Maßgeschneiderte Schadenmodelle mit der stückweisen Pareto-Verteilung

Dr. Ulrich Riegel

DAV
DEUTSCHE AKTUARVEREINIGUNG e.V.

ASTIN-Tagung, 30. April 2021

Agenda

1. Layerunabhängige Schadenmodelle
2. Pareto-Verteilung
3. Stückweise Pareto-Verteilung
4. PML-Kurven
5. Layerunabhängige Modelle für Sequenzen von Layern

Layerunabhängige Schadenmodelle

Wir betrachten Layer c_i xs a_i $(i = 1, \dots, n)$, z.B. die Layer eines nichtproportionalen Rückversicherungsprogramms oder künstliche Referenz-/Kalkulationslayer.

Wir setzen voraus, dass sich der Aktuar bei jedem Layer c_i xs a_i bereits für einen Schätzer $\widehat{e}_i$ des erwarteten Schadens entschieden hat.

Definition: Ein kollektives Modell heißt *layerunabhängig*, wenn es für jeden Layer c_i xs a_i den erwarteten Schaden $\widehat{e}_i$ liefert. Wenn das Modell nur für einen/einige (aber nicht für alle) Layer den erwarteten Schaden trifft, dann heißt es *layerspezifisch*.

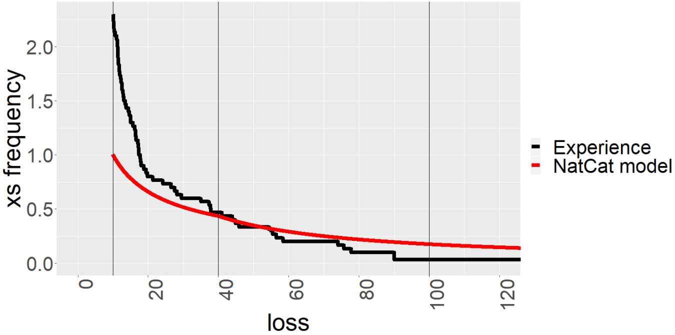
Warum sind layerunabhängige Modelle nützlich?

Wir betrachten zwei Beispiele!

Beispiel 1: NatCat Aggregate XL

Wir betrachten einen Cat-XL 90m xs 10m mit AAD 100m und AAL 100m für Hurrikan.

Unterhalb von 40m scheint das NatCat-Modell die Frequenz der Ereignisse zu unterschätzen:



Beispiel 1: NatCat Aggregate XL

Das NatCat-Modell scheint für den kompletten Layer 90m xs 10m zu billig zu sein. Daher teilen wir den Layer in zwei künstliche Teillayer auf, die wir unabhängig modellieren:

- Für den Layer 30m xs 10m verwenden wir das Burning Cost-Modell (kollektives Modell mit Poisson-Schadenzahl und der empirischen Schadenhöhenverteilung)
- Für den Layer 60m xs 40m verwenden wir das NatCat-Modell (ebenfalls ein kollektives Modell mit Poisson-Schadenzahl)

	NatCat-Modell	Geteiltes Modell
Erw. Schaden 30m xs 10m	18.53m	26.66m
Erw. Schaden 60m xs 40m	15.95m	15.95m
Erw. Schaden 90m xs 10m	34.48m	42.61m
Erw. Schaden nach AAD & AAL	3.68m	2.28m

Beispiel 1: NatCat Aggregate XL

Wie erwartet liefert das geteilte Modell einen höheren erwarteten Schaden im Layer 90m xs 10m. Aber warum erhalten wir einen nach AAD & AAL eine niedrigere erwartete Schadenlast?

Begründung:

- Der erwartete Schaden nach AAD & AAL hängt nicht nur vom erwarteten Schaden im Layer 90m xs 10m ab sondern auch ganz wesentlich von der Volatilität!
- Die Verwendung von unabhängigen Modellen für die künstlichen Teillayer ignoriert die Tatsache, dass die beiden Teillayer positiv korreliert sind. Dadurch wird die Schwankung unterschätzt.

$\Rightarrow$ Wir brauchen ein layerunabhängiges Modell, das für die beiden Teillayer jeweils den richtigen erwarteten Schaden liefert.

Beispiel 2: Motor-Marktquotierung

Das Ergebnis einer Marktquotierung ist oft eine Tabelle mit Raten für eine Folge von kurzen Layern:

Layer	Exp. Loss
1 xs 1	2,93%
1 xs 2	1,30%
1 xs 3	0,72%
1 xs 4	0,43%
1 xs 5	0,24%
1 xs 6	0,12%
1 xs 7	0,07%
1 xs 8	0,04%

Layer	Exp. Loss
1 xs 9	0,03%
1 xs 10	0,02%
1 xs 11	0,02%
1 xs 12	0,02%
1 xs 13	0,01%
1 xs 14	0,01%
Inf xs 15	0,12%

Hier ist es sehr praktisch, wenn man ein layerunabhängiges Marktmodell in den Pricing-Systemen zur Verfügung stellen kann!

