Oppgave 7

Løsning

```
{α, β, c} = {0, 0.0000202, 1.1015};
i = 0.03;
ξ = 0.7;
v = 1/(1 + i);
x = 30;
n = 35;
ω = 120;
λ = 0.03;
θ = 0.015;
lønn = 500 000;
μ = 0.057;
σ = 0.056;
```

```
<< "PlotLegends`"
<< "BarCharts`"; << "Histograms`"; << "PieCharts`"
```

$$p[y\_, t\_] := e^{-\left(\alpha\, t + \frac{\beta\, c^y\, (c^t - 1)}{\text{Log}[c]}\right)};$$

$$a[t\_] := \sum_{j=n-t}^{\omega-(x+t)} v^j\, p[x + t, j];$$

Tabellerer funksjonene for de verdiene som brukes. Tidsbesparende: Hente verdi i tabell tar kortere tid enn et funksjonskall.

```
at = Table[a[t], {t, 0, n - 1}];
px = Table[p[x, t], {t, 0, n - 1}];
```

nSim = antall simuleringer
ran1 = (n-1) x nSim matrise med N(0,1)-simuleringer
ran2 = n x nSim matrise med N(0,1)-simuleringer
simVt = simulert premiereserve
simFt = simulert forsikringsfond uten garanti
simFtG = simulert forsikringsfond med garanti
innbet = simulerte innbetalinger uten garanti
innbetG = simulerte innbetalinger med garanti

```
nSim = 1000;

ran1 = Partition[RandomReal[NormalDistribution[0, 1], nSim (n - 1)], n - 1];
ran2 = Partition[RandomReal[NormalDistribution[0, 1], nSim n], n];

simInnbet[sim_] := Module[{simL, simA, simP, simS, simVt, simFt}, simL = FoldList[(1 + λ) #1 + θ #2 #1 &, lønn, ran1[[sim]]];
  simS = 0.2 simL;
  simP = Prepend[Table[((t + 1) simS[[t + 1]])/n - (t simS[[t]])/n) at[[t + 1]], {t, 1, n - 1}], (simS[[1]] at[[1]])/n]; simA = e^(μ - σ²/2 + σ ran2[[sim]]);
  simVt = Table[{t, ((t + 1) simS[[t + 1]] at[[t + 1]] px[[t + 1]])/n}, {t, 0, n - 1}];
  simFt = Table[{t + 1, simVt[[t + 1, 2]] simA[[t + 1]]}, {t, 0, n - 1}];
  simFtG = Table[{t + 1, simVt[[t + 1, 2]] ((1 + i) + ξ Max[0, simA[[t + 1]] - (1 + i)])}, {t, 0, n - 1}];
  innbet = Prepend[Table[simVt[[t + 1, 2]] - simFt[[t, 2]], {t, 1, n - 1}], simP[[1]]];
  innbetG = Prepend[Table[simVt[[t + 1, 2]] - simFtG[[t, 2]], {t, 1, n - 1}], simP[[1]]];
  {innbet, innbetG}];
```

Mulige baner for faktisk nødvendig innbetaling med og uten garanti:

```
innbetEksempel = simInnbet[4];
plot1 = Plot[{innbetEksempel[[1, Round[k]]], innbetEksempel[[2, Round[k]]]}, {k, 1, 35}, PlotRange → All,
  PlotStyle → (RGBColor[#1, 0, 0] &) /@ {0, 1}]
```

```
Part::pspec: Part specification Round[k] is neither an integer nor a list of integers. ≫
```

```
Part::pspec: Part specification Round[k] is neither an integer nor a list of integers. ≫
```

```
Part::pspec : Part specification Round[k] is neither an integer nor a list of integers. >>
```

```
General::stop : Further output of Part::pspec will be suppressed during this calculation. >>
```
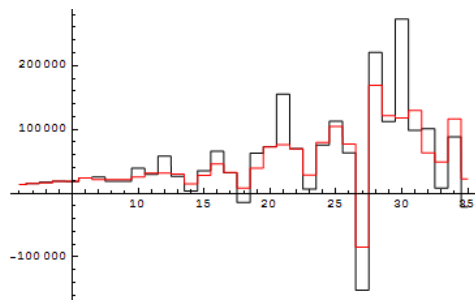


```
(*Display["p:STK4500/Oppgaver/Oppgave7/plot1.eps",Show[plot1],"EPS"];*)
```

