Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Anna Zalewska
Nr albumu: 234712

Ryzyko operacyjne: modelowanie, mierzenie i zabezpieczanie

Praca magisterska
na kierunku MATEMATYKA

Praca wykonana pod kierunkiem
dr inż. Przemysław Biecek
Instytut Matematyki Stosowanej i Mechaniki - Zakład Statystyki Matematycznej

Czerwiec 2010

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data | Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data | Podpis autora (autorów) pracy

Streszczenie

Niniejsza praca koncentruje się na metodach obliczania ryzyka operacyjnego z wykorzystaniem stworzonego przeze mnie pakietu `opVaR` napisanego dla języka programowania `R`. Przedstawione są liczne przykłady, ilustrujące rozwiązania zagadnień potrzebnych do obliczenia Value at Risk, m. in. problemu dopasowania rozkładów częstości i dotkliwości strat. Zwieńczeniem tych rozważań jest obliczenie wartości Value at Risk dla przykładowych danych.

Słowa kluczowe

VaR, ryzyko operacyjne, kwantyl

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.1 Matematyka

Klasyfikacja tematyczna

97M30 Financial and insurance mathematics

Tytuł pracy w języku angielskim

Modelling, measuring and hedging operational risk

Spis treści

Wprowadzenie . . . 5

1. Podstawowe pojęcia . . . 7

2. Regulacje - Bazylea II . . . 13
- 2.1. Podejście Najprostszego Wskaźnika . . . 13
- 2.2. Podejście Standaryzowane . . . 14
- 2.3. Podejścia Zaawansowane . . . 16
- 2.4. Kryteria kwalifikacji . . . 16
  - 2.4.1. Podejście Standaryzowane - kryteria . . . 16
  - 2.4.2. Podejścia Zaawansowane - kryteria . . . 17
- 2.5. Częściowy użytek . . . 20

3. Pakiet opVaR - wprowadzenie . . . 23
- 3.1. Dane . . . 23
- 3.2. Uwaga . . . 25
- 3.3. Wczytanie danych - wprowadzenie . . . 25
- 3.4. Macierz podsumowań strat . . . 26
- 3.5. Agregacja strat . . . 28

4. Częstość strat . . . 31
- 4.1. Histogramy strat . . . 31
- 4.2. Dopasowywanie rozkładu częstości strat . . . 39
  - 4.2.1. Komórka „Agency Services/Clients, Products & Business Practices” . 40
  - 4.2.2. Komórka „Corporate Finance/Execution, Delivery & Process Management” . . . 47

5. Dotkliwość strat . . . 57
- 5.1. Gęstość . . . 57
- 5.2. Dopasowywanie rozkładu dotkliwości strat . . . 63
  - 5.2.1. Komórka „Agency Services/Clients, Products & Business Practices” . 63

6. Value at Risk - wartość zagrożona ryzykiem straty . . . 95
- 6.1. Value at Risk dla wybranych komórek . . . 95
  - 6.1.1. Komórka „Commercial Banking/Clients, Products & Business Practices” 95
  - 6.1.2. Komórka „Agency Services/Clients, Products & Business Practices” . 99
- 6.2. Wszystkie linie biznesowe . . . 101
  - 6.2.1. „Agency Services” . . . 101
  - 6.2.2. „Asset Management” . . . 106

6.2.3. „Commercial Banking” 109
6.2.4. „Corporate Finance” 111
6.2.5. „Payment & Settlement” 113
6.2.6. „Retail Banking” 115
6.2.7. „Retail Brokerage” 117
6.2.8. „Trading & Sales” 119
6.2.9. Podsumowanie 121
6.3. Wszystkie komórki 124
6.4. Wszystkie straty 125
6.5. Value at Risk dla komórek, linii biznesowych i dla wszystkich strat 128
6.6. Podsumowanie 129

A. Lista rozkładów 131
A.1. Rozkłady dyskretne 131
A.1.1. Rozkład Poissona z parametrem λ 131
A.1.2. Rozkład dwumianowy z parametrami q, n $(0 < q < 1, n\ naturalne)$ . . 131
A.1.3. Rozkład ujemny dwumianowy z parametrami q, r 131
A.2. Rozkłady ciągłe 132
A.2.1. Rozkład beta 132
A.2.2. Rozkład Cauchy'ego 132
A.2.3. Rozkład χ^2 132
A.2.4. Rozkład wykładniczy 132
A.2.5. Rozkład Fishera 132
A.2.6. Rozkład gamma 133
A.2.7. Rozkład odwrotny gaussowski 133
A.2.8. Rozkład logarytmiczny normalny 133
A.2.9. Rozkład logistyczny 133
A.2.10. Rozkład normalny 133
A.2.11. Rozkład Weibulla 134

Bibliografia 135

Wprowadzenie

Praca składa się z sześciu rozdziałów. W pierwszym zdefiniowano podstawowe pojęcia dotyczące ryzyka operacyjnego i Value at Risk, opisano założenia zarządzania ryzykiem operacyjnym i zyski z niego płynące. Rozdział drugi koncentruje się na Bazylei II (Basel II), zbiorze rekomendacji do stosowania w prawie bankowym i regulacji, sporządzonym przez Bazylejski Komitet Nadzoru Bankowego (Basel Committee on Banking Supervision). Opisane zostały trzy rodzaje podejść do zarządzania ryzykiem operacyjnym i kryteria kwalifikacji, które powinien spełniać bank, aby mógł dane podejście wykorzystywać. Następnie, w kolejnym rozdziale, zostało zawarte wprowadzenie do stworzonego przez mnie pakietu `opVaR`, który jest głównym wynikiem pracy, i przedstawienie danych, które będą wykorzystane w obliczeniach. Rozdział ten zawiera również opisy sposobów kategoryzacji tych danych, ich agregacji i tworzenia podsumowań. Następnie dwa kolejne rozdziały są poświęcone dokonywaniu dopasowania do danych strat rozkładów częstości i dotkliwości. Wreszcie, w rozdziale `Value at risk - wartość zagrożona ryzykiem straty`, został obliczony VaR dla strat z każdej linii biznesowej, strat z każdej niepustej komórki zdefiniowanej przez podanie linii biznesowej i kategorii ryzyka, a także VaR dla wszystkich strat bez podziału na kategorie. W `Dodatku A` przedstawiona jest lista rozkładów prawdopodobieństwa wykorzystanych w pakiecie `opVaR`.

Rozdział 1

Podstawowe pojęcia

Ryzyko operacyjne od pewnego czasu jest tematem dyskusji, jednakże nie osiągnięto porozumienia odnośnie jego jednoznacznej definicji.

Dlaczego rozważamy ryzyko operacyjne? Po serii wielkich strat w latach 80 obserwujemy początki zarządzania ryzykiem i skoncentrowanie się na zmienności zarobków w zależności od stóp procentowych, wartości rynkowych i zdarzeń kredytowych. Dzięki mierzeniu można zredukować wpływ tych fluktuacji na zarobki, jednak nadal będą zachodzić duże straty, a wiele z nich powodowanych jest przez załamania systemu lub kontroli, niektóre są rezultatem przepisów prawnych, naturalnych katastrof i innych zdarzeń zewnętrznych. Główne źródło zmienności nie pochodzi więc z ryzyka finansowego. Nie odnosi się to do *finansowania* biznesu firmy, ale raczej do sposobu w jaki firma *operuje* swoim biznesem i jest to nazwane ryzykiem operacyjnym.

Wyobraźmy sobie, że chcemy odbyć podróż z Berlina do Londynu [1]. Mamy zaplanowaną trasę, wyliczone koszty i dni, jednak potem okazuje się, że pochłonęło to o wiele więcej czasu i środków. Winimy okoliczności specjalne i pech. Przyjaciel twierdzi jednak, że to nasza wina, gdyż należało zainwestować w systemy ostrzegania i bezpieczeństwa, nawigacji, system pozwalający na znośne tempo jazdy w złą pogodę, dostęp do zapasu pieniędzy. Tutaj właśnie mamy ryzyko operacyjne. Podstawowa propozycja to poprawienie systemów monitorujących i kontrolujących. Kluczowe składniki to:

- Definicja procesu (podróż z Berlina do Londynu);
- Wykonanie (czas, koszt, zarówno oczekiwane jak i rzeczywiste);
- Kategorie ryzyka (pogoda, błąd nawigacji, wypadek przy parkowaniu...);
- Systemy kontrolne (ABS, nawigacja GPS...).

Zauważmy, że:

- Nie są omawiane inne szlaki lub alternatywne sposoby podróży (byłaby to dyskusja bardziej strategiczna i zapewne wiodłaby w innym kierunku);
- Nie porównujemy oczekiwanych cen z rzeczywistymi (jest to poza naszą natychmiastową kontrolą);

[1] Przykład pochodzi z [King].

- Choć nie można kontrolować pogody, można złagodzić jej wpływ przez odpowiednie systemy;
- Nie rozważamy innych możliwych ryzyk, które mogłyby się zdarzyć (ale nie zdarzyły);

Te wszystkie dodatkowe systemy i kontrole mają swoją cenę. Jak zdecydować, które są potrzebne, a które jedynie dość przydatne?

Potrzeba więc drogi do ujęcia w ramy problemu, modelowania, liczenia miary ryzyka operacyjnego, by podjąć właściwą decyzję.

Definicja 1.0.1 *[King]* (*Ryzyko operacyjne*)

Ryzyko operacyjne to miara związku między aktywnościami biznesowymi firmy i wariancją jej rezultatów biznesowych.

Oczywiste jest, że nie można mieć strategii powiększenia zysku bez uwzględnienia ryzyka, bo ono też może wzrosnąć.
Zyski z zarządzania ryzykiem w procesach operacyjnych to m.in.:

- Unikamy nieoczekiwanych strat i poprawiamy wydajność operacyjną;
- Efektywne używanie kapitału, przypisanie kapitału do linii biznesowych;
- Zadowolenie udziałowców;
- Uporanie się z regulacjami.

Bazę do zarządzania ryzykiem operacyjnym zapewniają: znane relacje przyczyna - skutek (kluczowy element zarządzania ryzykiem), modele strat używające informacji statystycznych (kiedy np. brak przyczyn, przyczyny nie są znane) oraz wiarygodne scenariusze opisujące potencjalne straty i ich możliwe efekty.

Zarządzanie ryzykiem operacyjnym potrzebuje pewnej ramy, która musi zapewniać:

- Identyfikowanie ważnych dla firmy ryzyk;
- Klasyfikowanie ryzyka jako kontrolowalnego i niekontrolowalnego;
- Identyfikowanie przyczyn ryzyka kontrolowalnego;
- Przypisywanie ryzyku niekontrolowalnemu kategorii osłabiających;
- Otrzymywanie informacji o zmianach w ryzyku.

Istnieje kilka proponowanych podejść mających na celu spełnienie tych kryteriów.
Cztery z nich to:

1. Samoszacowanie ryzyka.
2. Analiza procesu.
3. Kategoryzacja strat.
4. Analiza wykonania.

Ramowe założenia to:

- Zarządzanie ryzykiem operacyjnym daje zysk wszystkim udziałowcom;
- Aby właściwie zarządzać ryzykiem operacyjnym, potrzeba systemu mierzenia, który łączy przyczynowo operacje firmy ze zmiennością zarobków;
- *Miara ryzyka operacyjnego musi odzwierciedlać zmienność zarobków jako rezultat strat powodowanych błędami i pomyłkami (odchyleniem od standardów, procedur lub planowanych czynności), załamań systemu, rzadkich zdarzeń, odzwierciedlając dwie kategorie: (a) wysoka częstotliwość, zdarzenia o niewielkim wpływie, które są oczekiwane i powodują straty operacyjne i (b) niska częstotliwość, wydarzenia o poważnym wpływie, które zdarzają się nieoczekiwanie i powodują niezwyczajne lub nadmierne straty;*
- Kontrola jest odzwierciedlana przez kombinację strat kontrolowalnych, wynikających z błędów i pomyłek (w ogólności o dużej frekwencji, niewielkim wpływie), oraz niekontrolowalnych, wynikających z wyjątków, załamań kontroli, rzadkich wydarzeń (w ogólności o niskiej frekwencji, dużym wpływie);
- Straty operacyjne są rezultatem kategorii przyczynowych w aktywnościach procesu dodawania wartości i muszą być przewidywane z użyciem techniki która wciela te kategorie do modelu;
- Straty nadmierne nie są związane z kategoriami przyczynowymi procesu dodawania wartości; muszą być analizowane z użyciem podejścia, które łączy aktualne wewnętrzne i zewnętrzne wydarzenia oraz generuje możliwe scenariusze ekstremalnych zdarzeń.

