

Modelling & Simulation

Afraz Salim

r0439731

Contents

1	Opdracht 1	3
2	Opdracht 2	3
3	Opdracht 4	4
4	Opdracht 5	5
5	Opdracht 6	5
6	Opdracht 7	6
7	Opdracht 8	7
8	Opdracht 9	8
9	Opdracht 10	10
10	Opdracht 11	10
11	Appendix	11

1 Opdracht 1

Schrijf een functie met de hoofding

`function [yield, invested, value] = SN_simulateSavingInvesting(budget, rate, months)`
die een spaarrekening met bovenstaande kenmerken simuleert.

De code voor deze opgave kan gevonden worden in 1.

2 Opdracht 2

Voer de volgende code uit en neem de figuur op in het verslag

.

```
yield = zeros(1,4);  
figure; hold all;  
for k = 1 : 4  
[yield(k), invested, value] = SN_simulateSavingInvesting(250, 2^(-2+k), 300);  
plot(value);  
end  
legend('0.5%', '1%', '2%', '4%', 'Location', 'NorthWest');
```

Figuur gegenereerd met de code uit opdracht 2:

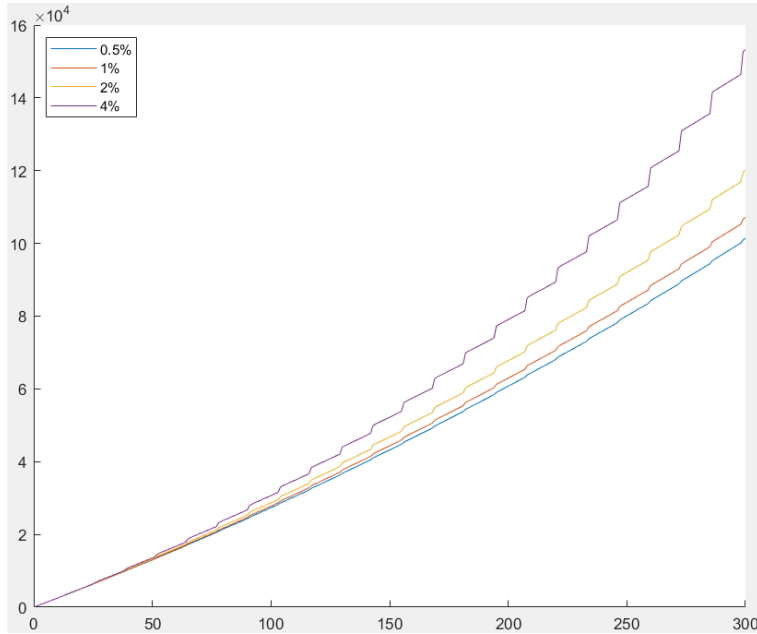


Figure 1: Figuur gegenereerd met de code uit opdracht 2.

Hoeveel bedragen de eindrendementen yield uitgedrukt in % in deze 4 scenario's? Hoeveel geld heb je in elk van deze scenario's zelf gestort, exclusief de ontvangen interesten, i.e., invested?

rate	invested(zonder interesten)	Yield
0.5%	96501.0507	0.050579
1%	96501.0507	0.1099
2%	96501.0507	0.24382
4%	96501.0507	0.58802

3 Opdracht 4

Schrijf een functie met de volgende hoofding

```
function [mu, sigma] = SN_estimateParameters(s)
```

.

Code voor deze opgave kan gevonden worden in 2.

4 Opdracht 5

Schrijf een functie met de hoofding

```
function [path] = SN_simulateFundPath(initialPrice, mu, sigma, months)
```

die een geometrische Brownse beweging met driftfactor μ en volatiliteit σ gedurende $(months-1)$ opeenvolgende maanden simuleert.

