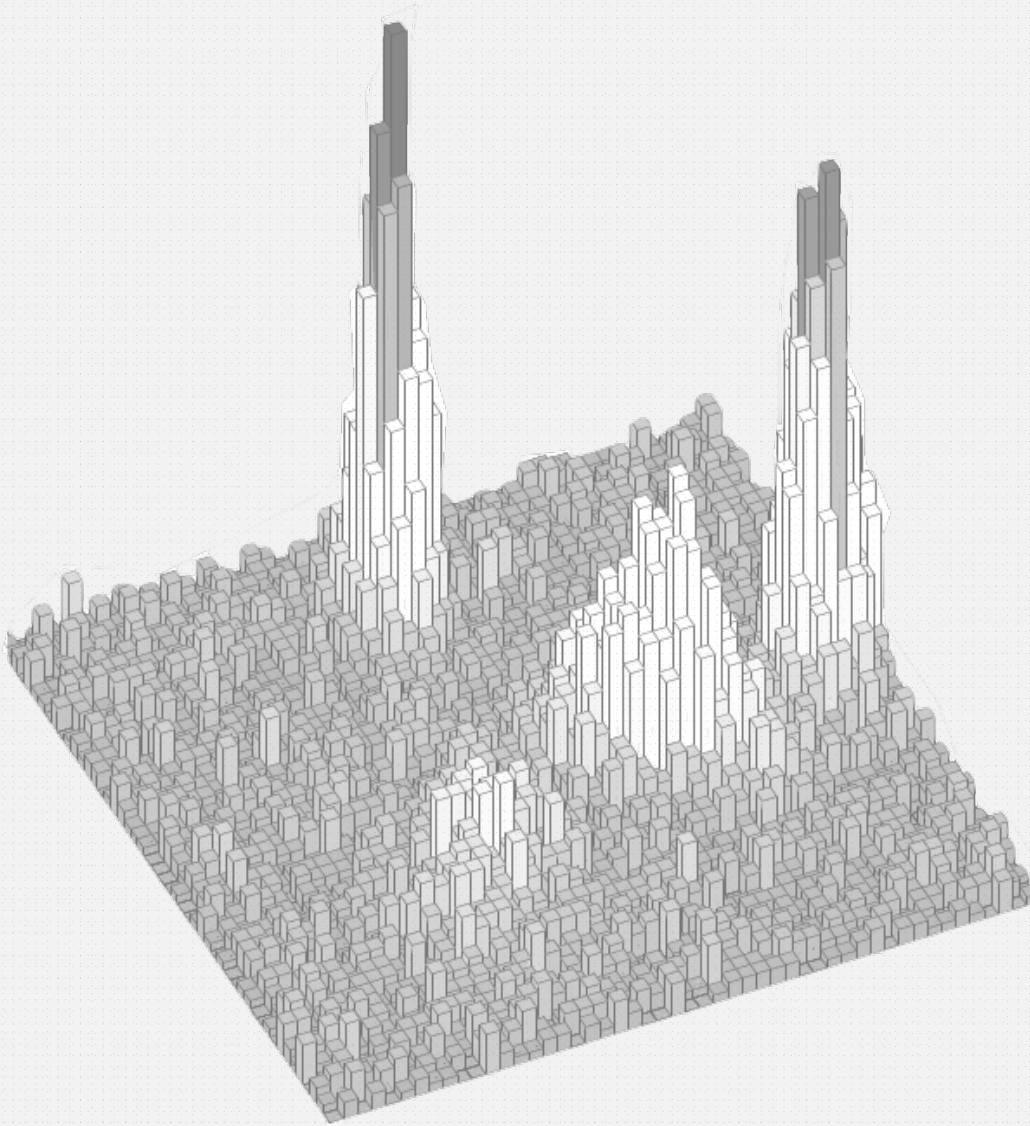


Practicum 2 - Monte Carlo-Simulaties

Bruno Vandekerkhove



Inhoudsopgave

Simuleren van Sparen en Beleggen	1
1.1 Opdracht 1	1
1.2 Opdracht 2	1
1.3 Opdracht 3	1
1.4 Opdracht 4	1
1.5 Opdracht 5	2
1.6 Opdracht 6	2
1.7 Opdracht 7	2
1.8 Opdracht 8	3
1.9 Opdracht 9	3
1.10 Opdracht 10	5
1.11 Opdracht 11	5
1.12 Opdracht 12	7
Evaluatie	7

Simuleren van Sparen en Beleggen

De broncode bevindt zich in de **src** folder. Het algemene script (**src/s0216676_script**) is opgedeeld in secties, één per opgave. Elke opgave wordt hieronder afzonderlijk beantwoord. Aan het einde van elk antwoord wordt (indien nodig) de broncode weergegeven.