Przez VaR (Value at Risk) rozumieć będziemy kumulatywną wartość oczekiwanych strat na określonym przedziale ufności (np. 95 procent) dla wyspecyfikowanego okresu (np. 1 rok). Niech $F_X(x)$ oznacza dystrybuantę przychodu w określonym przedziale czasu, określoną dla portfela ryzyk. Ujemny przychód to strata, ale tutaj straty będą dodatnimi wartościami zmiennej losowej X. VaR zmiennej losowej X to $100p-$tny procent rozkładu X, oznaczany jako $VaR_p(X) = \pi_p$. To pokazuje, czemu VaR jest często nazywany *kwantylową miarą ryzyka.* Kiedy firma ma dostępny ten zasób kapitału, może to pochłonąć $100p\%$ możliwego wyniku.

Definicja 1.0.2 *Niech X będzie zmienną losową oznaczającą stratę. Value-at-Risk, czyli VaR, zmiennej X, oznaczana przez $VaR_p(X)$ lub π_p, to $100p$-ty procent (lub kwantyl) rozkładu X.*

Dla rozkładów ciągłych mamy $VaR_p(X)$ dla zmiennej X jako wartość π_p spełniającą:

$$P(X > \pi_p) = 1 - p$$

Definicja 1.0.3 *Przez koherentną (coherent) miarę ryzyka rozumiemy miarę $\rho(X)$, która dla każdych dwóch zmiennych losowych X, Y ma następujące własności:*

1. *Podaddytywność: $\rho(X + Y) \leqslant \rho(X) + \rho(Y)$.*
2. *Monotoniczność: Jeśli $X \leqslant Y$ dla każdego możliwego wyniku, to $\rho(X) \leqslant \rho(Y)$.*
3. *Dodatnia jednorodność: Dla każdej stałej dodatniej c, $\rho(cX) = c\rho(X)$.*
4. *Niezmienniczość ze względu na przesunięcia: Dla każdej stałej dodatniej c, $\rho(X+c) = \rho(X) + c$*

Założenia te wydają się być dosyć intuicyjne, jednakże VaR nie spełnia warunku podaddytywności, nie jest więc koherentną miarą ryzyka.

Przykład 1.0.1 [2] *Niech Z będzie zmienną losową ciągłego typu, wyrażającą stratę, o następujących wartościach dystrybuanty:*

$$F_Z(1) = 0.91$$

$$F_Z(90) = 0.95$$

$$F_Z(100) = 0.96$$

95%-owy kwantyl, $VaR_{95\%}(Z) = 90$ ponieważ jest 5% szansy na przekroczenie 90. Przypuśćmy, że rozdzielamy ryzyko Z na dwa oddzielne, ale zależne, ryzyka X oraz Y, takie, że $X + Y = Z$. Np. niech:

$$X = \begin{cases} Z, & Z \leqslant 100 \\ 0, & Z > 100 \end{cases}$$

oraz

$$Y = \begin{cases} 0, & Z \leqslant 100 \\ Z, & Z > 100 \end{cases}$$

Dla zmiennej X mamy:

$$F_X(1) = 0.95$$

$$F_X(90) = 0.99$$

$$F_X(100) = 1$$

Stąd $VaR_{95\%}(X) = 1$.
Podobnie, dystrybuanta Y spełnia $F_Y(0) = 0.96$, ponieważ mamy 96% szansy na brak straty. Stąd 95%-owy kwantyl nie może przekraczać 0, czyli $VaR_{95\%} \leqslant 0$. Otrzymujemy więc, że suma 95%-owych kwantyli dla X i Y wynosi mniej, niż $VaR_{95\%}(Z)$, co zaprzecza podaddytywności.

Wady VaR:

- Nie mówi nam, jakich ekstremalnych strat możemy się spodziewać, stwierdzając jedynie, że przez np. 95% czasu strata nie powinna być większa niż dana wartość; nie ma natomiast mowy o pozostałym czasie (tutaj rozwiązaniem może być EVT - Extreme Value Theory);
- Znaczącą wadą VaR, z której trzeba zdawać sobie sprawę, jest fakt, że bazuje na danych z przeszłości, próbując przewidzieć przyszłość, która przecież może wyglądać zupełnie inaczej w wyniku poważnych zmian na rynku.

Metodologia dla ryzyka operacyjnego musi być w stanie mierzyć dwa główne źródła ryzyka operacyjnego (wariancję w aktywnościach procesu oraz rzadkie zdarzenia) – tradycyjne metody nie są skuteczne dla obu.

[2] Przykład pochodzi z [Kl& Pa & Wi].

Metodologia $Delta - EVT^{TM}$.

Metoda *Delta* jest używana do ustalenia progu maksymalnej straty, który determinuje klasyfikację „dużych strat”, modelowanych przez EVT. *Delta* używa kategorii, które prowadzą do straty i ich wrażliwości do generowania rozkładów strat. Ustala funkcje zarobków i na ich podstawie liczy wariancję. Jej głównym ograniczeniem jest to, że może być użyta tylko do ryzyka przypisywalnego. EVT jest używana do uporania się z ogonem rozkładu i ustalenia minimalnego progu, od którego zaczynają się „duże straty”. Nawet jeśli „duża strata” jest przypisywalna, i tak jest analizowana z użyciem EVT. EVT dopasowuje wartości strat powyżej progu do Uogólnionego Rozkładu Pareto (GPD, Generalised Pareto Distribution), częstotliwość strat jest dopasowywana do rozkładu Poissona, następnie zaś te dwa rozkłady są łączone za pomocą symulacji Monte Carlo w celu uzyskania rozkładu nadmiernej straty. Dopasowanie danych do GPD jest kwestią trudną i delikatną. Ponieważ pragniemy aby wszystkie straty były objęte klasyfikacją, jak również aby wpadały w tylko jedną z dwóch metod - idealnie byłoby, gdyby progi ustalone przez obie te metody były sobie równe.

Rozdział 2

Regulacje - Bazylea II

Bazylea II (Basel II) jest pewnym zbiorem rekomendacji do stosowania w prawie bankowym i regulacji, sporządzonym przez Bazylejski Komitet Nadzoru Bankowego (Basel Committee on Banking Supervision). Początkowo została opublikowana w czerwcu 2004, potem dołączono poprawki. Jej celem jest stworzenie międzynarodowych standardów, których można by używać przy tworzeniu regulacji odnośnie kwot, jakie mają być odłożone przez banki na potrzeby różnych ryzyk finansowych i operacyjnych.

Zacznijmy od definicji ryzyka operacyjnego, jaką podaje Bazylea II w paragrafie 644:

Definicja 2.0.4 *Ryzyko operacyjne definiuje się jako ryzyko straty wynikającej z porażek lub nieodpowiedniości procesów wewnętrznych, ludzi i systemów oraz zdarzeń zewnętrznych. Ta definicja zawiera ryzyko prawne (obejmujące np. podatność na kary i grzywny), ale wyłącza strategiczne i reputacyjne.*

Bazylea specyfikuje trzy rodzaje podejść do ryzyka operacyjnego:

1. Podejście Najprostszego Wskaźnika (ang. the Basic Indicator Approach);
2. Podejście Standaryzowane (ang. the Standarised Approach);
3. Podejścia Zaawansowane (ang. the Advanced Measurement Approaches).

Zaznacza przy tym, że Banki zachęcane są do podążania wzdłuż spektrum dostępnych podejść w miarę jak wypracowują bardziej skomplikowane systemy i praktyki w mierzeniu ryzyka operacyjnego. Jednak, aby używać podejść 2. lub 3., bank musi spełnić pewne kryteria kwalifikujące, zaprezentowane w dalszej części Bazylei II; przy tym banki aktywne międzynarodowo oraz takie, w których mamy znaczącą ekspozycję na ryzyko operacyjne, są tymi, od których oczekuje się używania technik bardziej skomplikowanych niż 1. i odpowiednich dla profilu ryzyka instytucji. Warto też zauważyć, że Bazylea II postanawia, iż bank nie otrzyma pozwolenia na powrót do prostszego podejścia, kiedy już nadzorujący zaakceptowali bardziej zaawansowane. Jednak w przypadku, kiedy bank przestaje spełniać kryteria uprawniające do używania bardziej zaawansowanego podejścia, nadzór może nakazać stosowanie prostszego, dopóki te kryteria nie będą znów spełnione.

2.1. Podejście Najprostszego Wskaźnika

Bank używający tego podejścia musi trzymać dla ryzyka operacyjnego kapitał równy pewnemu procentowi (oznaczanemu przez α) średniego pozytywnego wpływu brutto z ostatnich trzech lat. Jeśli dla jakiegoś roku mamy dochód brutto negatywny lub równy zero, wyrzucamy go zarówno z licznika, jak i z mianownika, podczas obliczania średniej. Wzór przedstawia się następująco:

$$K_{BIA} = [\Sigma(GI_{1\ldots n} \times \alpha)]/n$$

gdzie:
K_{BIA} = kapitał wymagany przez Basic Indicator Approach;
GI = roczny dochód brutto, jeśli jest dodatni, przez trzy ostatnie lata;
n = liczba poprzednich trzech lat dla których dochód brutto był dodatni;
$\alpha = 15\%$; jest to liczba ustanowiona przez Komitet Bazylejski.
Dochód brutto definiuje się jako czysty (netto) dochód odsetkowy plus czysty dochód nieodsetkowy. Intencją jest, aby miara była:

- Brutto pod względem prowizji (np. niezapłacone odsetki);
- Brutto pod względem kosztów operacyjnych (z zawarciem wynagrodzeń za serwis zewnętrzny);
- Z wyłączeniem zrealizowanych zysków/strat ze sprzedaży zabezpieczeń w księgach bankowych;
- Z wyłączeniem niezwykłych lub nieregularnych pozycji tak samo jak dochodu z ubezpieczeń.

Nie ma specjalnych kryteriów wejściowych dla używania Basic Approach. Jednakże, banki używające Basic Approach są zachęcane do stosowania przewodnika Komitetu „Sound Practices for the Management and Supervision of Operational Risk" (luty 2003).

2.2. Podejście Standaryzowane

W tym podejściu aktywności banku są dzielone na osiem linii biznesowych [1] :

1. Corporate Finance (Finansowanie Przedsiębiorstw);
2. Trading & Sales (Sprzedaż i Operacje Spekulacyjne);
3. Retail Banking (Bankowość Detaliczna);
4. Commercial Banking (Bankowość Komercyjna);
5. Payment & Settlement (Płatności i Rozliczenia);
6. Agency Services (Usługi Pośrednictwa);
7. Asset Management (Zarządzanie Aktywami na Zlecenie);
8. Retail Brokerage (Brokerskie Usługi Detaliczne).

[1] Tłumaczenie na polski pochodzi z dokumentu konsultacyjnego DK/04/OPR Komisji Nadzoru Finansowego, noszącego tytuł „Metody proste wyliczania wymogów kapitałowych z tytułu ryzyka operacyjnego".

Linie biznesowe są zdefiniowane szczegółowo w Aneksie 8 Bazylei II.