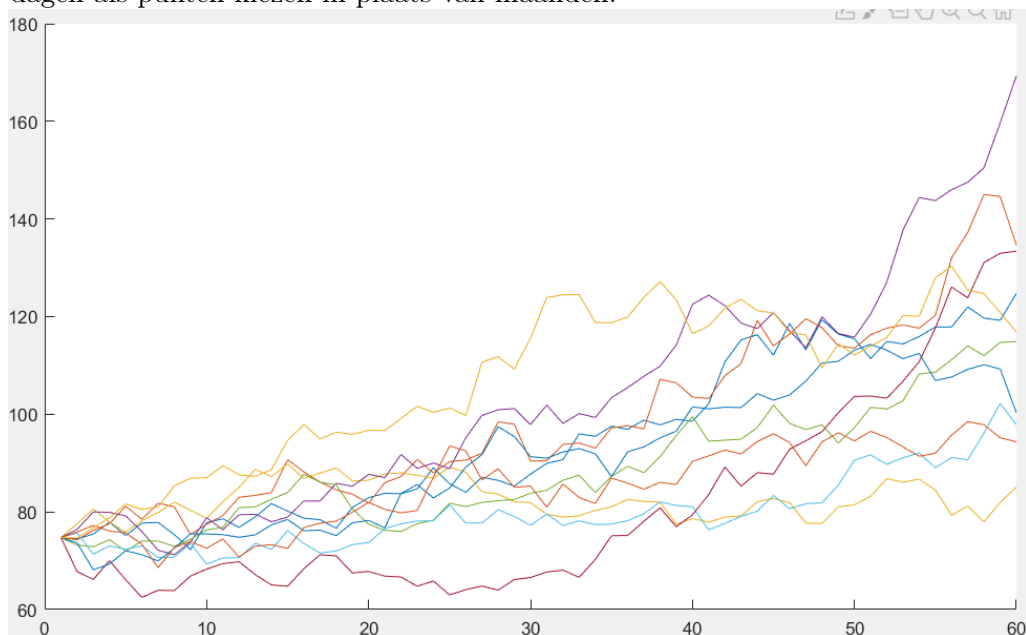
Code voor deze opgave kan gevonden worden in 3.

5 Opdracht 6

Simuleer 10 willekeurige paden van het aandelenfonds VWRL. Als invoer geef je de μ en σ die je in opdracht 4 berekende op basis van de kolomvector VWRL, als initialPrice neem je de meest recente marktprijs die je op <https://finance.yahoo.com/quote/VWRL> kunt raadplegen en als lengte van de simulatie months kies je 60 maanden. Plot deze 10 paden op eenzelfde grafiek en neem ze op in het verslag. Welke initiële prijs heb je gebruikt voor de 10 paden? Vermeld ook de datum. Zien de paden die je gegenereerd hebt er uit zoals de koers van VWRL van de afgelopen 5 jaar zoals in fig. 1? Motiveer indien je meent van niet.

Code voor deze opgave kan gevonden worden in 4.

Wij hebben twee verschillende grafieken op de volgende pagina. De eerste grafiek bestaat uit de gesimuleerde paden en tweede grafiek laat zien de echte pad. Initiële prijs werd gekozen als 83.05\$ op 12/18/2019. De punten worden gesimuleerd op maanden en die zullen meer realistisch lijken indien wij dagen als punten kiezen in plaats van maanden.