Trakt:

```
Timing[innbetTab = Table[simInnbet[j], {j, nSim}];]
```

```
{1.389, Null}
```

```
trakt = ({Take[Sort[#1], {0.05 nSim}], Take[Sort[#1], {0.95 nSim}]} &) /@ Transpose[Transpose[innbetTab][[1]]];
traktG = ({Take[Sort[#1], {0.05 nSim}], Take[Sort[#1], {0.95 nSim}]} &) /@ Transpose[Transpose[innbetTab][[2]]];
```

```
plot2 = Plot[{trakt[[Round[k], 1]], trakt[[Round[k], 2]], traktG[[Round[k], 1]], traktG[[Round[k], 2]]}, {k, 1, 35},
   PlotRange → All, PlotStyle → (RGBColor[#1, 0, 0] &) /@ {0, 0, 1, 1}, BaseStyle → {11, FontFamily → "Helvetica"}]
```

```
Part::pspec : Part specification Round[k] is neither an integer nor a list of integers. >>
```

```
Part::pspec : Part specification Round[k] is neither an integer nor a list of integers. >>
```

```
Part::pspec : Part specification Round[k] is neither an integer nor a list of integers. >>
```

```
General::stop : Further output of Part::pspec will be suppressed during this calculation. >>
```
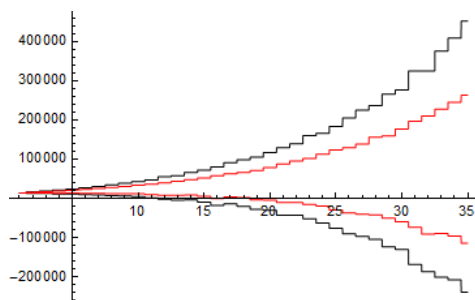


```
(*Display["p:STK4500/Oppgaver/Oppgave7/plot2.eps",Show[plot2],"EPS"];*)
```