W obrębie każdej linii biznesowej, dochód brutto jest wskaźnikiem, który służy jako zastępstwo do skalowania operacji biznesowych, a więc prawdopodobnie i skali ekspozycji ryzyka operacyjnego dla każdej z linii biznesowych. Wymagany kapitał dla każdej linii biznesowej jest naliczany przez mnożenie dochodu brutto przez czynnik (oznaczany β) przypisany do tej linii biznesowej. Należy odnotować, że w Standarised Approach dochód brutto jest mierzony dla każdej linii biznesowej, nie dla całej instytucji, np. w Corporate Finance wskaźnik jest dochodem brutto wygenerowanym w linii biznesowej Corporate Finance.
Całkowity wymagany kapitał liczy się jako trzyletnia średnia według linii biznesowych w każdym roku. W danym roku, ujemną wymaganą kwotę (wynikającą z ujemnego dochodu brutto) w dowolnej linii biznesowej mogą wyrównać dodatnie wymagane kwoty w innych liniach biznesowych (bez ograniczeń) [2]. Jednakże, jeśli zagregowany wymagany kapitał z wszystkich linii biznesowych daje w danym roku ma wartość ujemną, to w liczniku dajemy zero dla tego roku.
Całkowity wymagany kapitał liczymy według wzoru:

$$K_{TSA} = [\Sigma_{1...3} max[(GI_{1-8} \times \beta_{1-8}, 0)]]/3$$

gdzie:
K_{TSA} = kapitał wymagany przez Standarised Approach;
GI_{1-8} = roczny dochód brutto w danym roku, zdefiniowany jak dla Basic Indicator Approach, dla każdej z ośmiu linii biznesowych;
β_{1-8} = ustalony przez Komitet Bazylejski procent, odnoszący poziom wymaganego kapitału do poziomu dochodu brutto dla każdej z ośmiu linii biznesowych. Wartości te są przedstawione poniżej:

Linia Biznesowa	Wskaźnik β
Corporate Finance (β_1)	18%
Trading & Sales (β_2)	18%
Retail Banking (β_3)	12%
Commercial Banking (β_4)	15%
Payment & Settlement (β_5)	18%
Agency Services (β_6)	15%
Asset Management (β_7)	12%
Retail Brokerage (β_8)	12%

Warto jeszcze wspomnieć o Alternatywnym Podejściu Standaryzowanym (the Alternative Standarised Approach, ASA). Może ono zostać użyte przez bank za zgodą nadzoru, o ile bank zdoła wykazać, że podejście to zapewnia lepszą bazę dzięki np. unikaniu podwójnego liczenia ryzyka. Bazylea II stwierdza, że jeśli bank otrzymał zezwolenie na używanie ASA, nie będzie mógł już wrócić do Standarised Approach bez zgody nadzoru. ASA nie jest zalecana dla dużych, zdywersyfikowanych banków na głównych rynkach.
ASA odróżnia się od Standarised Approach dwoma liniami biznesowymi: Retail Banking i Commercial Banking. Dla tych linii biznesowych lokaty i pożyczki, przemnożone przez ustalone m, zastępują dochód brutto (wskaźnik oparty jest nie na zysku, lecz na wolumenie kredytów). β dla tych dwóch linii biznesowych pozostają niezmienione. Dla Retail Banking

[2] Zależy to jednak także od nadzoru narodowego.

mamy więc formułę:

$$K_{RB} = \beta_{RB} \times m \times LA_{RB}$$

gdzie:
K_{RB} = kapitał wymagany dla linii biznesowej Retail Banking;
LA_{RB} = wszystkie wyróżniające się lokaty i pożyczki (nieważone ryzykiem i brutto pod względem prowizji), uśrednione z trzech lat;
$m = 0.035$.
Taką samą formułę mamy dla Commercial Banking.
Poza tym bank może dla tych dwóch linii biznesowych użyć β równego 15%, a banki, które nie są w stanie podzielić swojego dochodu brutto na pozostałych 6 linii biznesowych, mogą zagregować dochód brutto dla tych 6 linii, używając β równego 18%, z liczeniem ujemnego dochodu brutto tak jak dla Standarised Approach.

2.3. Podejścia Zaawansowane

Pod AMA wymagany kapitał będzie równy mierze ryzyka generowanej przez wewnętrzny system mierzenia ryzyka wygenerowany przez bank, używający ilościowych i jakościowych kryteriów dla AMA, przedstawionych poniżej. Użycie AMA jest przedmiotem zatwierdzenia nadzoru.

Bank adaptujący AMA może za aprobatą nadzoru używać mechanizmu przypisywania dla celu zdeterminowania wymaganego kapitału międzynarodowo aktywnych podległych firm.

Nadzór oczekuje, że grupy bankowości AMA będą kontynuować wysiłki mające na celu opracowywanie technik alokacji ryzyka operacyjnego z wzrastającą wrażliwością na ryzyko, mimo początkowej aprobaty technik bazowanych na dochodzie brutto lub innych przybliżeniach dla ryzyka operacyjnego.

Ponadto, od banku adaptującego AMA będzie wymagane, aby wyliczał swoje wymagania kapitału używając tego podejścia tak samo jak i 1988 Accord zarysowanego w paragrafie 46 Bazylei.

2.4. Kryteria kwalifikacji

Dla the Standarised Approach i AMA obowiązują pewne kryteria, które trzeba spełnić, aby móc używać tych podejść.

2.4.1. Podejście Standaryzowane - kryteria

Kryteria dla tego podejścia dotyczą głównie spraw związanych z zarządzaniem ryzykiem i jego dokumentacją. Wymienia się m.in. takie zasady: nadzór będzie miał prawo nalegać na okres początkowego monitorowania tego podejścia w banku zanim będzie ono wykorzystane do wyliczenia wymagań kapitałowych dla celów regulacyjnych, bank musi wypracować specjalną politykę i mieć udokumentowane kryteria do przypisywania dochodu brutto do linii biznesowych oraz oczywiście mieć odpowiedni system zarządzania ryzykiem, śledzić straty w obrębie linii biznesowych, itp.

2.4.2. Podejścia Zaawansowane - kryteria

Aby używać tego podejścia, bank musi przekonać nadzór, że spełnia następujące minimalne kryteria:

- Jego dyrektorzy/zarząd i starsi menadżerowie są aktywnie zaangażowani w nadzór ramy zarządzania ryzykiem operacyjnym;
- Jego system zarządzania ryzykiem operacyjnym jest koncepcyjnie trafny i zaimplementowany w sposób integralny;
- Wykazuje się wystarczajacą pomysłowością w używaniu podejścia w głównych liniach biznesowych tak samo, jak i kontroli oraz strefie audytu.

Tak samo, jak przy poprzednim podejściu, bank będzie podlegał początkowemu monitorowaniu przez zarząd, zanim dostanie pozwolenie na stosowanie AMA do wyliczania kapitału regulacyjnego.
Bank musi rozsądnie estymować nieoczekiwane straty, bazując na połączeniu użycia wewnętrznych i odnośnych zewnętrznych danych o stratach, analizie scenariuszy, i specyficznego środowiska banku oraz wewnętrznych działów/faktorów. Poza tym powinien móc przypisywać kapitał ekonomiczny dla ryzyka operacyjnego według linii biznesowych w sposób, który stwarza pobudki do poprawy zarządzania ryzykiem operacyjnym linii biznesowych.

Standardy jakościowe

Wymienia się tu np. konieczność posiadania przez bank niezależnej funkcji zarządzania ryzykiem operacyjnym, która odpowiada m. in. za wypracowywanie strategii identyfikowania, mierzenia i kontrolowania/osłabiania ryzyka operacyjnego. Oczywiście wspomniana jest konieczność przypisywania kapitału ryzyka operacyjnego do głównych linii biznesowych i tworzenia pobudek do poprawy zarządzania ryzykiem operacyjnym. System mierzenia wewnętrznego ryzyka operacyjnego musi być ściśle zintegrowany z codziennymi procesami zarządzania ryzykiem operacyjnym; należy regularnie raportować o podatności na ryzyko operacyjne oraz doświadczonych stratach, cały system musi zaś być dobrze udokumentowany i stanowić przedmiot regularnej kontroli, a do jego ratyfikacji niezbędne jest stwierdzenie, że wewnętrzne procesy zatwierdzania działają w satysfakcjonujący sposób oraz przepływ danych wraz z procesami skojarzonymi z systemem mierzenia ryzyka są transparentne i dostępne.

Standardy ilościowe

Standardy trafności AMA Ważną informację stanowi fakt, że biorąc pod uwagę ciągłą ewolucję podejść analitycznych dla ryzyka operacyjnego, Komitet *nie specyfikuje podejścia lub założeń rozkładu użytych do wygenerowania miary ryzyka operacyjnego* dla celów regulacyjnych. Jednakże, bank musi być zdolny do zademonstrowania, że jego podejście obejmuje potencjalnie poważne „ogony” zdarzeń straty. Jakiekolwiek podejście zostało użyte, bank musi móc wykazać, że jego miara ryzyka operacyjnego spełnia standardy trafności porównywalne z tymi dla metody wewnętrznych ratingów [3] dla ryzyka kredytowego (to jest, porównywalne z jednorocznym okresem trzymania kapitału i 99.9 % przedziałem ufności) .
Komitet rozpoznaje, że standardy trafności AMA zapewniają sporo elastyczności, jednakże

[3]Tłumaczenie na polski pochodzi z dokumentu konsultacyjnego DK/04/OPR Komisji Nadzoru Finansowego, noszącego tytuł „Metody proste wyliczania wymogów kapitałowych z tytułu ryzyka operacyjnego”

przy wypracowywaniu swych systemów zarządzania ryzykiem i mierzenia ryzyka, bank musi utrzymywać rygorystyczne procedury dla wypracowywania modelu ryzyka operacyjnego oraz jego niezależnej ratyfikacji.

Szczegółowe kryteria Tutaj opisane są kryteria jakościowe, które zastosują się do wygenerowanych wewnętrznie miar ryzyka operacyjnego dla celów wyliczenia kapitału regulacyjnego. Są to, m. in. :

- Dowolny system musi być spójny z definicją ryzyka operacyjnego z paragrafu 644 oraz typami strat zdefiniowanymi w Aneksie 9 (główne kategorie to: Internal Fraud, External Fraud, Employment Practises and Workplace Safety; Clients, Products & Business Practises; Damage to Physical Assets, Business disruption and system failures, Execution, Delivery & Process Management);

- Nadzór będzie wymagał, aby bank liczył swój kapitał regulacyjny *jako sumę strat oczekiwanych (expected loss, EL) oraz strat nieoczekiwanych (unexpected loss, UL)*, chyba, że bank może wykazać, że w sposób dostateczny wychwytuje EL w wewnętrznych praktykach biznesowych. To oznacza, iż, aby bazować wymagania minimalnego kapitału regulacyjnego tylko na Ul, bank musi wykazać, że podatność na EL jest zmierzona i uzasadniona;

- Bankowy system mierzenia ryzyka *musi być wystarczająco „ziarnisty”*, aby wychwytywał główne czynniki ryzyka operacyjnego wpływające na kształt ogona estymacji strat;

- Miary ryzyka dla różnych oszacowań ryzyka operacyjnego muszą być dodawane dla celów wyliczania minimalnego wymagania kapitału regulacyjnego. Jednakże, bank może otrzymać pozwolenie na używanie wewnętrznie określonych korelacji w stratach ryzyka operacyjnego, o ile wykaże, że jego system dla określania tych korelacji są trafne, zaimplementowane w sposób integralny i uwzględniają niepewność towarzyszącą takim oszacowaniom korelacji;

- Dowolny system mierzenia ryzyka operacyjnego musi mieć pewne kluczowe cechy, aby spełnić wymagania nadzoru dotyczące trafności. Te elementy muszą zawierać użycie *danych wewnętrznych, odnośnych danych zewnętrznych, analiz scenariuszy oraz działów odzwierciedlających środowisko biznesowe i wewnętrzne systemy kontroli*;

- Bank potrzebuje mieć wiarygodne, transparentne, dobrze udokumentowane i weryfikowalne podejście do ważenia tych fundamentalnych składników w swoim ogólnym systemie mierzenia ryzyka operacyjnego. Na przykład, w pewnych przypadkach estymacja 99.9 % przedziału ufności bazowana na danych wewnętrznych i zewnętrznych stratach, może być niemiarodajna dla rozkładów z grubymi ogonami i małą ilością obserwowanych strat. W takich przypadkach analiza scenariuszyk, środowisko biznesowe i działy kontroli mogą grać bardziej dominującą rolę. We wszystkich przypadkach ważenie czterech głównych elementów powinno być wewnętrznie spójne i unikać podwójnego liczenia jakościowych oszacowań lub osłabiaczy ryzyka już rozpoznanych w innych elementach ramy.