Agenda

Definition Pareto-Verteilung

Bevor wir zur stückweisen Pareto-Verteilung kommen benötigen wir ein paar Tatsachen zur Pareto-Verteilung.

Definition: Seien $t > 0$ und $\alpha > 0$. Die *Pareto-Verteilung* $\mathrm{Pareto}(t, \alpha)$ hat die Verteilungsfunktion

$$F_{t,\alpha}(x) := \begin{cases} 0 & \text{für } x \leq t \\ 1 - \left(\frac{t}{x}\right)^{\alpha} & \text{für } x > t. \end{cases}$$

Diese Version der Pareto-Verteilung wird auch als *Pareto type I*, *European Pareto* oder *single-parameter Pareto* bezeichnet.

Eigenschaften der Pareto-Verteilung

Warum ist die Pareto-Verteilung bei Rückversicherern so beliebt?

Proposition:

a) Sei $X \sim \text{Pareto}(t, \alpha)$. Dann ist $cX \sim \text{Pareto}(ct, \alpha)$ für alle $c > 0$

b) Sei $X \sim \text{Pareto}(t_1, \alpha)$. Für $t_2 > t_1$ gilt dann $X|(X > t_2) \sim \text{Pareto}(t_2, \alpha)$

Folgerung:

- Das *Pareto-Alpha* α ist invariant unter Skalierung sowie Trunkierung von links
- Für Layer und Thresholds oberhalb t hängt das Verhältnis zwischen erwarteten Layerschäden und/oder xs-Frequenzen nur von α (und nicht von t) ab

R-Paket Pareto

Wir betrachten nun einige nützliche Konzepte wie die *Pareto-Extrapolation* und das *Pareto-Alpha zwischen zwei Layern*.

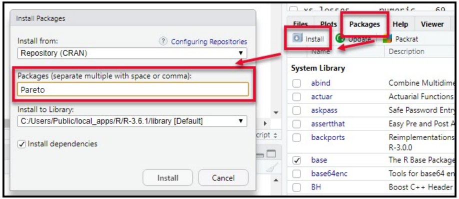
Zur Illustration verwenden wir das R-Paket `Pareto`, das Funktionen für die Pareto- und die stückweise Pareto-Verteilung bereitstellt.

Das Paket ist auf CRAN verfügbar:

Console:

```
> install.packages("Pareto")
```

In R Studio:



Mittlere Schadenhöhe im Layer

Mittlere Schadenhöhe im Layer: Seien $X \sim \text{Pareto}(t, \alpha)$ und $a, c \geq 0$. Dann ist

$$\text{E}(\min[c, \max(X - a, 0)]) = \int_a^{c+a} (1 - F_{t,\alpha}(x))\, dx$$

die mittlere Schadenhöhe im Layer c xs a.

Beispiel: $t = 1000$, $\alpha = 2$, Layer 4000 xs 1000

```
> library(Pareto)
> Pareto_Layer_Mean(4000, 1000, 2)      # Parameter t can be skipped since t = 1000
[1] 800
```

Für $t = 5000$ gibt es im Layer nur Totalschäden:

```
> Pareto_Layer_Mean(4000, 1000, 2, t = 5000)
[1] 4000
```

Pareto-Extrapolation

Wir betrachten zwei Layer c_i xs a_i und eine Pareto(t, α)-verteilte Schadenhöhe mit ausreichend kleinem t. Was ist der erwartete Schaden im Layer c_2 xs a_2 wenn der erwartete Schaden des Layers c_1 xs a_1 gegeben ist?

Beispiel:

Der erw. Schaden in 4000 xs 1000 sei 500. Wie hoch ist der erw. Schaden im Layer 5000 xs 5000 unter der Annahme $\alpha = 2$?

```
> Pareto_Extrapolation(4000, 1000, 5000, 5000, 2) * 500
[1] 62.5
> Pareto_Extrapolation(4000, 1000, 5000, 5000, 2, ExpLoss_1 = 500)
[1] 62.5
```

Pareto-Alpha zwischen zwei Layern

Ist der erwartete Schaden von zwei Layern gegeben, dann gibt es i.d.R. ein eindeutiges Pareto-Alpha α das konsistent zu dem Verhältnis der beiden Layer-Erwartungen ist.