Ekstra: Animering (Ctrl y). Ser ikke bare på 5 % og 95 % kvantiler som over, men på 5 %, 10 %, ..., 95 %.

```
trakt = Table[(Take[Sort[#1], {pr nSim/100}] &) /@ Transpose[Transpose[innbetTab][[1]]], {pr, 5, 95, 5}];
traktG = Table[(Take[Sort[#1], {pr nSim/100}] &) /@ Transpose[Transpose[innbetTab][[2]]], {pr, 5, 95, 5}];
```
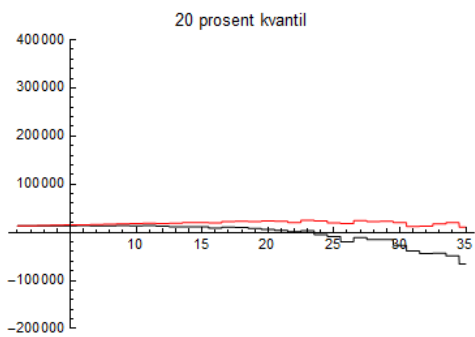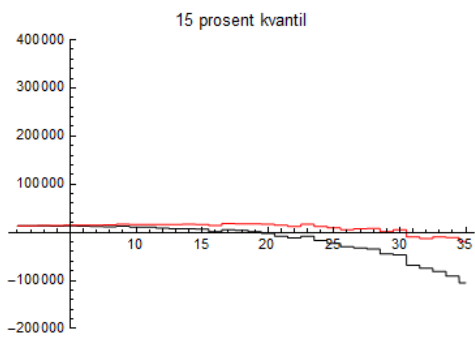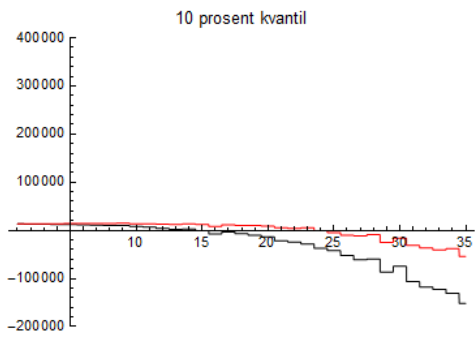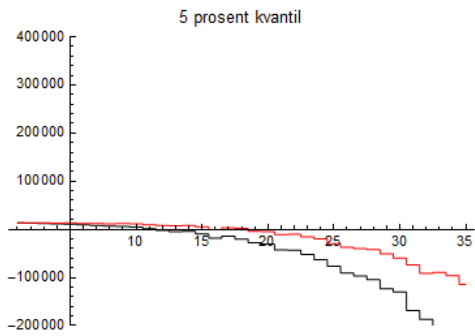
```
Do[Print[Plot[{trakt[[j, Round[k]]], traktG[[j, Round[k]]]}, {k, 1, 35}, PlotRange → {-200 000, 400 000},
    PlotStyle → (RGBColor[#1, 0, 0] &) /@ {0, 1}, PlotLabel → ToString[5 j] <> " prosent kvantil",