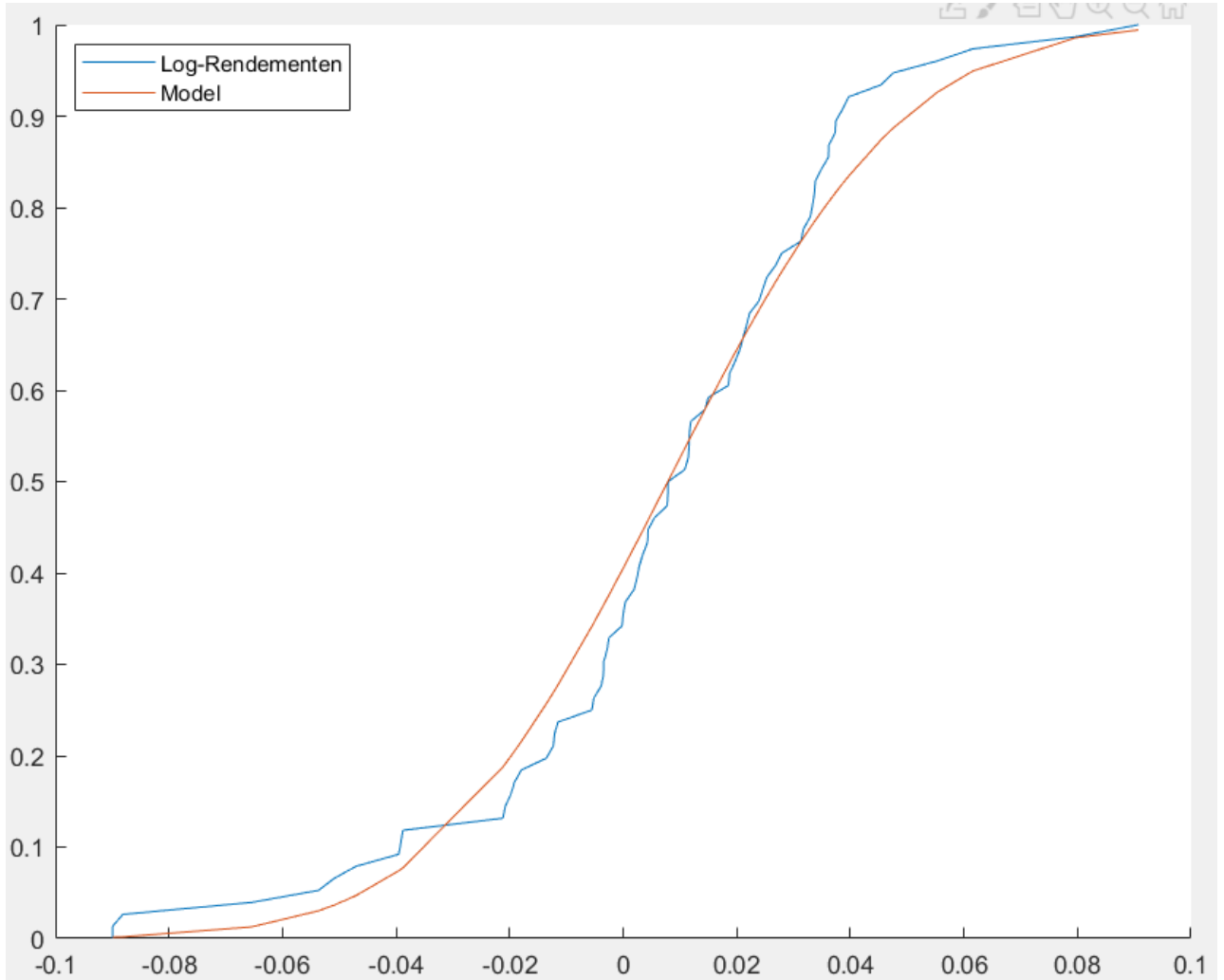




6 Opdracht 7

Vergelijk de empirische cumulatieve distributiefunctie van de log-rendementen van opeenvolgende maanden van VWRL met de cumulatieve distributiefunctie van het model, i.e., $N(\mu - \frac{1}{2}\sigma^2, \sigma^2)$. Als modelparameters μ en σ kies je de overeenkomstige populatiestatistieken die je in opdracht 4 hebt berekend. Hint: de functies `ecdf` en `cdf` kunnen nuttig zijn. Maak een figuur van deze twee cumulatieve distributiefuncties en neem deze op in het verslag. Kan je op basis van deze figuur besluiten dat de geobserveerde log-rendementen van VWRL gedurende de laatste 77 maanden inderdaad bij benadering voldoen een normale verdeling met gemiddelde $\mu - \frac{1}{2}\sigma^2$ en standaardafwijking σ .

Figuur van deze twee cumulatieve distributiefuncties:



Deze figuur laat zien dat de geobserveerde log-rendementen van VWRL gedurende de laatste 77 maanden inderdaad bij benadering voldoen aan normale verdeling.

7 Opdracht 8

Schrijf een functie met hoofding

`function [yield, invested, value, units] = SN_simulateFundInvestingPath(budget, pricePath, alpha)` die het periodiek beleggen in twee beursgenoteerde fondsen (bijvoorbeeld *EUN5* en *VWRL*) simuleert met de bijbehorende fiscaliteit (0.35% beurstaks en 6 euro transactiekosten voor elke aankoop) zoals hierboven beschreven.

Code voor deze opgave kan gevonden worden in 5.

8 Opdrach 9

Voer de code uit en neem de twee gegenereerde figuren op in het verslag.

Figuren gegenereerd door de code 6 kunnen gevonden worden op volgende pagina.