Dane wewnętrzne Wewnętrzne dane o stratach są kluczowe w wypracowaniu i funkcjonowaniu wiarygodnego systemu mierzenia ryzyka operacyjnego. Są decydujące w wiązaniu oszacowań ryzyka banku z jego aktualnymi danymi o stratach. Cel można osiągnąć na wiele

sposobów, włącznie z użyciem wewnętrznych danych strat jako oparcia dla oszacowań empirycznych ryzyka, jako podstawy zatwierdzania danych wejściowych i wyjściowych systemu mierzenia ryzyka, czy jako łącznik pomiędzy doświadczonymi stratami a zarządzaniem ryzykiem oraz decyzjami kontrolnymi.
Wewnętrzne dane o stratach znajdują zastosowanie głównie gdy są wyraźnie połączone z bieżącymi aktywnościami biznesowymi, procesami technologicznymi i procedurami zarządzania ryzykiem. Tak więc bank musi mieć udokumentowane procedury dla oszacowania tego związku historycznych danych o stratach, włącznie z np. sytuacjami takimi, jak skalowanie lub inne dopasowania (a także, kiedy można ich użyć, do jakiego stopnia itp.).
Poza tym wewnętrznie generowane miary ryzyka operacyjnego używane dla celów kapitału regulacyjnego muszą być bazowane na co najmniej pięcioletnim okresie obserwacji wewnętrznych danych o stratach, niezależnie od tego, czy używa się ich do budowania, czy weryfikacji. Kiedy bank po raz pierwszy zaczyna używać AMA, dopuszczalny jest okres trzech lat.

Aby zakwalifikować się dla celów regulacji kapitału, procesy zbierania danych o wewnętrznych stratach muszą spełniać poniższe kryteria:

- Aby asystować w zatwierdzaniu przez zarząd, bank musi być w stanie przydzielić swoje historyczne dane o stratach wewnętrznych do głównych kategorii strat z Aneksów 8 i 9 oraz udostępnić je nadzorowi na żądanie. Musi mieć udokumentowane, obiektywne kryteria przypisywania tych strat do wyspecyfikowanych linii biznesowych i typów zdarzeń. Jednakże, pozostawia się bankowi decyzję, do jakiego stopnia zaaplikuje te kategoryzacje w swoim systemie mierzenia wewnętrznego ryzyka operacyjnego;

- Wewnętrzne dane o stratach muszą być kompleksowe w takim znaczeniu, że ogarniają wszystkie materialne aktywności i podatności ze wszystkich stosownych podsystemów i lokalizacji geograficznych. Bank musi umieć wykazać, że pominięte aktywności lub podatności, indywidualnie lub w połączeniu, nie będą miały materialnego wpływu na ogólne oszacowanie ryzyka. Bank musi mieć właściwy de minimis [4] próg straty brutto dla zbierania danych o stratach wewnętrznych, np. 10 000 €. Właściwy próg może nieco wahać się między bankami i w samym banku, w zależności od linii biznesowej i/lub typu zdarzenia;

- Oprócz informacji o wielkości straty, bank powinien odnotowywać także informację o dacie zdarzenia, odzyskanych kwotach oraz informacje opisowe. Poziom szczegółowości takiego opisu powinien być proporcjonalny do kwoty straty brutto.

- Bank musi wypracować specyficzne kryteria przypisywania danych o stratach wynikających ze zdarzenia w scentralizowanej funkcji (np. w departamencie informatycznym) lub aktywności, która obejmuje więcej niż jedną linię biznesową, tak samo, jak i z powiązanych zdarzeń na przestrzeni upływu czasu;

- Te straty ryzyka operacyjnego, które są związane z ryzykiem kredytowym i historycznie były włączane w bazy danych ryzyka operacyjnego, nadal mają być traktowane jak ryzyko kredytowe w celu wyliczania minimalnego kapitału regulacyjnego pod tą ramą. Nie będą się one liczyły do kapitału dla ryzyka operacyjnego. Jednakże, dla celów zarządzania ryzykiem operacyjnym, bank musi identyfikować wszystkie materialne straty ryzyka operacyjnego zgodne z podaną definicją ryzyka operacyjnego i typami zdarzeń z Aneksu 9, włącznie z tymi związanymi z ryzykiem operacyjnym;

[4] „o rzeczach minimalnych”, w szacowaniu ryzyka odnosi się do poziomu ryzyka, który jest zbyt mały, aby go rozważać

- Te straty ryzyka operacyjnego, które są powiązane z ryzykiem rynkowym, są traktowane jako ryzyko operacyjne dla celów wyliczania minimalnego kapitału regulacyjnego w tej ramie.

Dane zewnętrzne System mierzenia ryzyka operacyjnego banku musi używać powiązanych danych zewnętrznych, zwłaszcza jeśli uważa się, że jest narażony na nieczęste, lecz potencjalnie poważne straty. Te dane powinny zwierać informacje o kwotach strat, skali operacji biznesowych, kiedy zaszło zdarzenie, wiadomości o przyczynach i okolicznościach i inne, które pomogą w oszacowaniu odnośności tego zdarzenia straty do innego banku. Bank musi mieć systematyczne procesy określania sytuacji, dla których trzeba użyć danych zewnętrznych oraz metod używanych do wcielenia tych danych (np. skalowanie).

Analiza scenariuszy Bank musi używać analizy scenariuszy w połączeniu z danymi zewnętrznymi aby ocenić swą podatność na zdarzenia o poważnym wpływie. To podejście opiera się na wiedzy starszych managerów i ekspertów od zarządzania ryzykiem. Oszacowania ekspertów mogą być wyrażone na przykład jako parametry przyjętego statystycznego rozkładu strat.

Środowisko biznesowe i wewnętrzne działy kontroli Oprócz danych o stratach, czy to rzeczywistych, czy bazowanych na scenariuszach, bank musi uchwycić kluczowe cechy środowiska biznesowego i wewnętrznych działów kontroli, które mogą zmienić jego profil ryzyka operacyjnego.

Osłabianie ryzyka

Pod AMA, bankowi zezwoli się na rozpoznawanie wpływ ubezpieczenia na osłabianie ryzyka w miarach ryzyka operacyjnego używanych do wyliczenia minimalnego kapitału regulacyjnego, z limitem 20 % całkowitego wymagania kapitałowego pod AMA.
Należy do tego spełniać m. in. następujące kryteria:

- Polisa ubezpieczeniowa musi mieć początkowy termin nie krótszy niż rok, dla krótszych należy stosować odpowiednie obcięcia, do 100 % włącznie, jeśli polisa ma termin nie dłuższy niż 90 dni;
- Polisa ubezpieczeniowa ma minimalny okres wypowiedzenia 90 dni;
- Wyliczenia osłabiania ryzyka muszą odzwierciedlać pokrycie przez ubezpieczenie w sposób przejrzysty jeśli chodzi o jego związek i spójność z aktualnym prawdopodobieństwem i wpływem straty;
- Ubezpieczenie jest zapewniane przez stronę trzecią.

2.5. Częściowy użytek

Bankowi można udzielić pozwolenia na używanie AMA dla pewnych części jego operacji i Basic Indicator Approach lub Standarised Approach dla balansu (użytek częściowy), pod warunkiem, że:

- Wszystkie ryzyka operacyjne są uchwycone;

- Wszystkie operacje banku w obrębie AMA spełniają jakościowe kryteria używania AMA, a te, które używają prostszego podejścia, spełniają jakościowe kryteria dla tego podejścia;
- W chwili zaimplementowania AMA, znacząca część ryzyka operacyjnego banku jest obejmowana przez AMA;
- Bank przedstawi nadzorowi plan, według którego wprowadzi AMA do wszystkich, poza niematerialnymi, części swych operacji.

Bank może określić, do zatwierdzenia przez nadzór, które linie biznesowe, struktury prawne, geograficzne lub inne chce objąć AMA.

W pewnych wyjątkowych sytuacjach nadzór może dozwolić na implementację innego niż AMA podejścia.

Rozdział 3

Pakiet opVaR - wprowadzenie

Zaprezentuję teraz działanie napisanego przeze mnie pakietu `opVaR` na danych symulowanych o strukturze zgodnej z rzeczywistością.

3.1. Dane

Oto nasze dane strat. Przykładowe dane strat są włączone w pakiet. Jest to lista R składająca się z 5 elementów. Zobaczmy jej proste podsumowanie:

```
> data(loss.data.object)
> summary(loss.data.object)

         Length Class      Mode
losses   4      data.frame list
risk     2      data.frame list
rcateg   7      -none-     character
business 2      data.frame list
blines   8      -none-     character
```

Dwa elementy nie będące data.frames, czyli `rcateg` i `blines`, to nazwy kategorii ryzyka i linii biznesowych.

```
> loss.data.object$rcateg

[1] "Business Disruption and System Failures"
[2] "Clients, Products & Business Practices"
[3] "Damage to Physical Assets"
[4] "Employment Practices and Workplace Safety"
[5] "Execution, Delivery & Process Management"
[6] "External Fraud"
[7] "Internal Fraud"

> loss.data.object$blines

[1] "Agency Services"      "Asset Management"     "Commercial Banking"
[4] "Corporate Finance"    "Payment & Settlement" "Retail Banking"
[7] "Retail Brokerage"     "Trading & Sales"
```

Dwa inne elementy, `loss.data.object$risk` i `loss.data.object$business`, mają przypisywać pewne numery do kategorii ryzyka i linii biznesowych. Jak widać, jest to połączone z bardziej szczegółowym podziałem strat, gdyż danej linii biznesowej lub kategorii ryzyka odpowiada więcej niż jeden numer. Na przykład:

```
> loss.data.object$business[loss.data.object$business[,2] == "Retail Banking",]

    Internal_BL_ID Path_Component_1
58             650   Retail Banking
59             651   Retail Banking
60             652   Retail Banking
61             653   Retail Banking
62             654   Retail Banking
294         500816   Retail Banking
295         510817   Retail Banking
296         510818   Retail Banking
297         520819   Retail Banking
298         520820   Retail Banking
299         520821   Retail Banking
300         520822   Retail Banking
301         520823   Retail Banking
302         530824   Retail Banking
303         530825   Retail Banking
304         530826   Retail Banking
305         530827   Retail Banking
306         540828   Retail Banking
307         540829   Retail Banking
308         540830   Retail Banking
309         540831   Retail Banking
```

Znajdujemy wszystkie numery odpowiadające linii biznesowej `"Retail Banking"`. Wszystkie te numery Internal_BL_ID to numery linii `"Retail Banking"`.

Teraz zobaczmy pierwszych sześć wierszy `loss.data.object$losses`:

```
> head(loss.data.object$losses)

  Internal_BL_ID Internal_RC_ID First_Date_of_Event Gross_Loss_Amount
1         240401            183          2002-01-03           1642.26
2         570946             77          2002-01-06           2498.33
3          20105             54          2002-01-08           7420.72
4          20107             57          2002-01-09          27019.26
5         300424             95          2002-01-10           1829.98
6          20105             92          2002-01-11          12164.67
```

Możemy zapytać, do jakiej linii biznesowej i kategorii ryzyka jest przypisana strata z pierwszego wiersza.

```
> loss.data.object$risk[loss.data.object$risk[,1] ==183,]
```

```
    Internal_RC_ID Path_Component_1
182            183   Internal Fraud
```

Tak więc kategorią ryzyka jest "Internal Fraud" ...

```
> loss.data.object$business[loss.data.object$business[,1] == 240401,]
```

```
    Internal_BL_ID   Path_Component_1
179         240401 Commercial Banking
```

...a linią biznesową jest "Commercial Banking".

3.2. Uwaga

W celu prawdziwego estymowania danych powinniśmy mieć pełne okresy, a takich nie mamy. Wszystkie estymacje zostały wykonane na tym około czteroletnim okresie pomiędzy "2002-01-03" i "2006-02-12".

Teraz zajmiemy się wczytaniem danych:

3.3. Wczytanie danych - wprowadzenie

Zacznijmy teraz przetwarzać dane. Chcemy mieć przedstawione wcześniej dane zaklasyfikowane do linii biznesowych i kategorii ryzyka. Użyjemy funkcji `read.loss()` - ta funkcja wymaga podania numeru linii biznesowej, numeru kategorii ryzyka i naszej listy `loss.data.object`. Niech linią biznesową będzie 1, a linią ryzyka 2.

```
>  loss.data.object$blines[1]
```

```
[1] "Agency Services"
```

```
> loss.data.object$rcateg[2]
```

```
[1] "Clients, Products & Business Practices"
```

Wartością naszej funkcji jest `x12`:

```
>  x12<- read.loss(b=1,r=2,loss.data.object)
> head(x12)
```

```
   First_Date_of_Event Gross_Loss_Amount
3           2002-01-08           7420.72
4           2002-01-09          27019.26
6           2002-01-11          12164.67
8           2002-01-16           4983.20
9           2002-01-16           5894.24
10          2002-01-16           3162.60
```

```
>  dim(x12)
```

```
[1] 806   2
```

Funkcja `read.loss()` wczytuje straty (daty i kwoty).