Opdracht 1

Hier is een implementatie :

```
1 function [yield, invested, value] = s0216676_simulateSavingInvesting(budget, rate, months)
2     value = repelem(budget * 1.02 .^ (0:floor(months/12)), 1, 12);
3     invested = sum(value(1:months));
4     for j = 13:12:months % Consider each month of january
5         win = sum(value(j-12:j-1) .* (((12:-1:1)/12) * (rate/100))); % Calculate savings
6         value(j) = value(j) + (win - 0.15 * (win > 980) * (win - 980));
7         value(j-12:j) = cumsum(value(j-12:j)); % Accumulate sums
8     end
9     value = value(1:months); value(j:end) = cumsum(value(j:end));
10    yield = value(months) / invested - 1;
11 end
```

Opdracht 2

Het resultaat van de gegeven code is te zien in figuur 1. De totale investering bedraagt zo'n 96091 euro. De relatieve winsten bedragen 5.46%, 11.34%, 23.34% en 50.82%.

Opdracht 3

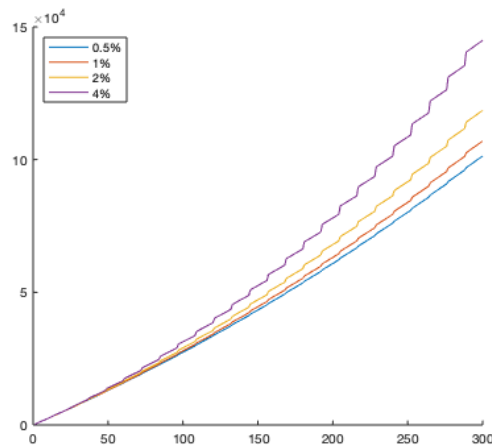
Het bestand wordt ingeladen.

```
1 load('Funds.mat')
```

Opdracht 4

Men kan de parameters berekenen door eerst σ te bepalen op basis van de rendementen, het gemiddelde te berekenen van diezelfde rendementen en vervolgens μ daaruit te bepalen.

De bekomen parameters bedragen $\mu = 3.208565 \times 10^{-3}$, $\sigma = 3.070426 \times 10^{-2}$ (voor EUN5) en $\mu = 7.857782 \times 10^{-3}$, $\sigma = 3.277546 \times 10^{-2}$ (voor VWRL).



Figuur 1: Simulatie van spaarrekeningen met verschillende rentevoeten.

```

1 function [mu,sigma] = s0216676_estimateParameters(s)
2     rendements = log(s(2:end) ./ s(1:end-1));q
3     sigma = std(rendements);
4     mu = mean(rendements) + 0.5 * sigma^2;
5 end

```

Opdracht 5

De implementatie maakt gebruik van een eenvoudige `for` loop.

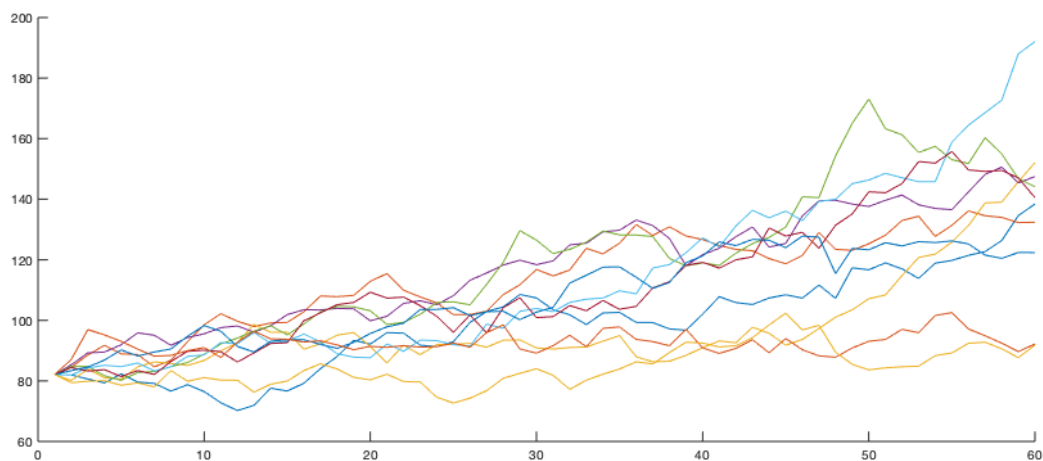
```

1 function [path] = s0216676_simulateFundPath(initialPrice, mu, sigma, months)
2     alpha = mu - 0.5 * sigma^2;
3     path = [initialPrice 2:months];
4     for t = 2:months
5         path(t) = path(t-1) * exp(alpha + sigma * randn);
6     end
7 end

