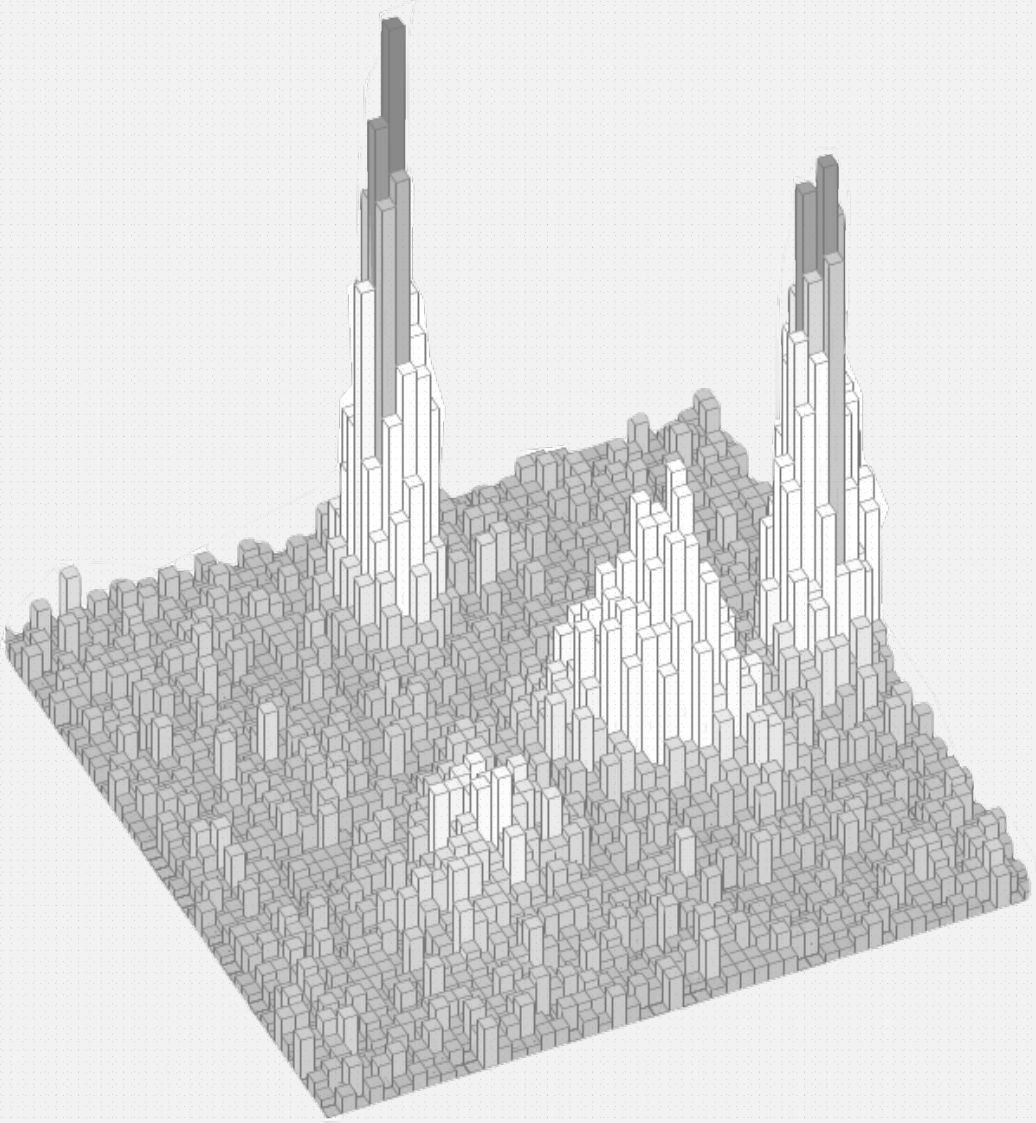
Practicum 2 - Monte Carlo-Simulaties

Bruno Vandekerkhove



Academisch Jaar 2019

G0Q57A: Modellering & Simulatie

Inhoudsopgave

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 :

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

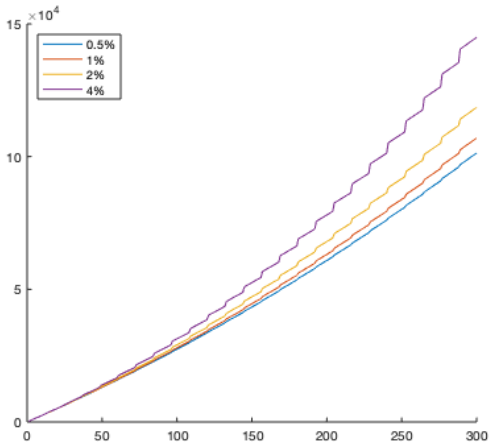
Het bestand wordt ingeladen.

```
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.