3.4. Macierz podsumowań strat

Wygodnie mogłoby mieć także mieć macierz podsumowań strat. Zauważmy, że `D[i,1,j]` to strata maksymalna, `D[i,2,j]` średnia strata, `D[i,3,j]` strata minimalna, a `D[i,4,j]` ilość strat przy `i` będącym numerem linii biznesowej i `j` będącym numerem kategorii ryzyka.

```
> D <- loss.matrix(loss.data.object)
> D

, , 1

     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
[4,]    0    0    0    0
[5,]    0    0    0    0
[6,]    0    0    0    0
[7,]    0    0    0    0
[8,]    0    0    0    0

, , 2

           [,1]      [,2]    [,3] [,4]
[1,] 1790529.75 47079.516  794.22  806
[2,]       0.00     0.000    0.00    0
[3,]   59600.88  4428.156  891.80  285
[4,]       0.00     0.000    0.00    0
[5,]       0.00     0.000    0.00    0
[6,]       0.00     0.000    0.00    0
[7,] 1181407.58 19857.792 1003.45  459
[8,]       0.00     0.000    0.00    0

, , 3

          [,1]      [,2]    [,3] [,4]
[1,]      0.00     0.000    0.00    0
[2,]      0.00     0.000    0.00    0
[3,]      0.00     0.000    0.00    0
[4,]  88421.37  5601.253 1028.65  128
[5,] 214527.15 14215.301  969.82  328
[6,]      0.00     0.000    0.00    0
[7,]      0.00     0.000    0.00    0
[8,] 274861.47  7164.078 1004.44  192

, , 4
```

```
          [,1]      [,2]    [,3] [,4]
[1,]      0.00      0.00    0.00    0
[2,]      0.00      0.00    0.00    0
[3,]  32229.74  4248.81 1009.01   61
[4,]      0.00      0.00    0.00    0
[5,] 143152.28 16925.76  974.28  137
[6,]      0.00      0.00    0.00    0
[7,]      0.00      0.00    0.00    0
[8,]      0.00      0.00    0.00    0

, , 5

           [,1]       [,2]    [,3] [,4]
[1,]       0.00      0.000    0.00    0
[2,] 2222723.89 91837.982 1008.43  139
[3,]       0.00      0.000    0.00    0
[4,]   27339.64  3715.261  951.42   38
[5,]   80946.39 11526.309 1005.58  126
[6,]   59086.99 10369.983 1083.44   15
[7,]  214323.74 11388.459  866.25  119
[8,]       0.00      0.000    0.00    0

, , 6

           [,1]       [,2]    [,3] [,4]
[1,] 2272729.37 64208.694  890.15  102
[2,]       0.00      0.000    0.00    0
[3,]   57369.05  6624.523 1066.77   54
[4,]   46101.82  5882.500 1095.27   32
[5,]       0.00      0.000    0.00    0
[6,]       0.00      0.000    0.00    0
[7,]       0.00      0.000    0.00    0
[8,]       0.00      0.000    0.00    0

, , 7

           [,1]        [,2]    [,3] [,4]
[1,]  310442.89  32131.851 1019.38   80
[2,] 3875031.69 132019.721 1151.49  173
[3,]   13110.72   2985.808 1005.67   64
[4,]       0.00      0.000    0.00    0
[5,]  144495.59  16282.654 1024.10  125
[6,]   29386.38   6080.325 1177.93   20
[7,]  111428.79  11412.683 1029.68  126
[8,]       0.00      0.000    0.00    0
```

Użyteczna mogłaby być także tabela. Użyjmy funkcji `loss.matrix.image()`:

```
> loss.matrix.image(data = loss.data.object)
```

Rysunek 3.1: Macierz podsumowań strat

Może to być także:

```
> loss.matrix.image(D,loss.data.object$blines,loss.data.object$rcateg)
```

Jest teraz jasna, że nie ma strat pochodzących z linii biznesowej `"Business Disruption and System Failures"` i dowolnej kategorii ryzyka.

W komórce `"Clients, Products & Business Practices"`, `"Agency Services"` mamy 806 strat. Średnia strata jest równa 47079.52, maksymalna 1790529.75 Te wartości można znaleźć także w `D[1, ,2]`; mamy minimalną stratę `D[1,3,2] = 794.22`. Ta komórka ma kolor `col3`, który jest przeznaczony do wyróżniania komórek z średnią stratą większą lub równą średniej stracie dla `D` plus odchylenie standardowe średniej straty dla `D`.

3.5. Agregacja strat

Teraz chcielibyśmy mieć możliwość zmieniania okresów od dni (`days`) do tygodni (`weeks`) lub miesięcy (`months`), a nawet kwartałów (`quarters`); ma to związek z agregacją strat. Zobaczmy na przykładzie:

```
> x65 <- read.loss(6,5,loss.data.object)
> dim(x65)
```

```
[1] 15  2
```

Te dane pochodzą z linii biznesowej o numerze 6 i kategorii ryzyka o numerze 5. Użycie funkcji `dim()` pozwala nam zobaczyć liczbę wierszy i kolumn w `x65`.

```
> x65

     First_Date_of_Event Gross_Loss_Amount
63            2002-02-20           6118.85
165           2002-04-23           7942.78
622           2003-01-21           3325.98
711           2003-03-06           3405.60
984           2003-07-01           1426.88
1211          2003-10-14           1083.44
1325          2003-12-03           3284.29
1444          2004-01-28           6855.69
1956          2004-08-09           1894.01
1957          2004-08-09          37248.23
2141          2004-10-19          13553.71
2249          2004-12-03           5690.19
2496          2005-03-02           1570.91
2500          2005-03-02           3062.20
3032          2005-08-23          59086.99
```

Bez agregowania:

```
> t0 <- period.loss(x65,"none"); t0

 [1]  6118.85  7942.78  3325.98  3405.60  1426.88  1083.44  3284.29  6855.69
 [9]  1894.01 37248.23 13553.71  5690.19  1570.91  3062.20 59086.99
```

Straty złączone po dniach ...

```
> t1 <- period.loss(x65,"days"); t1

 [1]  6118.85  7942.78  3325.98  3405.60  1426.88  1083.44  3284.29  6855.69
 [9] 39142.24 13553.71  5690.19  4633.11 59086.99
```

... tygodniach ...

```
>  t2 <- period.loss(x65,"weeks"); t2

 [1]  6118.85  7942.78  3325.98  3405.60  1426.88  1083.44  3284.29  6855.69
 [9] 39142.24 13553.71  5690.19  4633.11 59086.99
```

... miesiącach ...

```
>  t3 <- period.loss(x65,"months"); t3

 [1]  6118.85  7942.78  3325.98  3405.60  1426.88  1083.44  3284.29  6855.69
 [9] 39142.24 13553.71  5690.19  4633.11 59086.99
```

... i kwartałach:

```
> t4 <- period.loss(x65,"quarters"); t4

 [1]  6118.85  7942.78  6731.58  1426.88  4367.73  6855.69 39142.24 19243.90
 [9]  4633.11 59086.99
```

Czasem jednak chcielibyśmy widzieć również daty; wtedy zalecane jest użycie opcji `dts` („dates" - daty).

```
> t0 <- period.loss(x65,"none",dts=TRUE); t0

2002-02-20 2002-04-23 2003-01-21 2003-03-06 2003-07-01 2003-10-14 2003-12-03
   6118.85    7942.78    3325.98    3405.60    1426.88    1083.44    3284.29
2004-01-28 2004-08-09 2004-08-09 2004-10-19 2004-12-03 2005-03-02 2005-03-02
   6855.69    1894.01   37248.23   13553.71    5690.19    1570.91    3062.20
2005-08-23
  59086.99

>  t1 <- period.loss(x65,"days",dts=T); t1

2002-02-20 2002-04-23 2003-01-21 2003-03-06 2003-07-01 2003-10-14 2003-12-03
   6118.85    7942.78    3325.98    3405.60    1426.88    1083.44    3284.29
2004-01-28 2004-08-09 2004-10-19 2004-12-03 2005-03-02 2005-08-23
   6855.69   39142.24   13553.71    5690.19    4633.11   59086.99

> t2 <- period.loss(x65,"weeks",dts=T); t2

2002-02-18 2002-04-22 2003-01-20 2003-03-03 2003-06-30 2003-10-13 2003-12-01
   6118.85    7942.78    3325.98    3405.60    1426.88    1083.44    3284.29
2004-01-26 2004-08-09 2004-10-18 2004-11-29 2005-02-28 2005-08-22
   6855.69   39142.24   13553.71    5690.19    4633.11   59086.99

>  t3 <- period.loss(x65,"months",dts=T); t3

2002-02-01 2002-04-01 2003-01-01 2003-03-01 2003-07-01 2003-10-01 2003-12-01
   6118.85    7942.78    3325.98    3405.60    1426.88    1083.44    3284.29
2004-01-01 2004-08-01 2004-10-01 2004-12-01 2005-03-01 2005-08-01
   6855.69   39142.24   13553.71    5690.19    4633.11   59086.99

> t4 <- period.loss(x65,"quarters",dts=T); t4

2002-01-01 2002-04-01 2003-01-01 2003-07-01 2003-10-01 2004-01-01 2004-07-01
   6118.85    7942.78    6731.58    1426.88    4367.73    6855.69   39142.24
2004-10-01 2005-01-01 2005-07-01
  19243.90    4633.11   59086.99
```

Są to te same wyniki, co przedtem, jednak z dodanymi datami. Zauważmy, że tylko dla dni i braku okresu (`"none"`) są to oryginalne daty; dla tygodni, miesięcy i kwartałów są jedynie datami otwierającymi okres, w którym zaszła strata.

Rozdział 4

Częstość strat

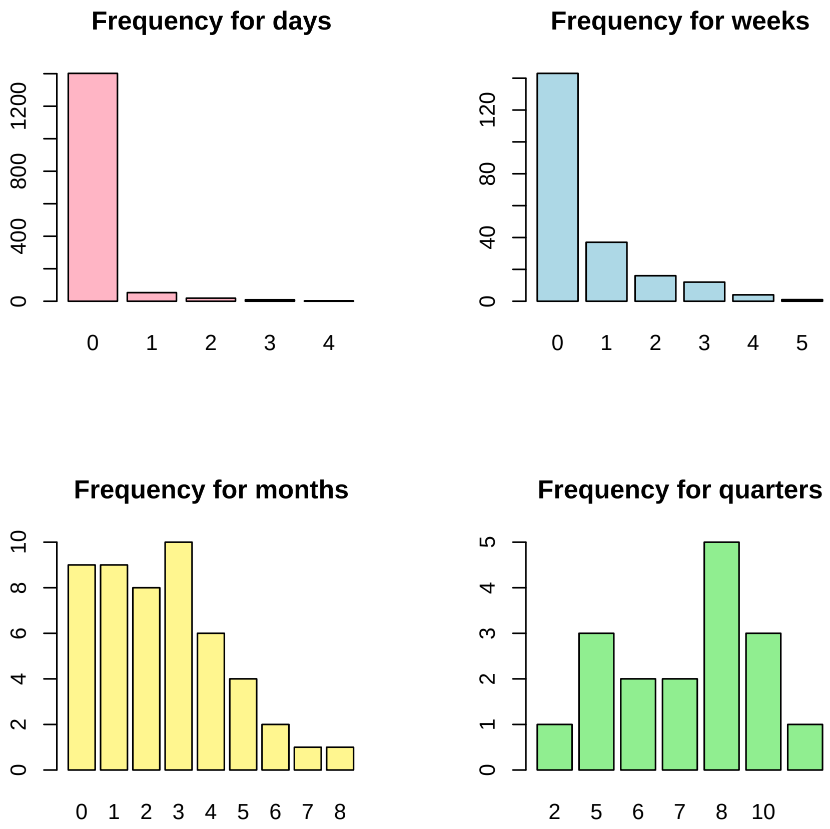
4.1. Histogramy strat

Teraz pragniemy dopasować rozkład częstości strat. Mamy funkcję rysującą histogramy częstości strat, `hist.period()`. Zobaczmy takie histogramy dla piątej linii biznesowej i piątej kategorii ryzyka. Co istotne, funkcja ta posiada opcję zmiany okresów (straty mogą być agregowane po dniach, tygodniach, miesiącach lub kwartałach). Zgodnie z *Basel II*, *„Bankowy system mierzenia ryzyka musi być wystarczająco „ziarnisty”, aby wychwytywał główne czynniki ryzyka operacyjnego wpływające na kształt ogona estymacji strat”*.

```
> x55<- read.loss(5,5,loss.data.object)
```

