Portfolio calculation

Constants:

|  |  |
| --- | --- |
| **Variable name** | **Description** |
|  | Sum of expected savings |
|  | Target stock weight |
|  | Target stock weight |
|  | Initial stock target |
|  | Gearing cap |
|  | Risk free rate |
|  | Margin rate |
|  |  |
|  |  |

Period 0 primo flow:

|  |  |
| --- | --- |
| **Variable name** | **Description** |
|  | First savings |
|  | Cash |
|  | New debt |
|  | New investments primo |
|  | Total debt |
|  | Portfolio value primo |
|  | Empirical stock/cash ratio |
|  |  |

Period 0 ultimo flow:

|  |  |
| --- | --- |
| **Variable name** | **Description** |
|  | Calculate interest |
|  | Market return |
|  | Portfolio value ultimo |
|  | Total value ultimo |
|  | Dollar stock target |
|  | Investment phase |
|  |  |

Period t >= 1 primo flow:

|  |  |
| --- | --- |
| **Variable name** | **Description** |
|  | Savings |
|  | Apply interest rate to cash |
|  |  |
|  | Calculate new debt |
|  | Total debt |
|  | Calculate cash contributions |
|  | New investments primo |
|  | Portfolio value primo |

Period t >= 1 ultimo flow:

|  |  |
| --- | --- |
| **Variable name** | **Description** |
|  | Calculate interest |
|  | Market return |
|  | Portfolio value ultimo |
|  | Total value ultimo |
|  | Empirical stock/cash ratio |
|  | Dollar stock target |
|  | Phase |

function nd(s, pv\_u, dst, td, cash, phase, g):

# determine new debt dependent on phase

# dont aquire debt or repay debt in phase 3 or 4

if phase == 3 or phase == 4:

return 0, s

if phase == 1:

# only phase where we add debt

new\_debt = min(g\*s, dst-pv\_u-s)

return new\_debt, s

if phase == 2:

stocks\_sold = max(dst-pv\_u, 0)

debt\_repayment = min(td, s + stocks\_sold)

leftover\_savings = max(s - debt\_repayment- stocks\_sold, 0)

return -debt\_repayment, leftover\_savings

function update\_cash(phase, pv\_u, tv\_u, s, cash, pi\_rf, pi\_hat):

# determine new cash dependent on phase

# dont aquire cash in phase 1-3

if phase <= 3:

return 0, s

if phase == 4:

desired\_cash = (pi\_hat-pi\_rf)\*(tv\_u+s)

desired\_savings = (pi\_rf-pi\_hat)\*(tv\_u+s)

return (cash + desired\_cash), desired\_savings

function phase\_check(phase, pi\_rf, pi\_rm, pi\_hat, td):

if phase == 4:

return 4

if td > 0:

#has target not been reached?

if pi\_hat < pi\_rm and phase <= 1:

return 1

else:

return 2

#if target has been reached and no debt remains

#is the value still above the target?

if pi\_hat < pi\_rf:

return 3

else:

return 4

function calc\_pi(gamma, sigma, mr, rate, cost = 0):

return (mr - cost - rate)/(gamma \* sigma)

function phase\_check(pv\_u, dst, phase, td):

if phase == 4:

return 4

#has target not been reached?

if pv\_u < dst and phase <= 2:

return 1

#any remaining debt?

if td > 0 and pv\_u >= dst:

return 2

#if target has been reached and no debt remains

#is the value still above the target?

If pv\_u > dst:

return 3

else:

return 4