Beispiel:

Erwarteter Schaden im Layer 4000 xs 1000: 500
Erwarteter Schaden im Layer 5000 xs 5000: 62,5.

Alpha zwischen den Layern:

```
> Pareto_Find_Alpha_btw_Layers(4000, 1000, 500, 5000, 5000, 62.5)
[1] 2
```

Probe: siehe vorhergehende Folie

Pareto-Alpha zwischen Frequenz und Layer

Gegeben seien die erwartete Frequenz xs einem Threshold und der erwartete Schaden eines Layers. Dann gibt es typischerweise ein eindeutiges α, das konsistent mit diesen Daten ist.

Beispiel:	Erwartete Frequenz xs 500:	3,0
	Erwarteter Schaden im Layer 4000 xs 1000:	600

Alpha zwischen Frequenz und Layer:

```
> Pareto_Find_Alpha_btw_FQ_Layer(500, 3, 4000, 1000, 600)
[1] 2
```

Probe:

```
> Pareto_Layer_Mean(4000, 1000, 2, t = 500) * 3
[1] 600
```

Fitten der Schadenerwartung von zwei Layern

Diese Methoden können verwendet werden um ein Poisson-Pareto-Modell zu fitten, das den erwarteten Schaden von zwei Layern trifft.

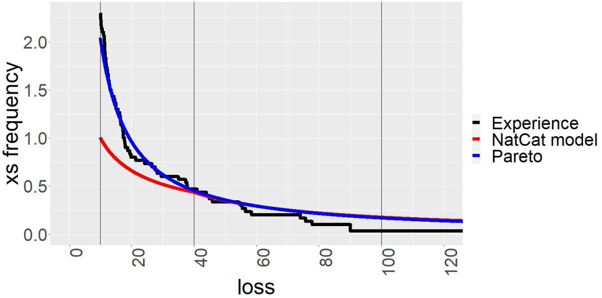
Beispiel 1 von oben (Aggregate XL):

Erw. Schaden des Layers 30 xs 10 (Burning Cost):	26.66
Erw. Schaden des Layers 60 xs 40 (NatCat-Modell):	15.95

```
> Pareto_Find_Alpha_btw_Layers(30, 10, 26.66, 60, 40, 15.95)
[1] 1.086263
> # Frequency @ 10:
> 26.66 / Pareto_Layer_Mean(30, 10, 1.086263)
[1] 2.040392
```

Fitten der Schadenerwartung von zwei Layern

Ein kollektives Modell $\sum_{n=1}^{N} X_n$ mit $X_N \sim \text{Pareto}(10, 1.09)$ und $N \sim \text{Poisson}(2.04)$ trifft die Schadenerwartung beider Layer.



Passt ganz gut!

Fitten der Schadenerwartung von zwei Layern

Welchen Effekt hat die Verwendung dieses Fits für den Aggregate XL?

	NatCat-Modell	Geteiltes Modell	Pareto-Modell
Erw. Schaden 30 xs 10	18.53	26.66	26.66
Erw. Schaden 60 xs 40	15.95	15.95	15.95
Erw. Schaden 90 xs 10	34.48	42.61	42.61
Erw. Schaden nach AAD & AAL	3.68	2.28	4.78

Frequenzextrapolation und Pareto-Alpha zwischen Frequenzen

Ist die xs-Frequenz f_1 bei t_1 gegeben, so kann die xs-Frequenz f_2 bei t_2 direkt berechnet werden:

$$f_2 = f_1 \cdot \left(\frac{t_1}{t_2}\right)^{\alpha}$$

Anders herum können wir das Pareto-Alpha berechnen, wenn die beiden Frequenzen f_1 und f_2 bei t_1 und t_2 gegeben sind:

$$\alpha = \frac{\log(f_2/f_1)}{\log(t_1/t_2)}.$$

Agenda

Definition der stückweisen Pareto-Verteilung

Definition:

Seien $\mathbf{t} := (t_1, \ldots, t_n)$ ein Vektor von Thresholds mit $0 < t_1 < \cdots < t_n < t_{n+1} := +\infty$ und $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_n)$ ein Vektor von Pareto-Alphas mit $\alpha_i \geq 0$ and $\alpha_n > 0$. Die *stückweise Pareto-Verteilung* $\text{Pareto}(\mathbf{t}, \boldsymbol{\alpha})$ hat die Verteilungsfunktion

$$F_{\mathbf{t},\boldsymbol{\alpha}}(x) := \begin{cases} 0 & \text{für } x < t_1 \\ 1 - \left(\frac{t_k}{x}\right)^{\alpha_k} \prod_{i=1}^{k-1} \left(\frac{t_i}{t_{i+1}}\right)^{\alpha_i} & \text{für } x \in [t_k, t_{k+1}). \end{cases}$$

Flexibilität der stückweisen Pareto-Verteilung

Die Familie der stückweisen Pareto-Verteilungen ist sehr flexibel:

Proposition: Die Menge der stückweisen Pareto-Verteilungen ist dicht im Raum der positivwertigen Verteilungen (bezüglich der Lévy-Metrik).