Piąta kategoria ryzyka to`"Execution, Delivery & Process Management"`, a piąta linia biznesowa to `"Payment & Settlement"`. .

```
> z<- {}
> par(mfrow=c(2,2))
>  z$days <- hist.period(x55,"days",col = "pink1")
>  z$weeks <- hist.period(x55,"weeks",col = "lightblue")
>  z$months <- hist.period(x55,"months",col = "khaki1" )
>  z$quarters <- hist.period(x55,"quarters",col = "lightgreen")
```



Rysunek 4.1: Histogramy częstości dla danych x55 i różnych okresów

Użyto różnych kolorów słupków.

```
> z

$days
$days$y

   0     1     2     3     4
1402    53    19     9     2


$weeks
$weeks$y
y
  0   1   2   3   4   5
143  37  16  12   4   1


$months
$months$y
```

```
y
 0  1  2  3  4  5  6  7  8
 9  9  8 10  6  4  2  1  1


$quarters
$quarters$y
y
 2  5  6  7  8 10 13
 1  3  2  2  5  3  1
```

Jak można łatwo sprawdzić (`z$days$y`), były `1402` dni bez strat, `53` z jedną stratą, `19` z dwoma, `9` z trzema i tylko `2` z czterema. Nie było dnia z pięcioma stratami. Z `z$weeks$y` dowiadujemy się, że były `143` tygodnie bez strat, ...

Oczywiście, jest jedno kluczowe pytanie wiążące się z łączeniem strat po dniach: czy weekendy powinny być liczone, czy nie?
Może dla danej linii biznesowej i linii ryzyka nie może być strat w weekendy? Czy możemy to sprawdzić?

Funkcja `weekdays` wydobywa dzień tygodnia; `x55[,1]` to daty, ale i tak należy użyć funkcji `as.Date`. Zobaczmy to dla `x55`.

Jaki widzimy, w naszych danych są weekendy, jednak nadal możemy się spodziewać pewnej asymetrii. Powódź nie rozróżnia pomiędzy dniem świątecznym a biznesowym, natomiast dla kogoś popełniającego defraudację może być to znacząca różnica.

```
> weekdays(as.Date(x55[,1]))

 [1]  "niedziela"     "czwartek"      "czwartek"      "czwartek"      "piątek"
 [6]  "piątek"        "niedziela"     "środa"         "niedziela"     "środa"
[11]  "czwartek"      "czwartek"      "wtorek"        "niedziela"     "poniedziałek"
[16]  "poniedziałek"  "środa"         "sobota"        "wtorek"        "sobota"
[21]  "czwartek"      "środa"         "czwartek"      "czwartek"      "poniedziałek"
[26]  "poniedziałek"  "poniedziałek"  "czwartek"      "środa"         "środa"
[31]  "sobota"        "czwartek"      "piątek"        "niedziela"     "poniedziałek"
[36]  "sobota"        "wtorek"        "niedziela"     "niedziela"     "czwartek"
[41]  "czwartek"      "wtorek"        "wtorek"        "niedziela"     "czwartek"
[46]  "czwartek"      "czwartek"      "czwartek"      "niedziela"     "niedziela"
[51]  "czwartek"      "piątek"        "piątek"        "poniedziałek"  "czwartek"
[56]  "wtorek"        "piątek"        "wtorek"        "piątek"        "piątek"
[61]  "poniedziałek"  "niedziela"     "niedziela"     "niedziela"     "sobota"
[66]  "sobota"        "czwartek"      "sobota"        "sobota"        "niedziela"
[71]  "sobota"        "wtorek"        "wtorek"        "wtorek"        "czwartek"
[76]  "czwartek"      "poniedziałek"  "poniedziałek"  "wtorek"        "piątek"
[81]  "piątek"        "piątek"        "piątek"        "środa"         "wtorek"
[86]  "piątek"        "piątek"        "niedziela"     "piątek"        "wtorek"
[91]  "wtorek"        "wtorek"        "wtorek"        "niedziela"     "sobota"
[96]  "sobota"        "sobota"        "wtorek"        "wtorek"        "środa"
```

```
[101]  "piątek"     "niedziela"     "niedziela" "wtorek"    "sobota"
[106]  "sobota"     "środa"         "środa"     "środa"     "wtorek"
[111]  "wtorek"     "wtorek"        "niedziela" "czwartek"  "niedziela"
[116]  "niedziela"  "niedziela"     "niedziela" "piątek"    "środa"
[121]  "sobota"     "poniedziałek"  "środa"     "niedziela" "niedziela"
[126]  "niedziela"
```

Jest wiele dni weekendowych. Możliwe, że wolno nam je traktować jak dni biznesowe. Ale jest jeden przykład bez dni weekendowych w naszym zbiorze danych. Ten prosty kod pomoże to wykazać:

```
> for(i in 1:length(loss.data.object$blines)){
+ for(j in 1:length(loss.data.object$rcateg)){
+ y<- read.loss(b = i, r = j, loss.data.object)
+ if(dim(y)[1]!=0){
+ w<- weekdays(as.Date(y[,1]))
+ if(!is.element("niedziela", w)&!is.element("sobota", w)){
+ print(paste(loss.data.object$blines[i], loss.data.object$rcateg[j],i,j))}
+ }
+ }}

[1] "Retail Banking Execution, Delivery & Process Management 6 5"
```

Dla danych `i`, `j`, mamy `y` będące stratami z `i`-tej linii biznesowej i `j`-ej kategorii ryzyka. Potem sprawdzamy, czy y jest niepusty; jeżeli zawiera jakieś straty, to sprawdzamy ich daty, zmieniamy je na dni tygodnia i szukamy weekendów. Jeśli jakieś są, wyświetlamy nazwy i numery kategorii ryzyka i linii biznesowej.
Jak widać, jest tylko jedna komórka w tabeli `loss.matrix.image` bez dni weekendowych. Jest to szósta linia biznesowa i piąta kategoria ryzyka. Rzut oka na tabelę mówi nam, że mamy tam tylko 15 strat. Wydaje się to zbyt mało aby zdecydować, czy weekendy traktować w tej komórce jak dni biznesowe, czy nie. Jeżeli moglibyśmy zdecydować, że odpowiedź brzmi „nie”, moglibyśmy użyć opcji `wknd`, która jest przeznaczona tylko dla `period = "days"`; oczywiście nie ma sensu używać jej dla innych okresów.

```
> x65<- read.loss(b = 6, r = 5,loss.data.object)
> x65

     First_Date_of_Event Gross_Loss_Amount
63            2002-02-20           6118.85
165           2002-04-23           7942.78
622           2003-01-21           3325.98
711           2003-03-06           3405.60
984           2003-07-01           1426.88
1211          2003-10-14           1083.44
1325          2003-12-03           3284.29
1444          2004-01-28           6855.69
1956          2004-08-09           1894.01
1957          2004-08-09          37248.23
2141          2004-10-19          13553.71
2249          2004-12-03           5690.19
```

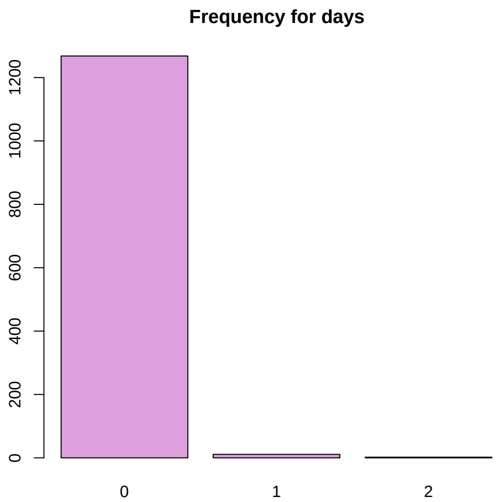
```
2496              2005-03-02               1570.91
2500              2005-03-02               3062.20
3032              2005-08-23              59086.99
```

Sprawdźmy daty:

```
> weekdays(as.Date(x65[,1]))

 [1]  "środa"   "wtorek"   "wtorek"  "czwartek"      "wtorek"
 [6]  "wtorek"  "środa"   "środa"    "poniedziałek"  "poniedziałek"
[11]  "wtorek"  "piątek"  "środa"    "środa"         "wtorek"
```

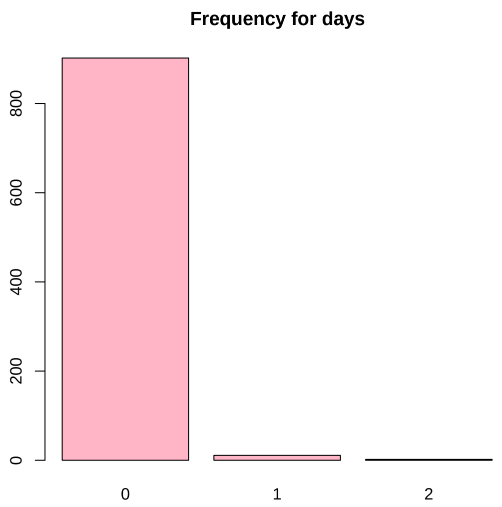
Jaka jest więc różnica pomiędzy histogramami z weekendami i bez? Jak wybrać właściwy? To zadanie raczej dla menadżera ryzyka, my po prostu zobaczymy efekt:

```
> h1 <- hist.period(x65,"days",col = "plum")$y
> h1

   0    1    2
1268   11    2
```



Rysunek 4.2: Histogram dla x65

```
> h2 <- hist.period(x65,"days",col = "pink1",wknd= F)$y
> h2

  0   1   2
902  11   2
```



Rysunek 4.3: Histogram dla x65; wknd=F

Naturalnie, ilość dni z dodatnimi liczbami strat jest taka sama, niezależnie od `wknd`. Obliczmy:

```
> dates<- as.Date(x65[,1])
> max(dates) - min(dates) +1

Time difference of 1281 days
```

Różnica czasu to ilość dni między początkową i końcową datą, wliczając zarówno datę początkową, jak i końcową.

Oczywiście:

```
> sum(h1)

[1] 1281
```

...co jest równe `max(dates) - min(dates) +1`.

