],
numMatur[[i]] = 0]];
inforceEoY = inforceBoY - numDeath -
numLapses - numMatur;
CommVector = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == 1,
CommVector[[i]] = InitCommP[[typ]]*Prem +
InitCommSA[[typ]]*PC,
CommVector[[i]] = RenCommP[[typ]]*Prem +
RenCommSA[[typ]]*PC]];
ExpensesVector = ConstantArray[{}, periodnew];
For[i = 1, i <= periodnew, i++,
If[polYearVect[[i]] == 1,
ExpensesVector[[i]] = InitExpFix[[typ]] +
InitExpP[[typ]]*Prem + RenExpFix[[typ]]*
(1 + InflRateFixExp)^(i - 1) +
RenExpP[[typ]]*Prem,
ExpensesVector[[i]] = RenExpFix[[typ]]*
(1 + InflRateFixExp)^(i - 1) +
RenExpP[[typ]]*Prem]];
PremiumReal = PremVector*inforceBoY;
CommReal = CommVector*inforceBoY;
ExpensesReal = ExpensesVector*inforceBoY;
NetPremium = PremiumReal - CommReal - ExpensesReal;
```

```
NetPremEq = PadRight[NetPremium, 65];
PayoutSA = (numDeath + numMatur)*PC;
PayoutSAEq = PadRight[PayoutSA, 65];
Decrems = numDeath + numMatur +
numLapses*(1 - SurrFee[[typ]]);
PayoutCV = f0*Decrems;
PayoutCVEq = PadRight[PayoutCV, 65];
SavPrem = PremVector - AlphaVect - BetaVect -
GammaVect - riskpr;
PayoutSP = LowerTriangularize[Table[Decrems[[i]]*
SavPrem[[j]], {i, 1, Length[Decrems]},
{j, 1, Length[Decrems]}]];
PayoutSPEq = PadRight[PayoutSP, {65, 65}];
{NetPremEq, PayoutSAEq, PayoutCVEq, PayoutSPEq}]
```

In[7]:=

```
fceScen[scen_, cc_, dd_] :=
Module[{invIncome, incomerate, discrate, discfact,
        Tdiscfact, itriangle, PayoutSPint, sumPayoutSPint},
invIncome = Scenarios[[scen]];
incomerate = 1 + Map[Max[TIR[[1]], #] &,
invIncome - InvestmentMargin];
discrate = DiscountScenarios[[scen]];
discfact = Table[Product[1/(1 + discrate[[i]]),
{i, 1, k}], {k, 1, Length[discrate]}];
Tdiscfact = Table[Product[incomerate[[i]], {i, 1, j}],
{j, 1, Length[discrate]}]*cc;
itriangle =
LowerTriangularize[Table[Product[incomerate[[i]],
{i, l, j}], {j, 1, Length[discrate]},
{l, 1, Length[discrate]}]];
PayoutSPint = itriangle*dd;
sumPayoutSPint = ConstantArray[0, Length[discrate]];
For[w = 1, w <= Length[discrate], w++,
sumPayoutSPint[[w]] = Total[PayoutSPint[[w]]]];
{Tdiscfact, sumPayoutSPint, discfact}]
```

In[8]:=

```
VektCF := Module[{AOld, BOld, COld, DOld, ANew, APol,
        BNew, BPol, CNew, CPol, DNew, DPol, allscen, C2, D2,
        discF},
AOld = ConstantArray[0, 65];
BOld = ConstantArray[0, 65];
COld = ConstantArray[0, 65];
DOld = ConstantArray[0, 65];
For[u = 1, u <= Length[PolicyType], u++,
{APol, BPol, CPol, DPol} = fcepro1MP[PolicyType[[u]],
IncDates[[u]], EntryAge[[u]], Sex[[u]],
PolicyPeriod[[u]], ValuationDate,
SumAssured[[u]], Premiums[[u]],
CVatValDate[[u]], CountPol[[u]]];
```

```
ANew = APol + AOld; AOld = ANew;
BNew = BPol + BOld; BOld = BNew;
CNew = CPol + COld; COld = CNew;
DNew = DPol + DOld; DOld = DNew];
allscen = ConstantArray[0, Length[Scenarios]];
For[s = 1, s <= Length[Scenarios], s++,
{C2, D2, discF} = fceScen[s, CNew, DNew];
allscen[[s]] =
Total[(ANew - BNew - C2 - D2)*discF]];
allscen]
```