Anmerkungen:

- Man kann also jede positivwertige Verteilung beliebig genau durch eine stückweise Pareto-Verteilung approximieren.
- Eine sehr gute Approximation erfordert jedoch typischerweise viele Pareto-Stücke.
- Die stückweise Pareto-Verteilung ist eine gute Alternative zu diskreten Verteilungen, da sie angenehmer in der Handhabung ist.

R-Paket Pareto

Das R-Paket `Pareto` stellt Funktionen für die stückweise Pareto-Verteilung zur Verfügung, z.B.

Mittlere Schadenhöhe im Layer:

```
> t <- c(1000, 2000, 3000, 4000)
> alpha <- c(2, 1, 3, 20)
> PiecewisePareto_Layer_Mean(4000, 1000, t, alpha)
[1] 826.6969
```

Layer-Varianz:

```
> PiecewisePareto_Layer_Var(4000, 1000, t, alpha)
[1] 922221.2
```

Simulation:

```
> losses <- rPiecewisePareto(1000, t, alpha)
```

R-Paket Pareto

Maximum Likelihood-Schätzer für die Pareto-Alphas:

```
> PiecewisePareto_ML_Estimator_Alpha(losses, t)
[1]  2.1363325  0.9422449  3.3452221 16.8548997
```

Verteilungsfunktion:

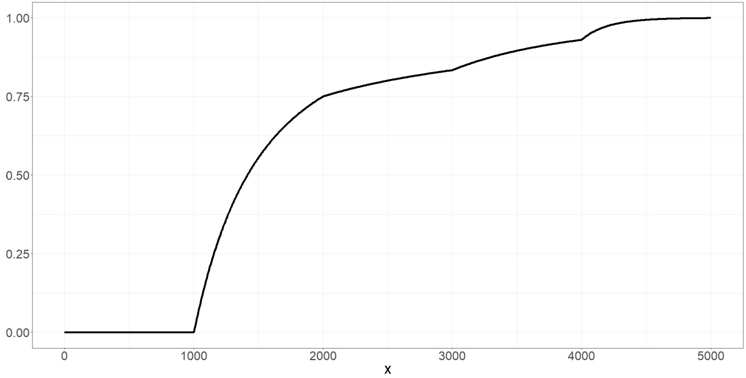
```
> x <- 1:10 * 1000
> pPiecewisePareto(x, t, alpha)
[1] 0.0000000 0.7500000 0.8333333 0.9296875 0.9991894 0.9999789
[7] 0.9999990 0.9999999 1.0000000 1.0000000
```

Dichtefunktion:

```
> dPiecewisePareto(x, t, alpha)
[1] 0.000000e+00 1.250000e-04 1.666667e-04 3.515625e-04 3.242592e-06
[6] 7.048328e-08 2.768239e-09 1.676381e-10 1.413089e-11 1.546188e-12
```

R-Paket Pareto

Verteilungsfunktion:



Agenda

PML-Kurven

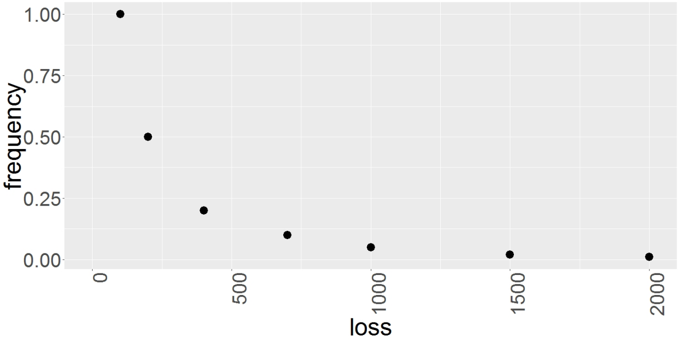
Bei Naturgefahren arbeitet man häufig mit PML-Kurven, z.B.

Wiederkehrperiode	Frequenz	Schadenhöhe
1	1.000	100
2	0.500	200
5	0.200	400
10	0.100	700
20	0.050	1000
50	0.020	1500
100	0.010	2000

Wie kann man aus einer solchen Kurve ein kollektives Modell ableiten?