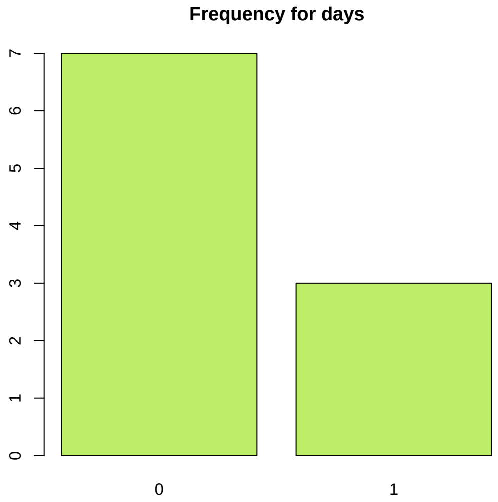
Teraz jest właściwa pora na ostrzeżenie. Weźmy pewne daty:

```
> new.dates <- c("2010-01-01", "2010-01-03", "2010-01-10")
> weekdays(as.Date(new.dates))

[1]  "piątek"  "niedziela"  "niedziela"
```

Teraz złączmy je z jakimikolwiek danymi stratami (tylko ze względu na to, że nasze dane muszą być dwukolumnowe: daty i straty).

```
> a<- cbind(new.dates,1:3)
> hist.period(a,"days", col =  "darkolivegreen2")$y
```

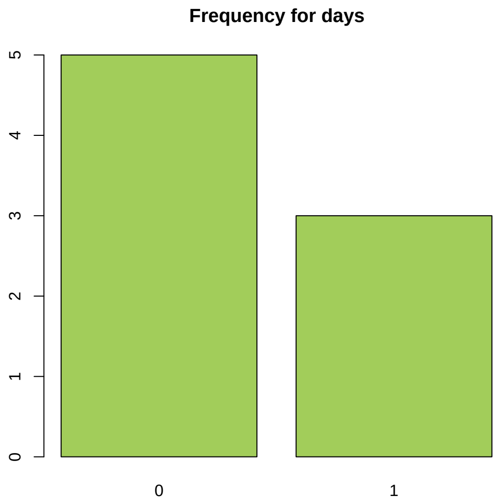
```
0 1
7 3
```



Prawidłowo: 7 dni bez strat, 3 ze stratami, razem 10. Mamy 10 dni (od "2010-01-01" do "2010-01-10") i 7 z nich jest bez strat. Używając `wknd = F` kiedy niektóre daty strat są dniami weekendowymi (w naszym przypadku są to "2010-01-03" i "2010-01-10") jest oczywiście niepoprawne i powoduje dalsze błędy. W następnym przykładzie mamy:

```
> hist.period(a,wknd=F, col =  "darkolivegreen3")$y
```

```
0 1
5 3
```



Dni są domyślne, więc nie trzeba ich podawać.
Mamy razem 8 dni, co jest błędem, gdyż z 10 dni obejmujących nasz okres musimy wyrzucić 4 dni:

```
> weekdays(seq(as.Date(new.dates[1]), length.out=10, by="1 day"))
```

```
[1]  "piątek"  "sobota"    "niedziela"  "poniedziałek"  "wtorek"
[6]  "środa"   "czwartek"  "piątek"     "sobota"        "niedziela"
```

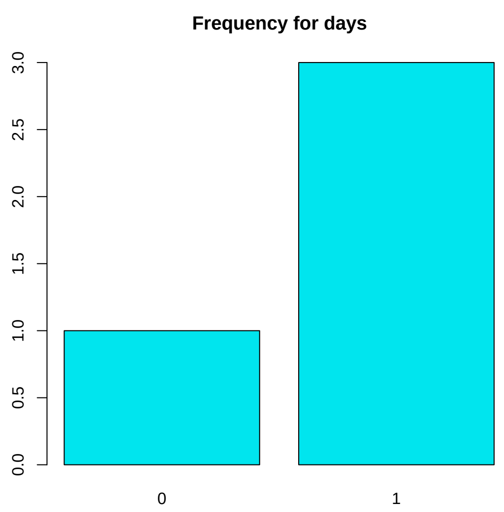
Te dwie straty w niedzielę nie powinny być liczone, więc wynik jest błędny.
. . . a może to być nawet bardziej irracjonalne:

```
> new.dates2 <- c("2010-01-01", "2010-01-03","2010-01-04")
> weekdays(as.Date(new.dates2))
```

```
[1]  "piątek"  "niedziela"  "poniedziałek"
```

```
> b <- cbind(new.dates2,1:3)
> hist.period(b, col = "turquoise2")$y
```

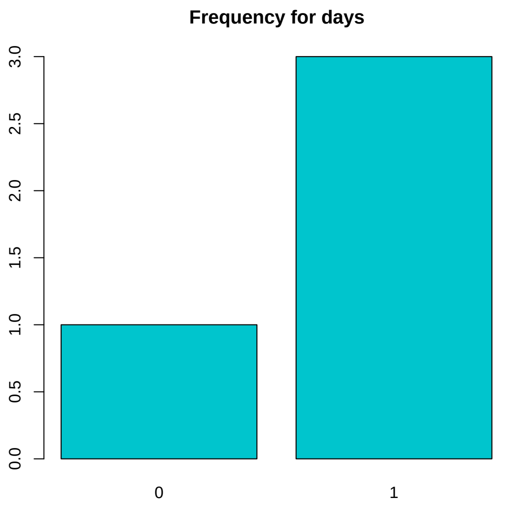
```
0 1
1 3
```



To oczywiście jest poprawne, `wknd = TRUE`.

```
> hist.period(b,wknd=F,col = "turquoise3")$y
```

```
0 1
1 3
```



Nie ma różnicy, pomimo `wknd = FALSE`!
Od `"2010-01-01"` do `"2010-01-04"` mamy 4 dni, nie licząc weekendu daje to tylko 2 dni, piątek (1 strata) i poniedziałek (1 strata), a więc nie może być łącznie trzech dni z jedną stratą, bo daje to sprzeczność.

To wszystko pokazuje, że opcji `wknd` należy używać ostrożnie i, być może, iż powinniśmy zwrócić szczególną uwagę na tygodnie aby zweryfikować nasze teorie. Oczywiście, mamy także okresy kwartalne i miesięczne, ale jak widzimy, kształt naszych histogramów nie wydaje się być bardzo regularny, gdyż mamy mniej danych. Sprawdźmy:

```
> min(as.Date(loss.data.object$losses[,3]))
[1] "2002-01-03"
> max(as.Date(loss.data.object$losses[,3]))
[1] "2006-02-12"
```

Te dwie daty to najwcześniejsza i najpóźniejsza data z naszego zbioru danych.

Pierwszym kwartałem, który się pojawia, jest kwartał rozpoczynający się `"2002-01-01"`, a ostatnim - rozpoczynający się `"2006-01-01"`; daje to tylko 17 kwartałów.
Należy być świadomym, że `hist.period()` przyjmuje, że dane strat były zbierane w całkowitych okresach, tzn. w naszym przypadku dane były rejestrowane od `"2002-01-01"` do `"2006-01-01"`, i daty te powinny być podane jako argumenty `begin` i `end`. Jeśli nie są podane, `begin` i `end` stają się datami pierwszej i ostatniej straty, i tak właśnie staje się w naszym przykładzie. Możemy powtórzyć tę samą argumentację dla miesięcy, ale oczywiście mamy tam więcej danych, zaś tygodnie wydają się raczej wiarygodne, może nawet bardziej niż dni, gdzie mamy opcje `wknd` i `crt` (korekta dla świąt, niewykorzystana w przykładzie).

4.2. Dopasowywanie rozkładu częstości strat

Teraz chcemy dopasować rozkład strat do naszych danych. Lepiej jest mieć możliwie duży zbiór danych dla celów naszego przykładu, gdyż dopasowywanie rozkładu do 3 punktów nie wydaje się sensowne, ale 500 punktów wydaje się już rozsądniejszą liczbą. Zobaczmy:

```
> D[,4,]

     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    0  806    0    0    0  102   80
[2,]    0    0    0    0  139    0  173
[3,]    0  285    0   61    0   54   64
[4,]    0    0  128    0   38   32    0
[5,]    0    0  328  137  126    0  125
[6,]    0    0    0    0   15    0   20
[7,]    0  459    0    0  119    0  126
[8,]    0    0  192    0    0    0    0
```

To pokazuje, jak wiele strat mamy dla danej linii biznesowej i kategorii ryzyka. Oczywiście mamy maksimum w `D[1,4,2]` (to oznacza pierwszą linię biznesową i kategorię ryzyka):

```
> max(D[,4,])

[1] 806
```

Wybierzemy tę linię i tę kategorię.

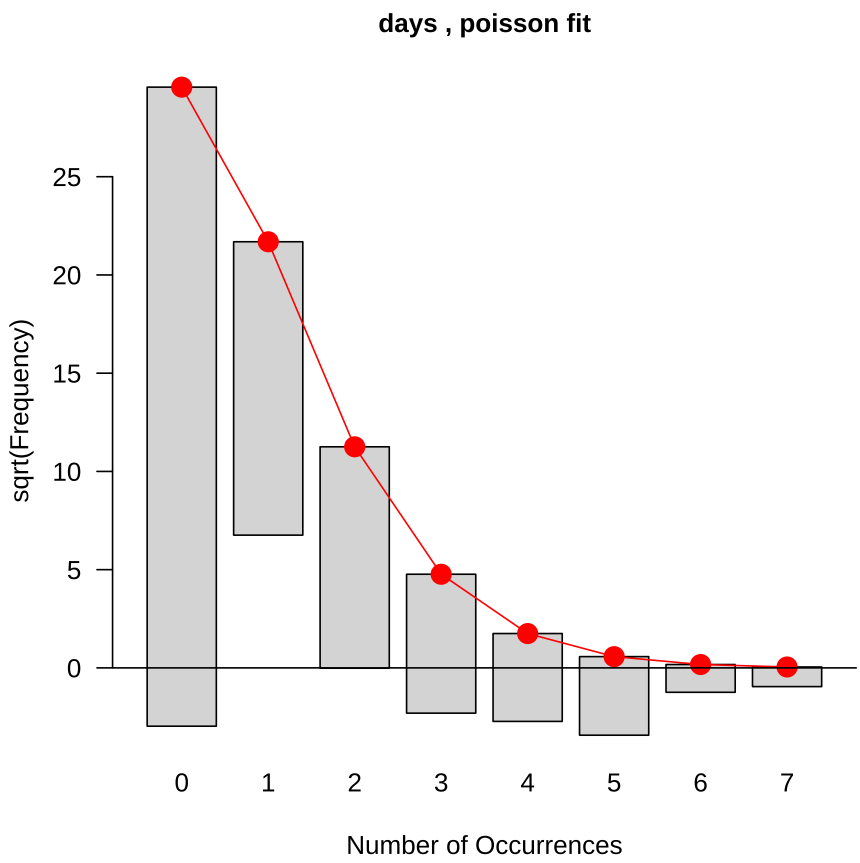
4.2.1. Komórka „Agency Services/Clients, Products & Business Practices”

```
> x12<- read.loss(b=1,r=2,loss.data.object)
> dim(x12)

[1] 806   2
```

Dopasowanie rozkładu Poissona

Teraz chcemy dopasować rozkład `"poisson"` do naszych danych.

```
> y1<- root.period(x12,"days","poisson")

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 379.8419  6 6.012827e-79
```




```
> t <- y1$table; t

Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed       fitted
     0     1058 8.737622e+02
     1      223 4.704424e+02
     2      127 1.266455e+02
     3       50 2.272907e+01
     4       20 3.059391e+00
     5       16 3.294414e-01
     6        2 2.956243e-02
     7        1 2.273816e-03
```

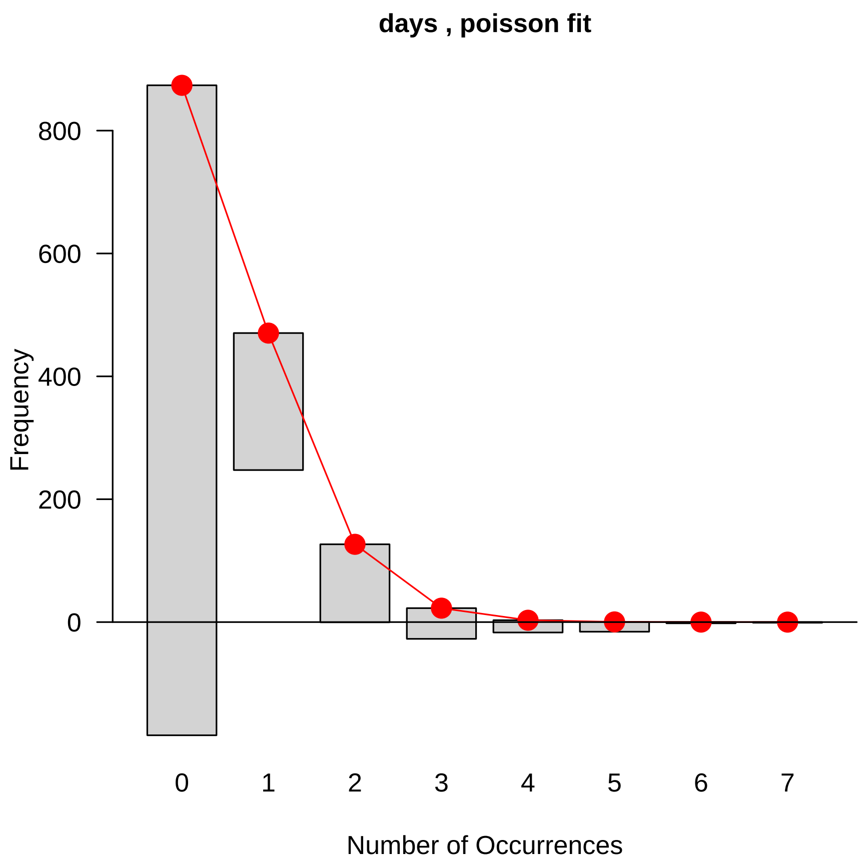
Jest on dopasowany bardzo słabo i można to stwierdzić patrząc na obrazek. Zobaczymy wartości naszej funkcji:

```
> par <- y1$param; par

$lambda
[1] 0.5384102
```

```
> p <- y1$p; p

[1] 6.012827e-79
```