```

Opdracht 6

In figuur 2 worden de resulterende paden afgebeeld. Een aantal paden wijken komen ruwweg van de werkelijke koers afgebeeld in de opgave en de ze komen allen realistisch over. De initiële waarde bedraagt telkens 82.10 euro wat overeenstemt met de waarde in dollar op 29 november 2019.



Figuur 2: Simulatie van spaarrekeningen met verschillende rentevoeten.

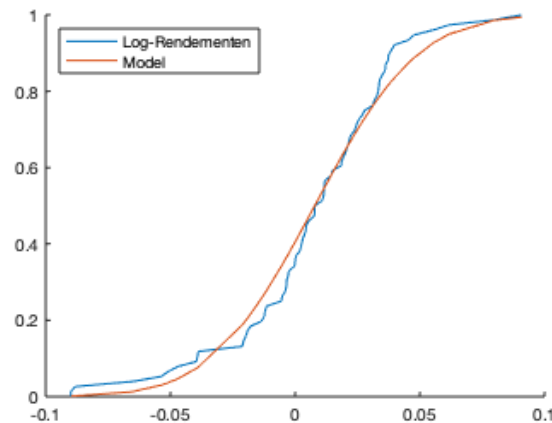
```

1 figure; hold all;
2 for i = 1:10
3     plot(s0216676_simulateFundPath(82.10, mu, sigma, 60)); % (29/11/2019)
4 end

```

Opdracht 7

De cumulatieve distributiefuncties zijn afgebeeld in figuur 3. Op basis hiervan kan men met een zekere graad van vertrouwen beweren dat de log-rendementen effectief normaal verdeeld is. Normaal gezien maakt men gebruik van specifieke testen om dit te kwantificeren.



Figuur 3: Cumulatieve distributiefuncties van de log-rendementen en het model.

```

1 [f,x] = ecdf(log(VWRL(2:end) ./ VWRL(1:end-1)));
2 [f_norm] = normcdf(x, mu, sigma);
3 figure; hold all;
4 plot(x, f, x, f_norm);
5 legend('Log-Rendementen', 'Model', 'Location', 'NorthWest');

```

Opdracht 8

Het kon ook met een lus, maar ik deed het deze keer met ingebouwde functies.

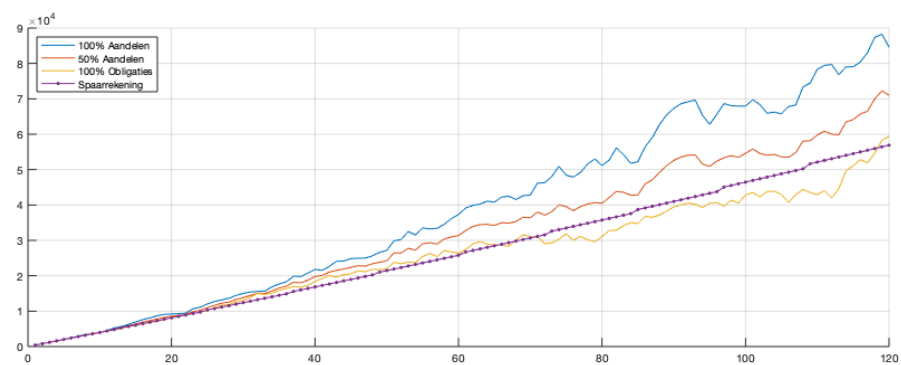
```

1 function [yield, invested, value, units] = ...
2     s0216676_simulateFundInvestingPath(budget, pricePath, alpha)
3 months = size(alpha,1);
4 budgets = repelem(budget * (1.02 .^ (0:floor(months/12))), 12, 2);
5 invested = sum(budgets(1:months,1));
6 units = (budgets(1:months,:) .* [alpha (1-alpha)] - 6) / 1.0035;
7 for i = 1:2
8     units(units(:,i) < 0, 3-i) = (budgets(units(:,i) < 0) - 6) / 1.0035;
9     units(units(:,i) < 0, i) = 0;
10 end
11 units = cumsum(units ./ pricePath);
12 value = units .* pricePath;
13 yield = sum(value(end,:)) / invested - 1;

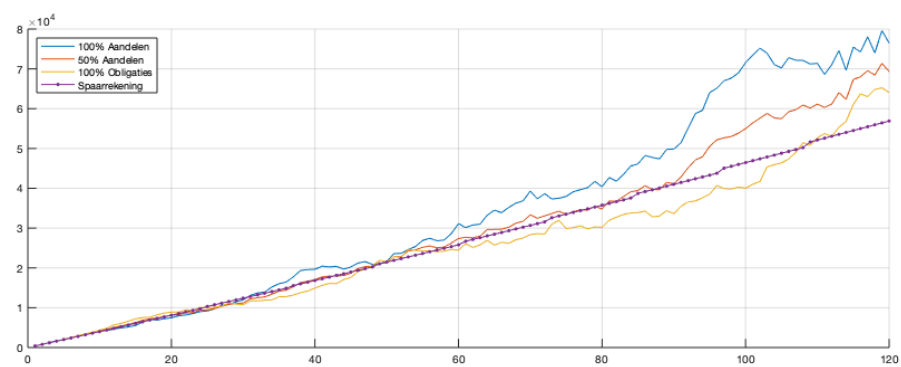
```