```
function [mu,sigma] = s0216676_estimateParameters(s)
    rendements = log(s(2:end) ./ s(1:end-1)); % Base doesn't matter
    sigma = std(rendements);
    mu = mean(rendements) + 0.5 * sigma^2;
end
```

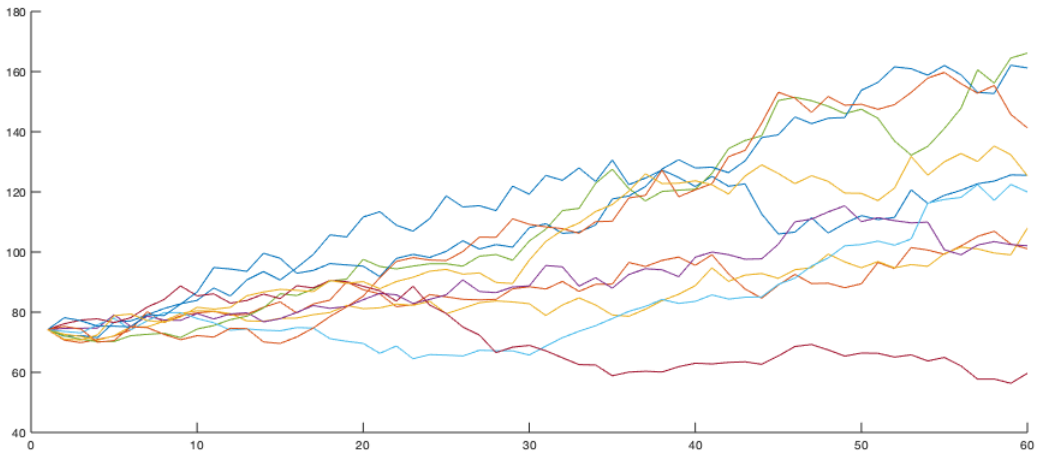
Opdracht 5

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

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

Opdracht 6

In figuur 2 worden de resulterende paden afgebeeld. Een aantal paden wijken af van de werkelijke koers afgebeeld in de opgave, maar de meesten komen realistisch over. De initiële waarde bedraagt telkens 74.19 euro wat overeenstemt met de waarde in dollar op 26 november 2019.

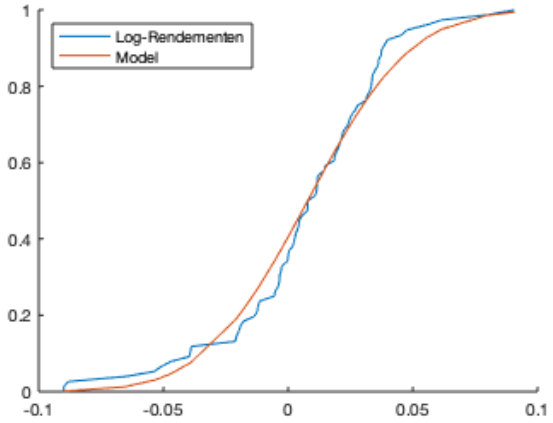


Figuur 2: Simulatie van spaarrekeningen met verschillende rentevoeten.

```
figure; hold all;
for i = 1:10
    plot(s0216676_simulateFundPath(74.19, mu, sigma, 60)); % Converted to euros (26/11/2019)
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.