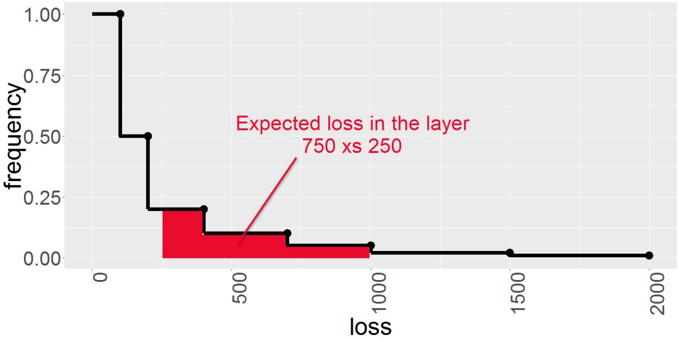
Interpolation von PML-Kurven

Wie interpoliert man am besten zwischen den gegebene Punkten?



Interpolation von PML-Kurven

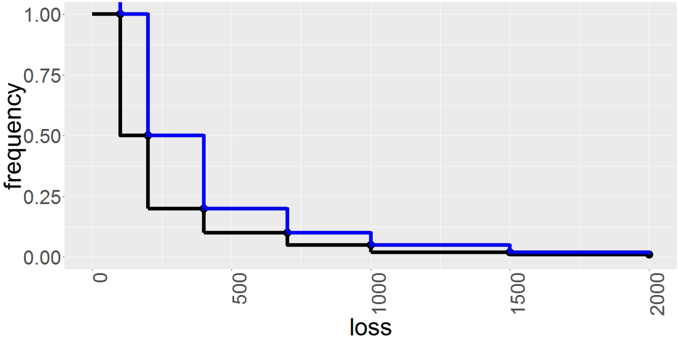
Eine Möglichkeit:



Das ist ziemlich billig, wenn man nur wenige Punkte hat . . .

Interpolation von PML-Kurven

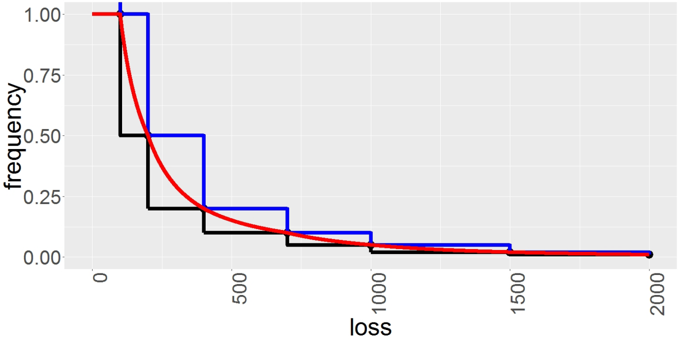
Alternativ könnte man folgendermaßen interpolieren:



Das ist ziemlich teuer, wenn man nur wenige Punkte hat . . .

Interpolation von PML-Kurven

Besserer Ansatz:



Was haben wir gemacht?

Interpolation von PML-Kurven

Seien $f_1 > \cdots > f_n > 0$ die xs-Frequenzen und $0 < t_1 < \cdots < t_n$ die zugehörigen Schadenhöhen aus der PML-Kurve.

Für $i = 1, \ldots, n-1$ seien

$$\alpha_i = \frac{\log(f_{i+1}/f_i)}{\log(t_i/t_{i+1})}$$

die Pareto-Alphas zwischen (t_i, f_i) und (t_{i+1}, f_{i+1}). Außerdem wählen wir ein $\alpha_n > 0$.

Seien $\mathbf{t} := (t_1, \ldots, t_n)$ und $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_n)$.

Dann erhält man ein kollektives Modell $\sum_{n=1}^{N} X_n$ mit der gewünschten Eigenschaft, indem man N mit $E(N) = f_1$ und $X_n \sim \text{Pareto}(\mathbf{t}, \boldsymbol{\alpha})$ wählt.

Interpolation von PML-Kurven

Bemerkung:

Mit dem gewählten α_n wird die PML-Kurve oberhalb des letzten Punktes t_n extrapoliert. Analog kann man auch ein α_0 wählen und unterhalb von t_1 extrapolieren. In den extrapolierten Bereichen ist das Modell jedoch nur mit großer Vorsicht zu verwenden!

Beispiel:

```
> return_period <- c(1, 2, 5, 10, 20, 50, 100)
> amount <- c(100, 200, 400, 700, 1000, 1500, 2000)
> Fit <- Fit_PML_Curve(return_period, amount)
> 1 / Excess_Frequency(Fit, amount)
[1]   1   2   5  10  20  50 100
```

Agenda

Layerunabhängige Schadenmodelle

Wir haben gesehen, dass man mit der Pareto-Verteilung ein kollektives Modell fitten kann, dass die erwarteten Schäden von zwei Layern trifft.