Opdracht 9

De resulterende figuren zijn afgebeeld op de volgende pagina.

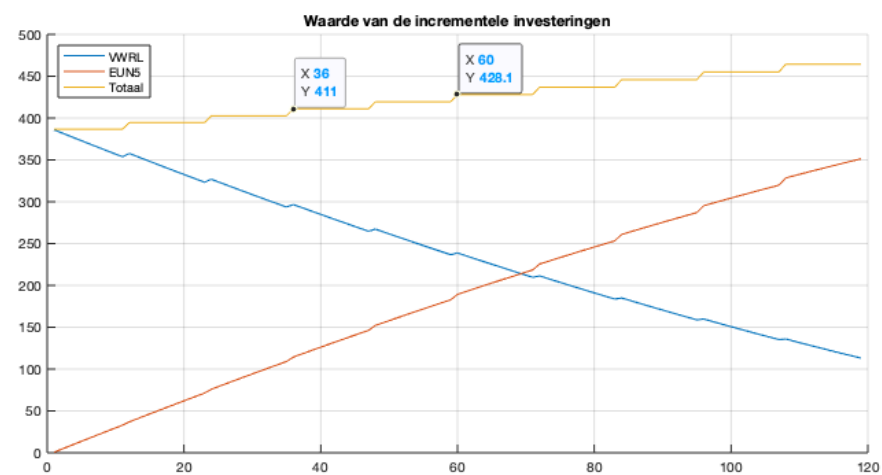


(a) Iteratie 1



(b) Iteratie 2

Figuur 4: Waarde van vier type investeringen doorheen een periode van 120 maanden.



Figuur 5: Waarde van de investering op maandelijkse basis.

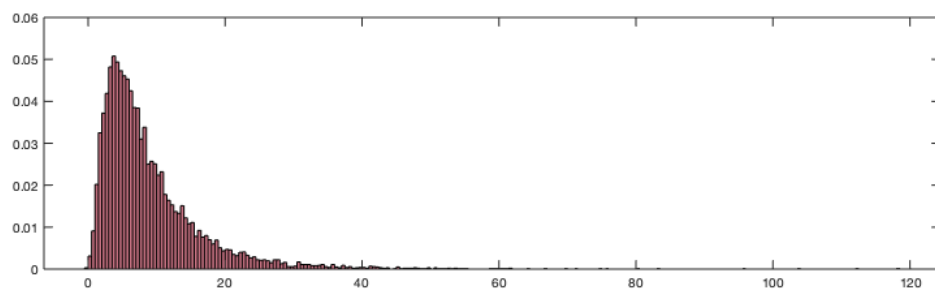
Opdracht 10

De functie werd als volgt geïmplementeerd.