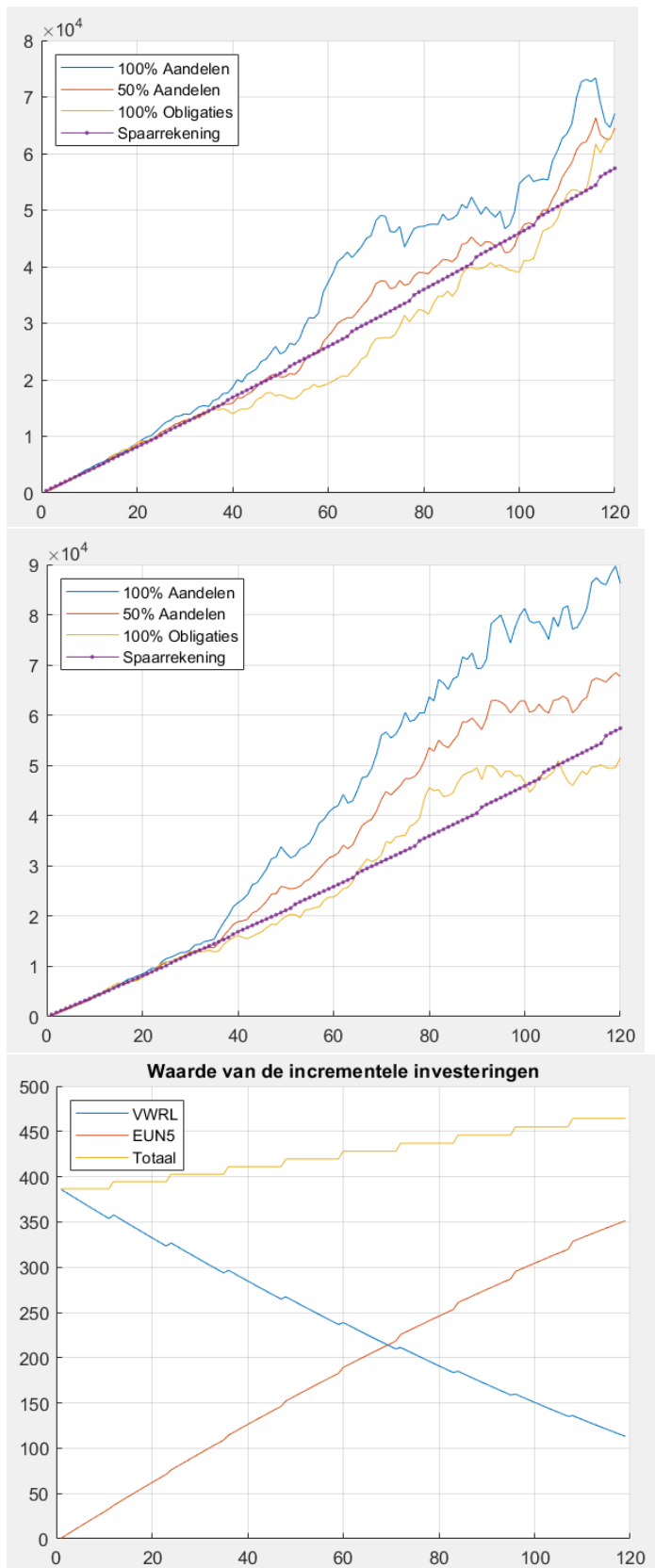


Figure 2: A boat.

9 Opdracht 10

Schrijf een functie met hoofding

```
function [yields, invested] = SN_simulateFundInvesting(budget, priceHistory,alpha, N)
```

die de functie uit opdracht 8 gebruikt om het periodiek beleggen in een fonds te simuleren met een Monte Carlo-simulatie. De startdag van de simulatie is steeds 1 januari.

Code voor deze opdracht is gegeven in 7.

10 Opdracht 11

We simuleren vervolgens het eindrendement voor enkele combinaties van de EUN5 en VWRL fondsen met behulp van de functie uit de voorgaande opdracht. Kies $n = 600$ maanden (50 jaar) bij een spaarbudget van e300 per maand. Voor de spaarrekening zullen we de gunstige basisrente van 2.0% hanteren. Als prijsgeschiedenis gebruiken we de laatste 60 maanden van VWRL en EUN5.

Hoeveel bedraagt het rendement (uitgedrukt in %) van de spaarrekening?

Spaarbudget	maanden	basisrente	Yield(relatieve)
300	600	2%	0.5236

Histogram op de volgende pagina laat zien waar de meeste punten liggen. De kans om een punt te krijgen in een interval waarbij punten dicht bij elkaar liggen, is groot. Dit kan verklaard worden aan de hand van betrouwbaarheidsinterval. Op de histogram zien wij dat de rendement altijd positief is, wat niet realistisch is.