    BaseStyle → {11, FontFamily → "Helvetica"}]], {j, 1, 19}];
```

```
Part::pspec : Part specification Round[k] is neither an integer nor a list of integers. >>
```
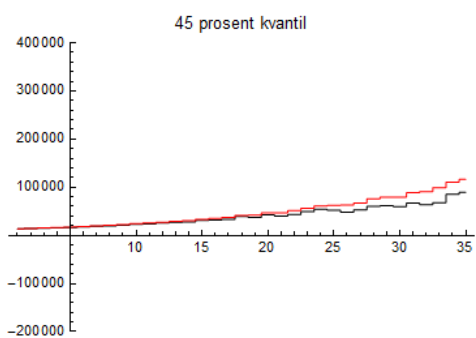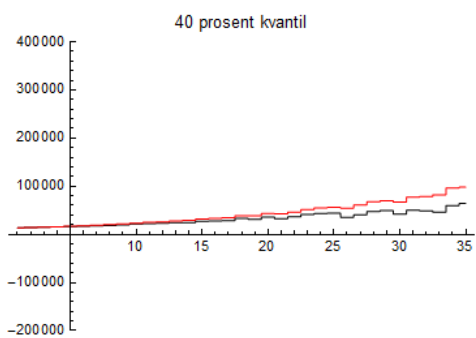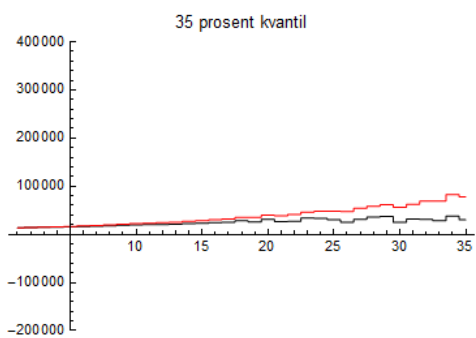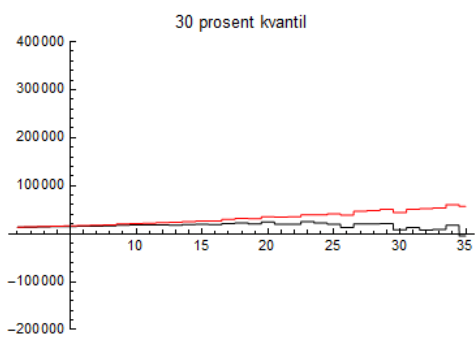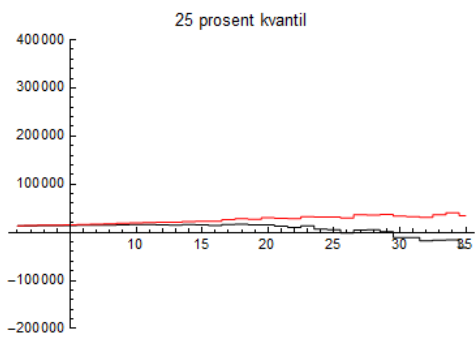
Part::pspec: Part specification Round[k] is neither an integer nor a list of integers. »
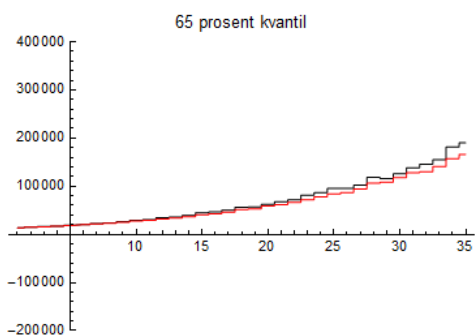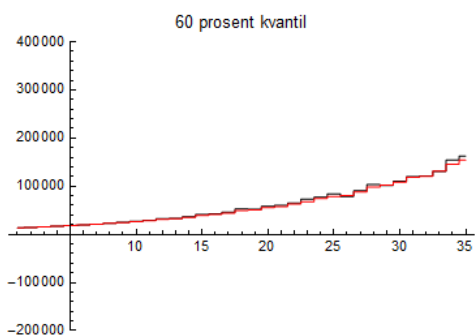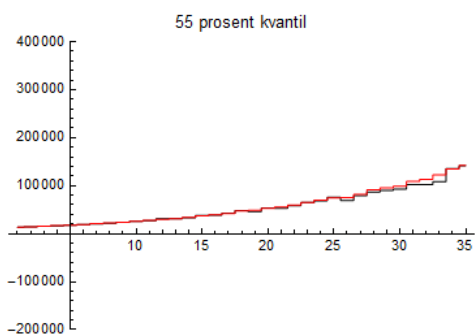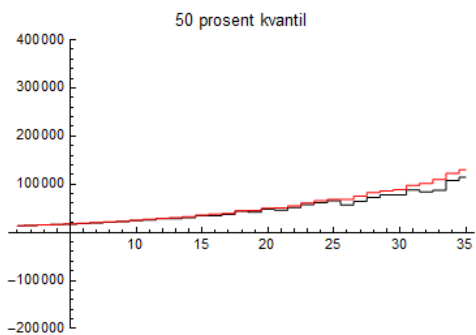
Part::pspec: Part specification Round[k] is neither an integer nor a list of integers. »