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

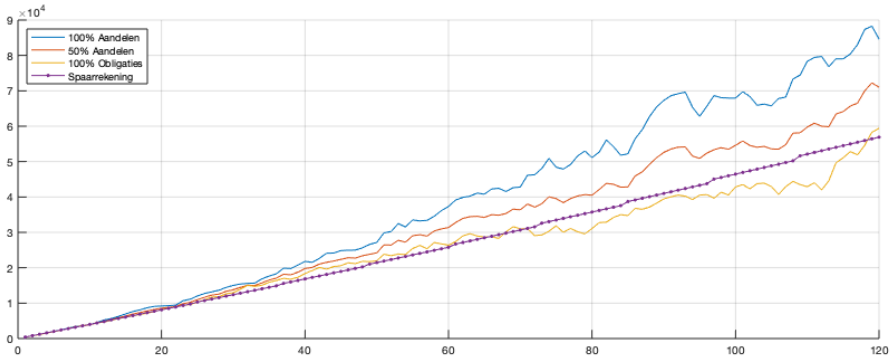
Opdracht 8

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

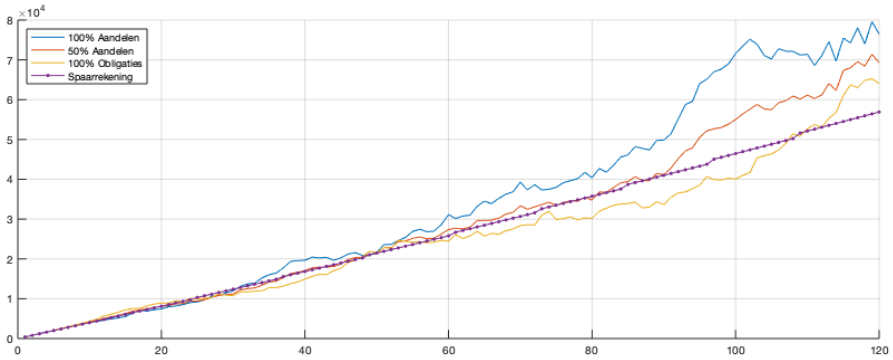
```
function [yield, invested, value, units] = ...
    s0216676_simulateFundInvestingPath(budget,pricePath,alpha)
    months = size(alpha,1);
    budgets = repelem(budget * (1.02 .^ (0:floor(months/12))'), 12, 2);
    invested = sum(budgets(1:months,1));
    units = (budgets(1:months,:) .* [alpha (1-alpha)]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

Opdracht 9

De resulterende figuren zijn afgebeeld op de volgende pagina.

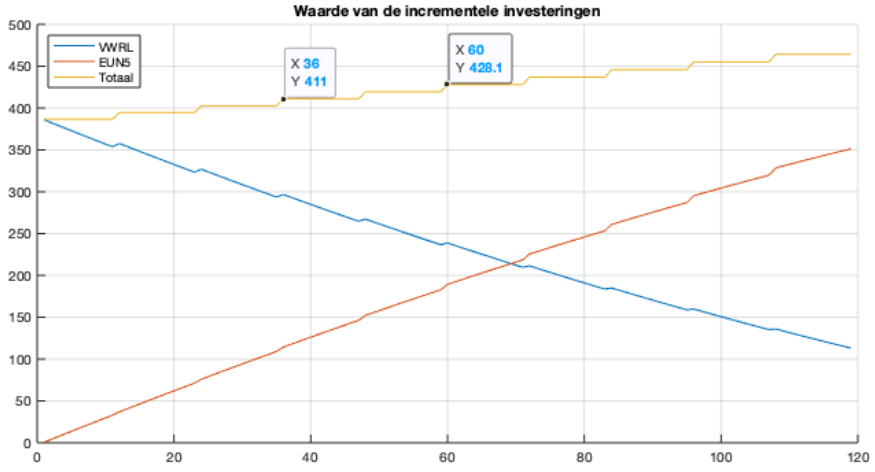


(a) Iteratie 1



(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.

```
function [yields,invested] = s0216676_simulateFundInvesting(budget, priceHistory, alpha, N)
    months = size(alpha, 2);
    [mu1,sigma1] = s0216676_estimateParameters(priceHistory(:,1));
    [mu2,sigma2] = s0216676_estimateParameters(priceHistory(:,2));
    yields = 1:N;
    for i = 1:N
        path1 = s0216676_simulateFundPath(priceHistory(end,1), mu1, sigma1, months);
        path2 = s0216676_simulateFundPath(priceHistory(end,2), mu2, sigma2, months);
        [yields(i),invested,¬,¬] = s0216676_simulateFundInvestingPath(budget, [path1 path2], ...
            alpha);
    end
end
```

Opdracht 11

Opdracht 12

Evaluatie

Ik spendeerde ongeveer 17 uur aan het practicum en zo'n 3 uur aan het verslag. De moeilijkheidsgraad lag volgens mij goed. De terminologie was meer dan duidelijk en de taak (net zoals het andere practicum) best interessant.