<https://www.matheboard.de/archive/2495/thread.html>

**Annahme:**

*TotalNeeded\_max* := Der Vermögensbetrag der über die Rentenphase (n Jahre) gebraucht wird um komfortabel zu leben

*TotalNeeded\_min* := Der Vermögensbetrag der über die Rentenphase (n Jahre) gebraucht minimal gebraucht wird.

**Fragestellung:**

Welchen Beträge *Total\_Cash* und *Total\_Stocks* werden am Anfang des Rentenalters benötigt um

* Jährlich den Betrag *TotalNeeded\_max / n* zur Verfügung zu haben, unter der Annahme dass sich der Aktienmarkt normal positive entwickelt
* Jährlich den Betrag *TotalNeeded\_min / n* zur Verfügung zu haben, unter der Annahme dass der Aktienmarkt um einen gegebenen Faktor zu Beginn des Renteneintritts eintritt und sich nicht mehr erholt

Und wie hoch sind die jährlichen Entnahmeraten *Rate\_Cash, Rate\_Stocks\_Max, Rate\_Stocks\_Max*?

Anmerkung:

Die Cash Rate ist immer gleich da nur der Anteil der Aktien an den konstanten oder einbrechenden Aktienmarkt anpassen muss.

Die jährliche Rate wird mit einem Faktor multipliziert der sich aus der Sparkassenformel ergibt:

Kn = K0 \* q^n – R \* (q^n -1)/(q-1)

Kn ist 0, da von Kapitalverzehr ausgegangen wird.

K0 ist das Anfangskapital, R die jährliche Entnahmerate.

Die Formal muss nach K0 aufgelöst werden so dass

K0 = R \* z

gilt.

Z ist der Faktor der bei n Jahren Laufzeit den Zinseszins beinhaltet. Angenommen n=13 und q=1.0, d.h. das Vermögen wächst garnicht, dann wäre z=n=13, d.h. man würde einfach das Anfangskapital durch die Anzahl der Jahre teilen. Je höher der Zinssatz desto niedriger der Faktor. Bei 8% Zinsen und q=1.08 wäre z bei ca 7, d.h. man kann durch den Zinseszins eine (hohe) Rate von K0/7 statt K0/13 pro Jahr abheben.

Z = -(p^n - 1) / ((p-1) \* (p^n)));

Berechnen mit sympy (<https://live.sympy.org/>)

Kn, K0, q, n = symbols('Kn K0 q n')

eq1 = Eq(K0 \* q\*\*n - (K0/n)\*(q\*\*n-1)/(q-1), Kn)

sol = solve(eq1,K0)

Damit kann man nun die Zinsfaktoren

z\_stocks\_max

z\_stocks\_min

z\_cash

vorberechnen.

K0 = R \* -(q\*\*n-1) / ( (q-1) \* (q\*\*n) )

K0 = R\_cash \* z\_cash\_max

K0 = R\_stocks \* z\_stocks\_max

K0 = R\_cash \* z\_cash\_min

K0 = (R\_stocks \* z\_stocks\_min) / CrashFactor

z\_stocks\_max= -( q\_stocks\_max \*\*n-1) / (( q\_stocks\_max -1)\*( q\_stocks\_max \*\*n))

z\_stocks\_min= -( q\_stocks\_min \*\*n-1) / (( q\_stocks\_min -1)\*( q\_stocks\_min \*\*n))

z\_cash = -( q\_cash \*\*n-1) / (( q\_cash -1)\*( q\_cash \*\*n))

Für die finale Raten kann man nun ein Gleichungssystem mit 5 Gleichung und 5 Unbekannten aufstellen:

1. Rate\_Cash \* z\_cash + Rate\_Stocks\_Max \* z\_stocks\_max = Total\_Cash + Total\_Stocks
2. Rate\_Cash \* z\_cash + Rate\_Stocks\_Min \* z\_stocks\_min = Total\_Cash + (Total\_Stocks\* stocks\_crashFactor\_badCase)
3. Total\_Cash = Rate\_Cash \* z\_cash
4. Rate\_Cash + Rate\_Stocks\_Max = comfort\_total\_needed\_Year
5. Rate\_Cash + Rate\_Stocks\_Min = minimum\_total\_needed\_Year

Mit sympy:

Rate\_Cash, z\_cash, Rate\_Stocks\_Max, Rate\_Stocks\_Min, z\_stocks\_max, z\_stocks\_min, Total\_Cash, Total\_Stocks, stocks\_crashFactor\_badCase, comfort\_total\_needed\_Year, minimum\_total\_needed\_Year = symbols('Rate\_Cash z\_cash Rate\_Stocks\_Max Rate\_Stocks\_Min z\_stocks\_max z\_stocks\_min Total\_Cash Total\_Stocks stocks\_crashFactor\_badCase comfort\_total\_needed\_Year minimum\_total\_needed\_Year ')

eq1=Eq(Rate\_Cash \* z\_cash + Rate\_Stocks\_Max \* z\_stocks\_max, Total\_Cash + Total\_Stocks)

eq2=Eq(Rate\_Cash \* z\_cash + Rate\_Stocks\_Min \* z\_stocks\_min, Total\_Cash + (Total\_Stocks\* stocks\_crashFactor\_badCase))

eq3=Eq(Total\_Cash, Rate\_Cash \* z\_cash)

eq4=Eq(Rate\_Cash + Rate\_Stocks\_Max, comfort\_total\_needed\_Year)

eq5=Eq(Rate\_Cash + Rate\_Stocks\_Min, minimum\_total\_needed\_Year)

sol = solve((eq1, eq2, eq3, eq4, eq5),( Rate\_Cash, Rate\_Stocks\_Min, Rate\_Stocks\_Max, Total\_Cash, Total\_Stocks))

Ergebnis:

Rate\_Cash: (comfort\_total\_needed\_Year\*stocks\_crashFactor\_badCase\*z\_stocks\_max - minimum\_total\_needed\_Year\*z\_stocks\_min)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Rate\_Stocks\_Max: z\_stocks\_min\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Rate\_Stocks\_Min: stocks\_crashFactor\_badCase\*z\_stocks\_max\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Total\_Cash: z\_cash\*(comfort\_total\_needed\_Year\*stocks\_crashFactor\_badCase\*z\_stocks\_max - minimum\_total\_needed\_Year\*z\_stocks\_min)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min),

Total\_Stocks: z\_stocks\_max\*z\_stocks\_min\*(-comfort\_total\_needed\_Year + minimum\_total\_needed\_Year)/(stocks\_crashFactor\_badCase\*z\_stocks\_max - z\_stocks\_min)}