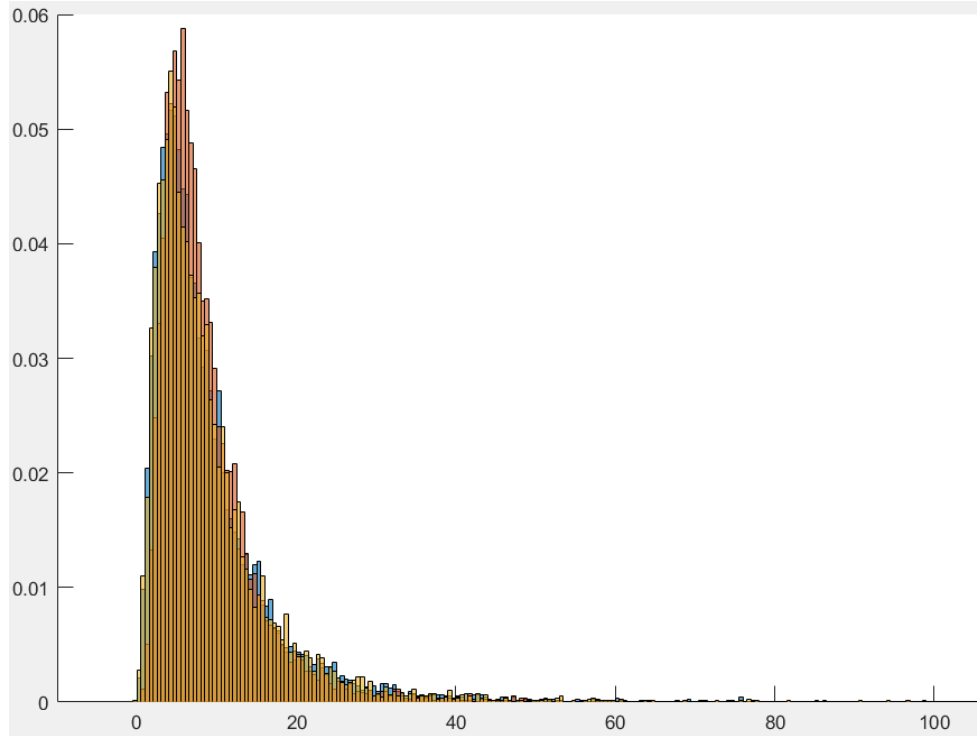


Figure 3: Histogram van simulateFuncInvesting: alphaAggressive,alphaBalanced en alphaDefensive.

11 Appendix

Listing 1: Code voor opdracht 1

```
function [yield,invested,value] = r0439731_simulateSavingInvesting(budget,rate,months)
    value = zeros(1,months);
    interests_per_month = zeros(1,12);
    yield = 0;
    invested = 0;
    rate_increment = 0.02;
    current_budget=budget;
    budget_per_month = current_budget;
    current_month = 1;
    index = 1;
    previous_debt = 0;
    interests_on_previous_debt = 0;
    for i = 1:months
        value(1,i) = budget_per_month;
        if(mod(i,13) == 0)
            if(i-13 > 0)
                previous_debt = previous_debt + value(1,(i-12));
                interests_on_previous_debt = (previous_debt*(rate/100));
            end
            previous_debt = interests_on_previous_debt +sum(interests_per_month);
            value(1,i) = value(1,i) + interests_on_previous_debt;
            budget_per_month = budget_per_month + previous_debt;
            interests_per_month = zeros(1,12);
            current_month = 1;
            index = 1;
        end
        if(mod(i,12) == 0)
            current_budget = current_budget+(rate_increment*current_budget);
        end
        interests_per_month(1,index) = (((13-current_month)/12)*(rate/100))*current_budget;
        budget_per_month = current_budget + budget_per_month;
        invested = invested + current_budget;
        current_month = current_month +1;
        index = index+1;
    end
    yield = (value(end)-invested)/invested;
end
```

Listing 2: Code voor opdracht 4

```
function [mu,sigma] = r0439731_estimateParameters(s)
    firstVector = s(1:length(s)-1);
    secondVector = s(2:length(s));
    result = log(secondVector ./firstVector);
    sigma = std(result);
    mu = mean(result) + 0.5*(sigma^2);
end
```

Listing 3: Code voor opdracht 5

```
function [path] = r0439731_simulateFundPath(initialPrice,mu,sigma,months)
    path = zeros(months,1);
    path(1) = initialPrice ;
    t = 1;
    for i = 2:months
        path(i) = path(i-1) * exp(((mu-0.5*(sigma^2))*t) + sigma*sqrt(t)*randn);
    end
end
```