General::stop: Further output of Part::pspec will be suppressed during this calculation. »

5 prosent kvantil

10 prosent kvantil

15 prosent kvantil

20 prosent kvantil

25 prosent kvantil

30 prosent kvantil

35 prosent kvantil

40 prosent kvantil

45 prosent kvantil

50 prosent kvantil

55 prosent kvantil

60 prosent kvantil

65 prosent kvantil

70 prosent kvantil

75 prosent kvantil

80 prosent kvantil

85 prosent kvantil

90 prosent kvantil

95 prosent kvantil
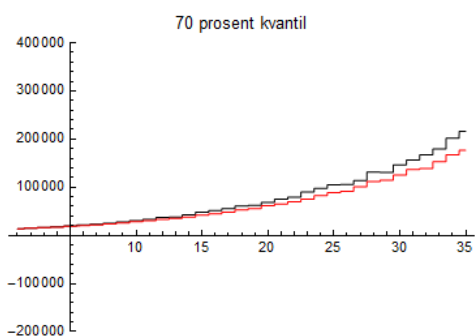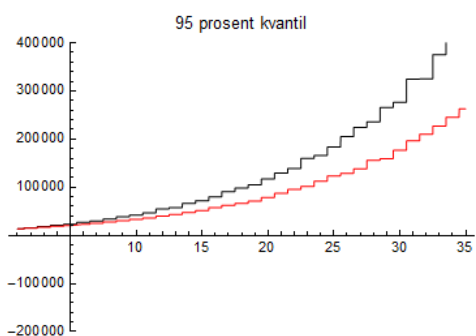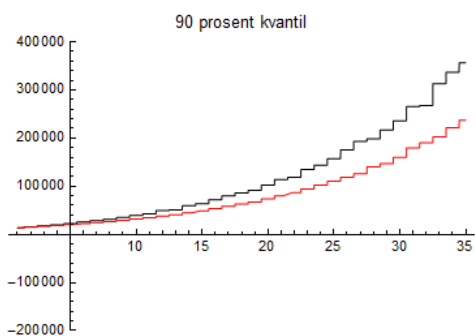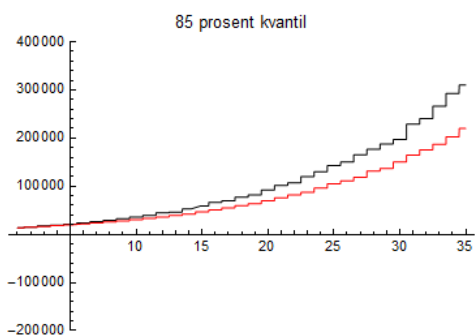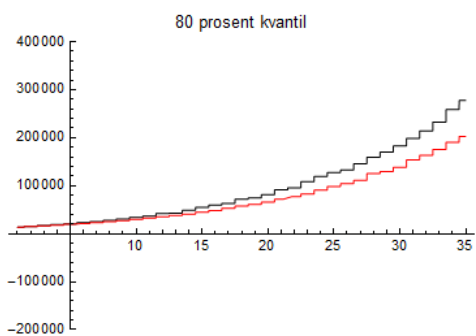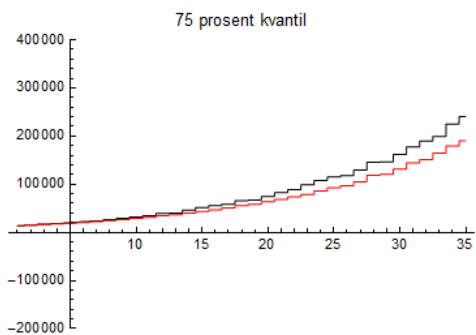
Kontantverdi til innbetalingene med og uten garanti:

```
simAvkastning = Table[e^(μ - σ²/2 + σ ran2[j]), {j, 1, nSim}];

kontantVerdi = Table[innbetTab[[j, 1]].Delete[FoldList[#2 #1 &, 1, 1/simAvkastning[[j]]], n], {j, 1, nSim}];

kontantVerdiG = Table[innbetTab[[j, 2]].Delete[FoldList[#2 #1 &, 1, 1/simAvkastning[[j]]], n], {j, 1, nSim}];

plot3 = Show[Histogram[kontantVerdi/10^6, HistogramCategories → Table[i/30, {i, 0, 1000}], HistogramRange → {0, 2},
    DisplayFunction → Identity], Histogram[kontantVerdiG/10^6, HistogramCategories → Table[i/30, {i, 0, 1000}],
   HistogramRange → {0, 2}, DisplayFunction → Identity, BarStyle → RGBColor[1, 1, 1]], DisplayFunction → $DisplayFunction,
  Ticks → {Automatic, False}, BaseStyle → {11, FontFamily → "Helvetica"}]
```
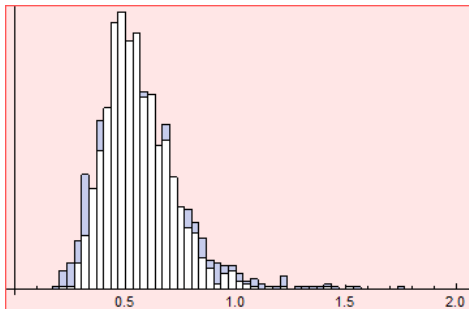


```
(*Display["p:STK4500/Oppgaver/Oppgave7/plot3.eps",Show[plot3],"EPS"];*)

(Show[Histogram[kontantVerdi/kontantVerdiG [[Range[#1]]], HistogramCategories → Table[i/20, {i, 0, 100}], HistogramRange → {0.5`, 1.5`},
     DisplayFunction → Identity, BaseStyle → {11, FontFamily → "Helvetica"}], Plot[200, {x, 0.5`, 1.5`}, DisplayFunction → Identity],
   DisplayFunction → $DisplayFunction, PlotLabel → "Antall simuleringer:\t" <> ToString[#1]] &) /@ Range[10, nSim, 10];
```

Kan lage en alternativ funksjon som ikke lagrer grafikken og dermed sparer minne. Problemet er at animeringen ikke blir like glatt.

```
visAnimering := Module[{nb = EvaluationNotebook[], Target, tar, tid = 0}, Target = ToString[tar] <> ToString[$SessionID];
   CellPrint[Cell["", "Graphics", CellTags → Target]];
   (Show[Histogram[kontantVerdi/kontantVerdiG [[Range[#1]]], HistogramCategories → Table[i/20, {i, 0, 100}], HistogramRange → {0.5`, 1.5`},
       BarStyle → RGBColor[1, 1, 1], DisplayFunction → Identity, BaseStyle → {35, FontFamily → "Helvetica"}],
     Plot[200, {x, 0.5`, 1.5`}, DisplayFunction → Identity], PlotLabel → "Antall simuleringer:\t" <> ToString[#1],
     ImageSize → 800, DisplayFunction → (NotebookFind[nb, Target, All, CellTags];
       NotebookWrite[nb, Cell[GraphicsData["PostScript", ExportString[#1, "EPS"]], "Graphics", CellTags → Target]] &) &) /@
    Range[10, nSim, 10];]

(*visAnimering*)
```

Created with Wolfram *Mathematica* 6