Die stückweise Pareto-Verteilung kann verwendet werden,

- wenn man die Layererwartungswerte für mehr als zwei Layer reproduzieren möchten und/oder
- wenn man zusätzlich gegebene xs-Frequenzen in den Fit einbeziehen möchte.

Beachte: Bei Integralfranchisen ist es sehr wichtig, dass das Modell eine realistische xs-Frequenz aufweist. Integralfranchisen werden häufig bei der Separierung von Basis- und Cat- bzw. Großschäden verwendet.

Einfacher Fitting-Algorithmus

Wir betrachten eine Sequenz $0 < a_1 < \cdots < a_n < a_{n+1} := +\infty$ von Prioritäten. Sei $c_i := a_{i+1} - a_i$ und sei e_i der erw. Schaden des Layers c_i xs a_i. Außerdem sei f_1 die erwartete xs-Frequenz bei a_1.

Diese Daten kann man mit der stückweisen Pareto-Verteilung leicht fitten:

- Berechne das Pareto-Alpha α_1 zwischen f_1 und c_1 xs a_1
- Berechne die xs-Frequenz f_2 bei a_2: $\quad f_2 := (a_1/a_2)^{\alpha_1} \cdot f_1$
- Berechne das Pareto-Alpha α_2 zwischen f_2 und c_2 xs a_2
- Berechne die xs-Frequenz f_3 bei a_3: $\quad f_3 := (a_2/a_3)^{\alpha_2} \cdot f_2$
- ...
- Verwende ein kollektives Modell $\sum_{n=1}^{N} X_n$ mit $E(N) = f_1$ und $X_n \sim \text{Pareto}(\mathbf{t}, \boldsymbol{\alpha})$.

Einfacher Fitting-Algorithmus

Einfache R-Implementierung:

```
simple_matching_approach <- function(a, e, f_1) {
  n <- length(a)
  c <- c(diff(a), Inf)
  f <- numeric(n)
  f[1] <- f_1
  alpha <- numeric(n)
  for (i in seq_along(a)) {
    alpha[i] <- Pareto_Find_Alpha_btw_FQ_Layer(a[i], f[i], c[i], a[i], e[i])
    if (i < n) {
      f[i+1] <- f[i] * (a[i] / a[i+1])^alpha[i]
    }
  }
  return(list(frequency = f_1, t = a, alpha = alpha))
}
```

Einfacher Fitting-Algorithmus

Beispiel:

```
> a <- c(1000, 1500, 2000, 2500)
> e <- c(100, 90, 50, 40)
> f_1 <- 0.21
> simple_matching_approach(a, e, f_1)
$frequency
[1] 0.21

$t
[1] 1000 1500 2000 2500

$alpha
[1] 0.2270887 0.4157705 5.0346512 4.4534806
```

Einfacher Fitting-Algorithmus

Man kann folgende Aussage zeigen:

Proposition:

Für $n \leq 3$ ist es immer möglich eine xs-Frequenz $f_1 > e_1/c_1$ zu finden, so dass der einfache Fitting-Algorithmus ein kollektives Modell liefert, dessen erw. Schaden für den Layer c_i xs a_i gleich e_i ist $(i = 1, \ldots, n)$.

Einfacher Fitting-Algorithmus

Die schlechte Nachricht: Der einfach Fitting-Algorithmus funktioniert oft nicht, wenn wir vier oder mehr Layer haben!

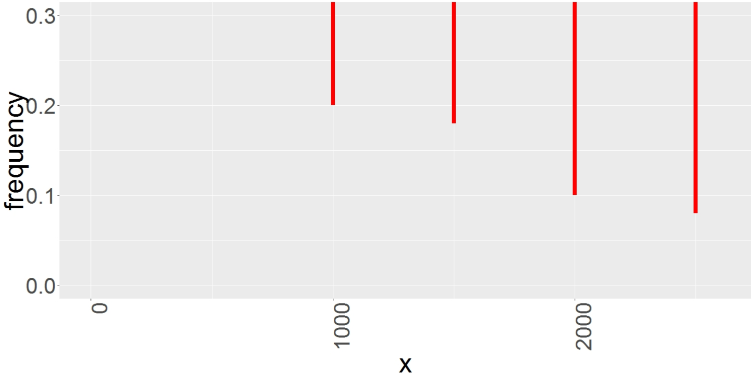
Zum Beispiel kann man folgende Layer-Sequenz nicht mit diesem Ansatz fitten:

i	Haftung c_i	Priorität a_i	Erw. Schaden e_i	Rate on Line e_i/c_i
1	500	1.000	100	0,20
2	500	1.500	90	0,18
3	500	2.000	50	0,10
4	500	2.500	40	0,08

Wenn man mit f_1 anfängt, so dass man die ersten drei Layererwartungen trifft, so bekommt man immer $f_4 < 0{,}06$, was kleiner als das Rate on Line des vierten Layers ist ...