Listing 4: Code voor opdracht 6

```
function simualte_mod()
load('Funds.mat');
[mu, sigma] = r0439731_estimateParameters(VWRL(:, 1));
length = 60;
initialPrice = 83.05;
initialPrice = initialPrice* 0.90;
fig = figure;
hold on;
for i = 1:10
    path = r0439731_simulateFundPath(initialPrice, mu, sigma, length);
    plot(path);
end
hold all ;
saveas(fig , 'ex4.png');
end
```

Listing 5: Code voor opdracht 8

```

function [yield, invested, value, units] = r0439731_simulateFundInvestingPath(budget, pricePath, alpha)
    [rows,cols] = size(pricePath);
    units = zeros(rows,cols);
    budget_per_month = budget;
    current_budget = budget_per_month;
    percentage_increment = 0.02;
    tax = 1.0035;
    invested = current_budget;
    transaction_costs = 6;
    if(((current_budget*alpha(1) -transaction_costs)/tax) >= 0))
        units(1,1) = (((current_budget*alpha(1) -transaction_costs)/tax)/pricePath(1,1));
        invested = invested + budget_per_month;
    elseif (((current_budget - transaction_costs)/tax) > 0)
        units(1,2) = (((current_budget - transaction_costs)/tax)/pricePath(1,2));
        invested = invested + budget_per_month;
    end
    for i = 2:rows
        if(((current_budget*alpha(i) -transaction_costs)/tax) >= 0) && (((current_budget*(1-alpha(i)) -
            units(i,1) = (((current_budget*alpha(i) -transaction_costs)/tax)/pricePath(i,1));
            units(i,2) = (((current_budget*(1-alpha(i)) - transaction_costs)/tax)/pricePath(i,2));
            invested = invested + budget_per_month;

        elseif(((current_budget*alpha(i) -transaction_costs)/tax) < 0) && (((current_budget*(1-alpha(i))
            units(i,2) = (((current_budget - transaction_costs)/tax))/pricePath(i,2));
            invested = invested + budget_per_month;

        elseif(((current_budget*alpha(i) -transaction_costs)/tax) >= 0) && (((current_budget*(1-alpha(i))
            units(i,1) = (((current_budget -transaction_costs)/tax))/pricePath(i,1);
            invested = invested + budget_per_month;
        else
            %     units(i,1) = units(i-1,1);
            %     units(i,2) = units(i-1,2);
        end
        if(mod(i,12) == 0)
            budget_per_month = budget_per_month + (budget_per_month*percentage_increment);
        end
        current_budget = budget_per_month;
    end
    first = cumsum(units(:,1));
    second = cumsum(units(:,2));
    P = [first,second];
    units = [first,second];
    value = P .* pricePath;

    total = sum(value(end,:));
    yield = total/invested -1;
end

```

Listing 6: Code voor opdracht 9

```

for jj = 1 : 2
    figure; hold all;
    weights = [1 0.5 0];
    pathVWRL = SN_simulateFundPath(37.78, 7.8578e-03, 3.2775e-02, 120);
    pathEUN5 = SN_simulateFundPath(128.25, 3.0856e-03, 3.0704e-02, 120);
P = [pathVWRL pathEUN5];
for k = 1 : 3
[~,~,value] = SN_simulateFundInvestingPath(400, P, weights(k)*ones(120,1));
plot( sum(value,2) )
end
[~, ~, value] = SN_simulateSavingInvesting(400, 2, 120);
plot(value, '-');
legend( 100 % Aandelen , 50 % Aandelen , 100 % Obligaties , Spaarrekening , ...
        Location , NorthWest )
grid on;
end

```

Listing 7: Code voor opdracht 10

```

function [yields, invested] = r0439731_simulateFundInvesting(budget, priceHistory, alpha, N)
    [mu,sigma] = r0439731_estimateParameters(priceHistory);
    [rows,cols] = size(alpha);
    yields = zeros(1,N);
    first = priceHistory(end,1);
    second = priceHistory(end,2);
    for i = 1 :N
        if(mod(i,100) == 0)
            disp("Iteration " + i);
        end
        pricePathA = r0439731_simulateFundPath(first,mu,sigma,rows);
        pricePathB = r0439731_simulateFundPath(second,mu,sigma,rows);
        [yield, invested, ~, ~] = r0439731_simulateFundInvestingPath(budget, [pricePathA pricePathB]);
        yields(i) = yield;
    end
end
end

```