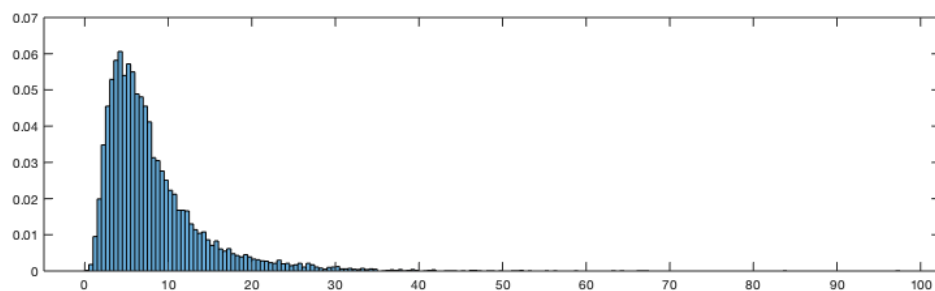
```
1 function [yields,invested] = s0216676_simulateFundInvesting(budget, priceHistory, alpha, N)
2     months = size(alpha, 2);
3     [mu1,sigma1] = s0216676_estimateParameters(priceHistory(:,1));
4     [mu2,sigma2] = s0216676_estimateParameters(priceHistory(:,2));
5     yields = 1:N;
6     for i = 1:N
7         path1 = s0216676_simulateFundPath(priceHistory(end,1), mu1, sigma1, months);
8         path2 = s0216676_simulateFundPath(priceHistory(end,2), mu2, sigma2, months);
9         [yields(i),invested,~,~] = s0216676_simulateFundInvestingPath(budget, [path1 path2], ...
10             alpha);
11     end
12 end
```

Opdracht 11

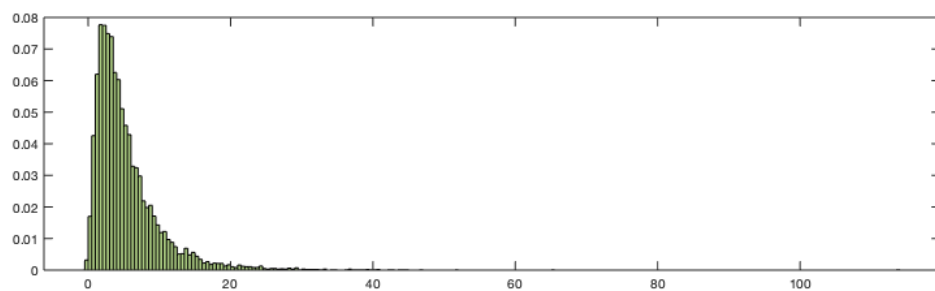
De histogrammen staan gezamenlijk afgebeeld in figuur 7 (zie volgende bladzijde) en afzonderlijk hieronder. Het rendement van de spaarrekening bedroeg ongeveer 47.22%. Het valt op dat hoe agressiever men speelt, hoe waarschijnlijker het wordt dat men grote winsten boekt.