Einfacher Fitting-Algorithmus

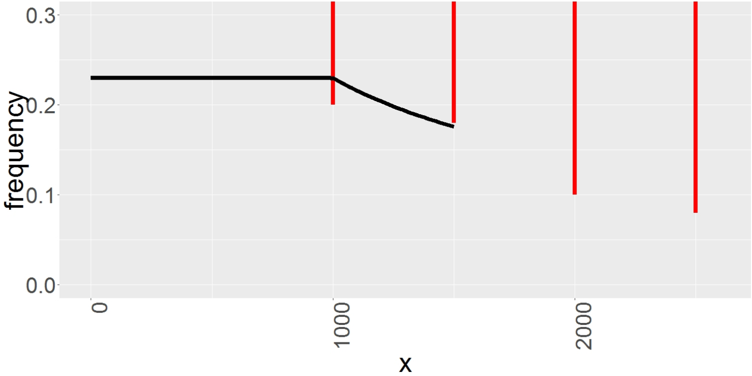
Die roten Linien illustrieren die zulässigen Eintrittsfrequenzen für die Layer an. Die Frequenz-Kurve muss durch die roten Linien gehen!



Einfacher Fitting-Algorithmus

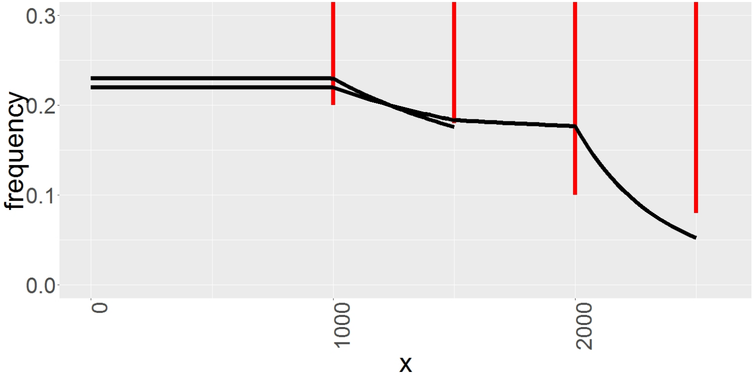
Mit $f_1 = 0{,}23$ ist f_2 kleiner als das Rate on Line des zweiten Layers ...

$\Rightarrow$ Wir müssen f_1 reduzieren!



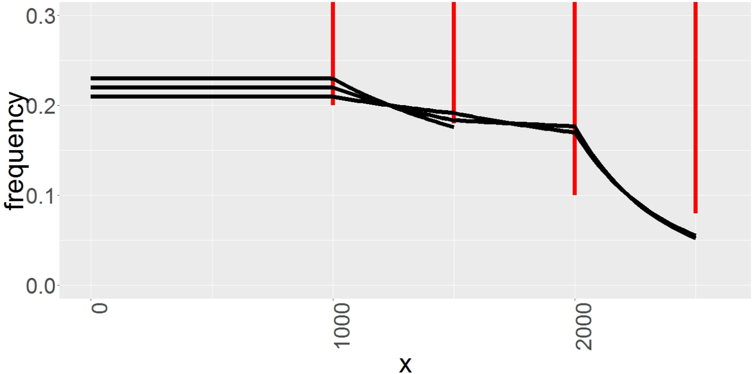
Einfacher Fitting-Algorithmus

Mit $f_1 = 0{,}22$ fittet der Algorithm die ersten drei Layer, aber f_4 ist kleiner als das Rate on Line des vierten Layers ...



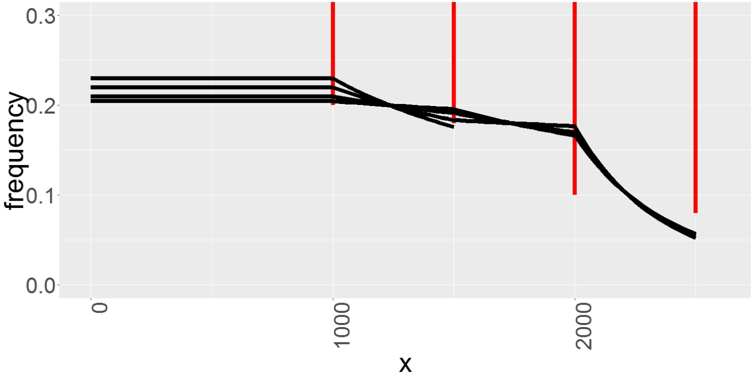
Einfacher Fitting-Algorithmus

Weitere Reduktion von $f_1 > 0{,}2$ bringt nur einen kleinen Anstieg von f_4.
$f_1 = 0{,}21$:



Einfacher Fitting-Algorithmus

Weitere Reduktion von $f_1 > 0{,}2$ bringt nur einen kleinen Anstieg von f_4.
$f_1 = 0{,}205$:

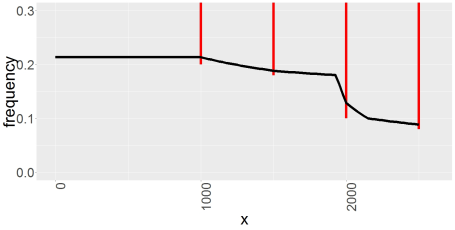


Verbesserter Fitting-Algorithmus

Es gibt einen besseren Fitting-Algorithmus, der zwei Pareto-Stücke pro Layer verwendet:

- Funktioniert zuverlässig für beliebig viele Layer
- Man kann zusätzlich Eintrittsfrequenzen für die Layer vorgeben
- Verfügbar im R-Paket `Pareto`

Anwendung auf unser Problem-Beispiel:



Anwendung auf Beispiel 2

Beispiel 2 von oben (Motor-Marktquotierung):

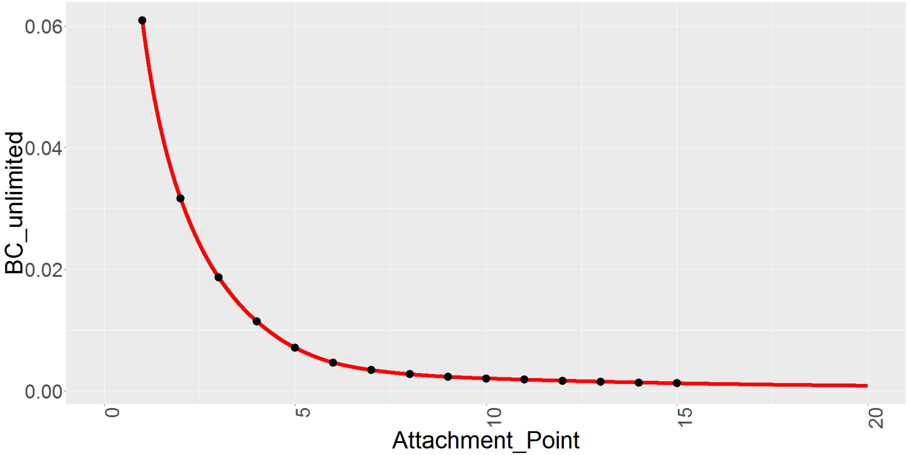
```
> a <- 1:15
> c <- c(diff(a), Inf)
> e <- c(0.0293, 0.0130, 0.0072, 0.0043, 0.0024, 0.0013, 0.0007, 0.0004, 0.0003,
+        0.0002, 0.00017, 0.00015, 0.00014, 0.00012, 0.0012)
> Fit <- Fit_References(c, a, e)
```

Probe:

```
> Layer_Mean(Fit, c, a)
 [1] 0.02930 0.01300 0.00720 0.00430 0.00240 0.00130 0.00070 0.00040 0.00030 0.00020
[11] 0.00017 0.00015 0.00014 0.00012 0.00120
```

DAV
DEUTSCHE
AKTUARVEREINIGUNG e.V.

Anwendung auf Beispiel 2



Literatur

S.W. Philbrick (1985) *A practical guide to the single parameter Pareto distribution*, PCAS LXXII, 44–84.

U. Riegel (2008) *Generalizations of common ILF models*, Blätter der DGVFM 29, 45–71.

U. Riegel (2018) *Matching tower information with piecewise Pareto*, European Actuarial Journal 8, 437–460.

U. Riegel (2018) *Pareto: The Pareto, Piecewise Pareto and Generalized Pareto Distribution*, R package version 2.4.2, https://github.com/ulrichriegel/Pareto#pareto.

M. Schmutz, R.R. Doerr (1998) *Das Pareto-Modell in der Sach-Rückversicherung. Formeln und Anwendungen*, Swiss Re Publications, Zürich

Fragen?