(a) Agressief

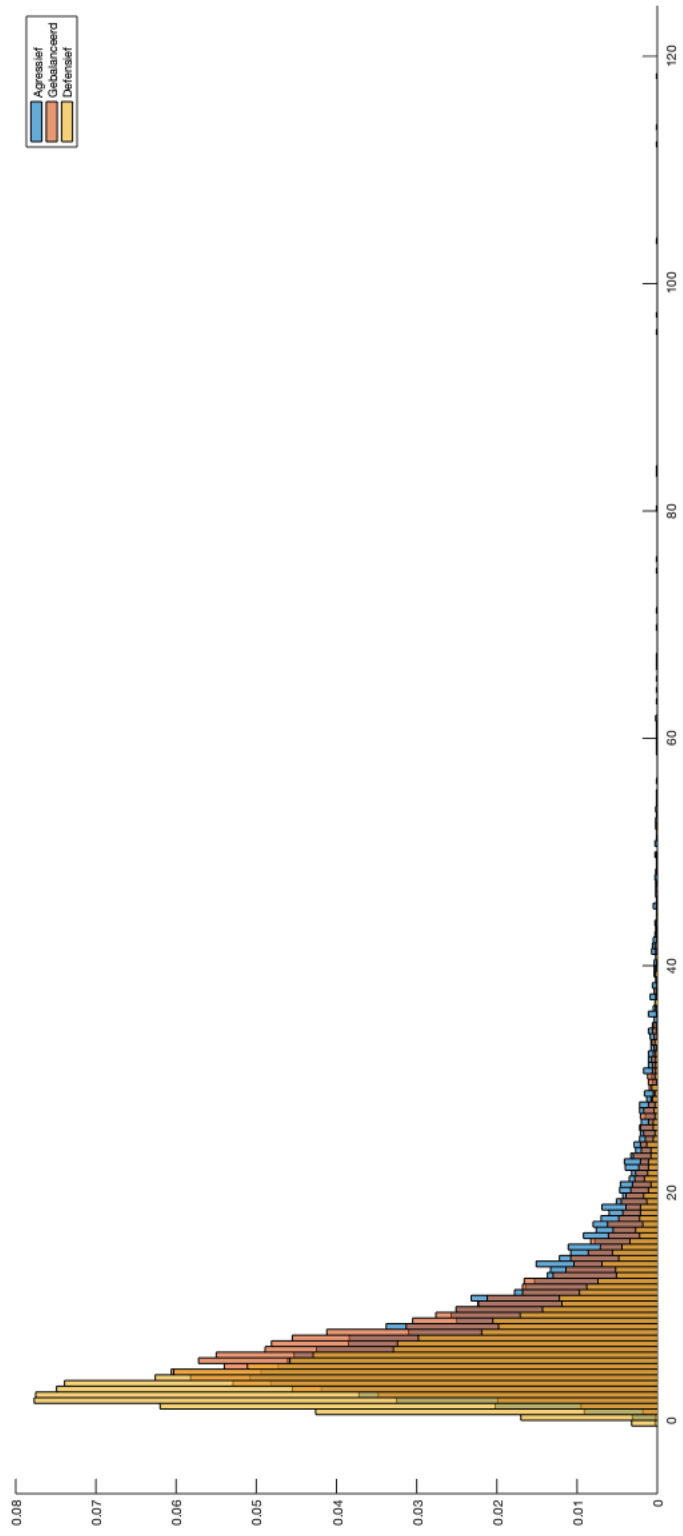


(b) Gebalanceerd



(c) Defensief

Figuur 6: Afzonderlijke histogrammen gegenereerd na Monte-Carlo simulaties.



Figuur 7: Histogram gegenereerd na het uitvoeren van de Monte-Carlo simulaties.

Opdracht 12

In tabellen 1 en 2 worden eerst de kwantielen beschouwd (met $x = 28, y = 76$), vervolgens de kans op negatieve eindrendementen en ten slotte de kans op het behalen van een vermogen van 750.000 euro. Het mediaan van het vermogen bereikt op de leeftijd van 70 jaar bedroeg 481.385, 1.613.076, 1.545.397 en 1.124.821 euro voor de vier scenario's (spaarrekening, agressief, gebalanceerd, defensief). Wat zou leiden tot een maandelijks inkomen van 3.863, 4.033, 3.863 en 2.812 euro. Enkel de spaarrekening zou dus niet tot het gewenste maandelijks inkomen van 1.790 euro leiden.

		0.1%	2.5%	25%	50%	75%	97.5%
$n = 300$	Spaarrekening	0.3528	0.3528	0.3528	0.3528	0.3528	0.3528
	Agressief	0.0594	0.2239	1.0057	1.6295	2.5004	5.1955
	Gebalanceerd	0.2826	0.4080	1.0486	1.5445	2.2071	4.3076
	Defensief	-0.1099	0.0196	0.6483	1.1841	1.9159	4.1599
$n = 12 * (70 - x)$	Spaarrekening	0.6449	0.6449	0.6449	0.6449	0.6449	0.6449
	Aggressive	0.5868	0.9126	2.7824	4.5118	7.2536	17.2040
	Gebalanceerd	0.9435	1.2371	2.8345	4.2806	6.4681	14.1306
	Defensief	0.1593	0.3652	1.6187	2.8435	4.7228	11.2958

Tabel 1: Kwantielen voor de 8 mogelijke scenario's.

		$P(y < 0)$	$P(v \geq 750.000)$
$n = 300$	Spaarrekening	0.0	0.0
	Agressief	0.0064	0.0561
	Gebalanceerd	0.0003	0.0281
	Defensief	0.0224	0.0245
$n = 12 * (70 - x)$	Spaarrekening	0.0	0.0
	Aggressive	0.0003	0.9167
	Gebalanceerd	0.0	0.9463
	Defensief	0.0036	0.7639

Tabel 2: Kans op negatieve eindrendementen en op het bereiken van een vermogen van 750.000 euro voor de 8 mogelijke scenario's.

Evaluatie

Ik besteedde ongeveer 17 uur aan het practicum en zo'n 3 uur aan het verslag. De moeilijkheidsgraad lag goed. De terminologie was meer dan duidelijk en de taak (net zoals het andere practicum) best interessant. De simulatie was niet de meest realistische omdat het onderliggende model een beperkt voorspellingsvermogen heeft. Dat de kans op een negatief rendement zo laag is, is al op zich vrij onwaarschijnlijk.