Oczywiste jest, że ten rozkład nie jest dobrze dopasowany, ponieważ `p` jest bardzo małe. Musimy też być świadomi używania skali `sqrt` - należy pamiętać, że naprawdę wygląda to tak:

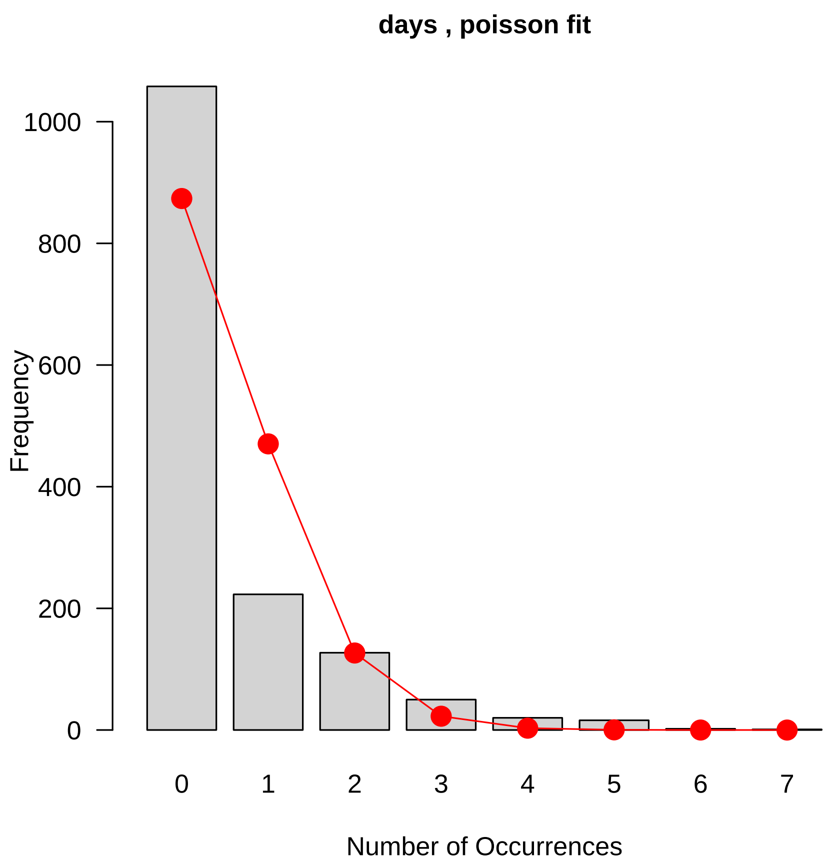
```
> y2 <- root.period(x12,"days","poisson",scale = "raw")

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 379.8419  6 6.012827e-79
```



Zwróćmy uwagę na skalę `"raw"`!

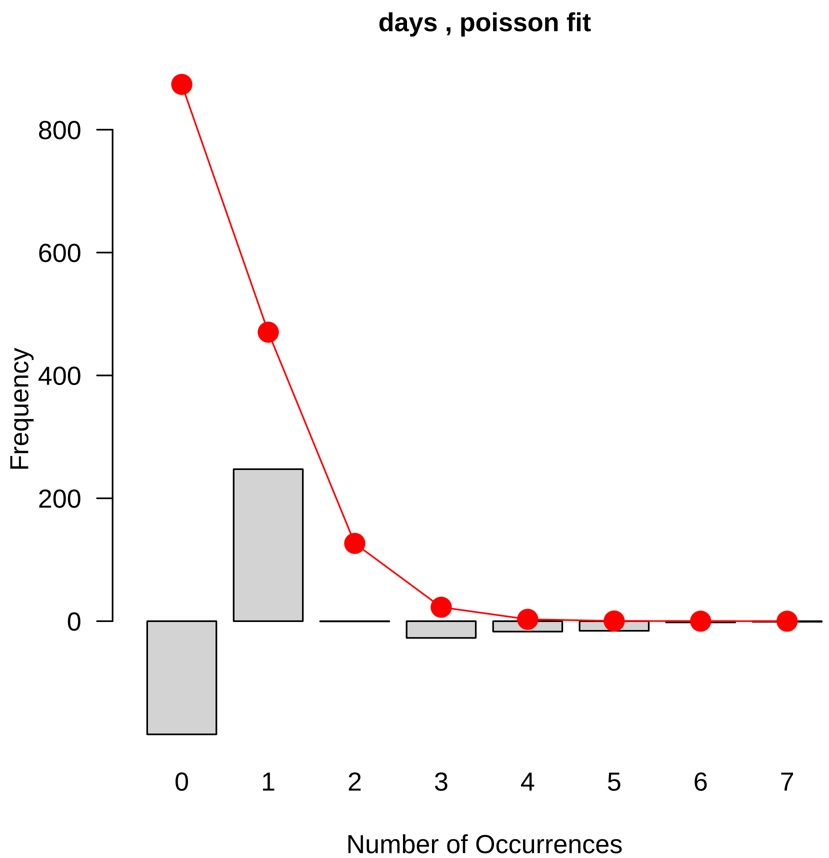
A teraz wypróbujmy inną opcję:

```
> y3<-root.period(x12,"days","poisson",scale = "raw",bar = "standing")

        Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 379.8419  6 6.012827e-79
```



To jest obrazek ze skalą `raw` i `standings` użytym dla słupków (słupki stojące, zamiast domyślnych `hanging`, wiszących).

```
> y4<-root.period(x12,"days","poisson",scale = "raw",bar = "deviation")

        Goodness-of-fit test for poisson distribution

                      X^2 df      P(> X^2)
Likelihood Ratio 379.8419  6 6.012827e-79
```



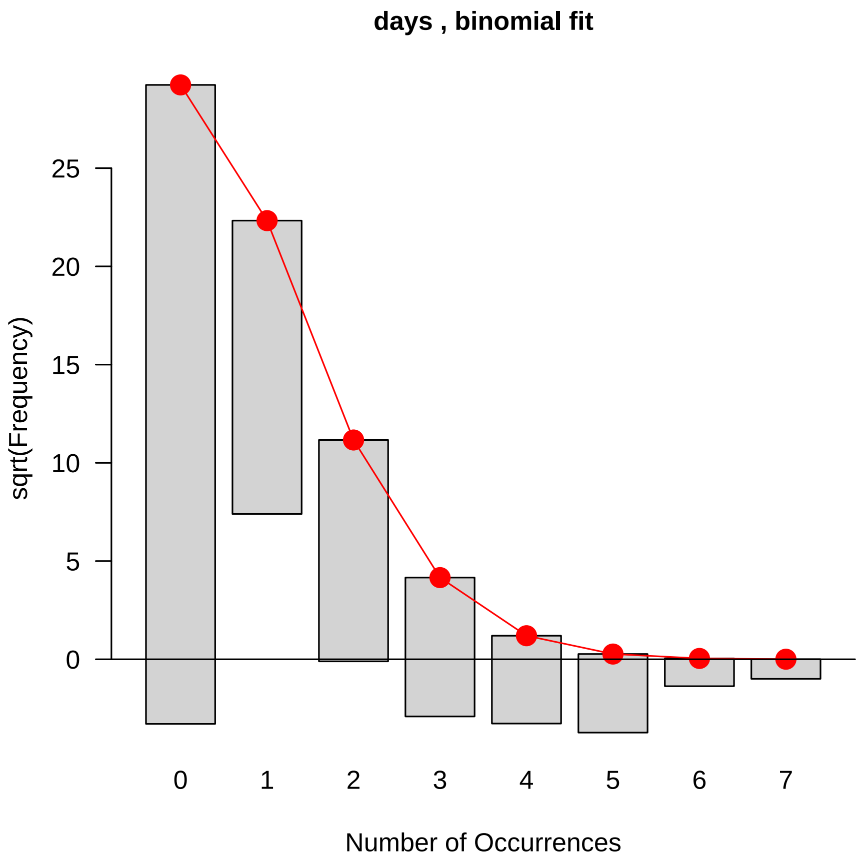
Tutaj zaś mamy pokazane odchylenia.

Definitywnie, nie jest to dobre dopasowanie. Niestety, rozkład `"binomial"` jest nie lepszy - lub nawet gorszy, patrząc na wartość `p`:

Dopasowanie rozkładu dwumianowego

```
> root.period(x12,"days","binomial")

        Goodness-of-fit test for binomial distribution

                      X^2 df       P(> X^2)
Likelihood Ratio 529.9351  6 2.983201e-111
$table

Observed and fitted values for binomial distribution
with parameters estimated by ‘ML’
```

```
 count observed       fitted
     0     1058 8.548920e+02
     1      223 4.986354e+02
     2      127 1.246460e+02
     3       50 1.731015e+01
     4       20 1.442363e+00
     5       16 7.211071e-02
     6        2 2.002868e-03
     7        1 2.384121e-05

$param
$param$prob
[1] 0.07691574

$param$size
[1] 7


$p
[1] 2.983201e-111
```



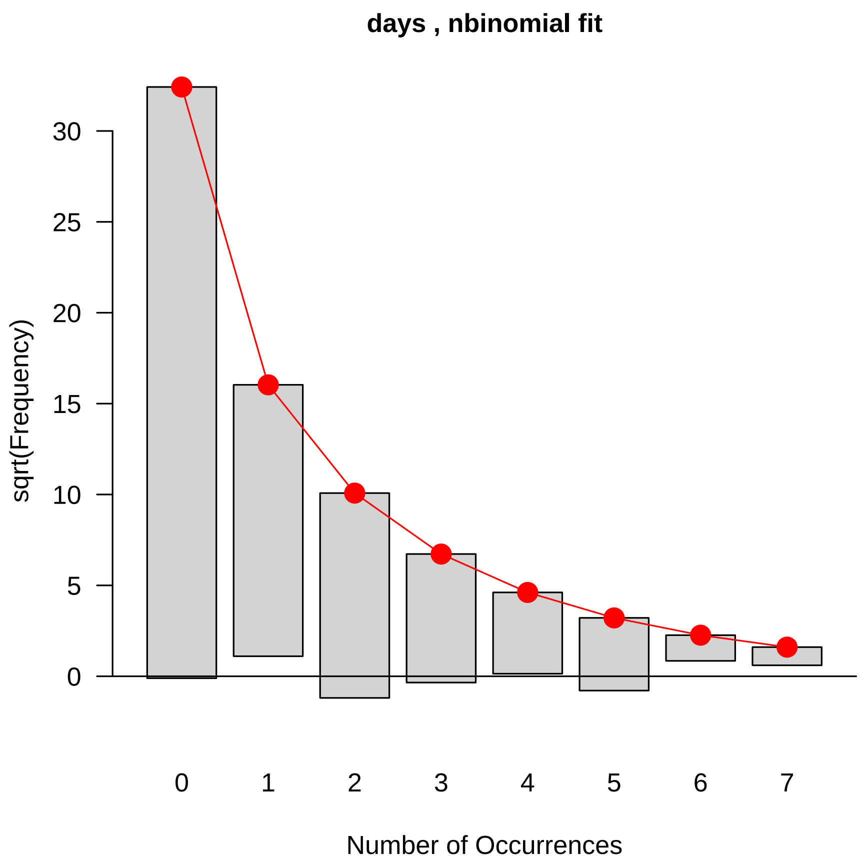
Wiadomość ostrzegawcza:
`warning("size was not given, taken as maximum count")`
pochodzi z `goodfit()` i jest wyświetlana zawsze, kiedy nie jest podany parametr `size`; zobacz pomoc dla `goodfit()`.

Rozkład nie jest dobrze dopasowany, co widać po niewielkiej wartości `p`.

Dopasowanie rozkładu ujemnego dwumianowego

A teraz wypróbujemy rozkład `"nbinomial"`:

```
> root.period(x12,"days","nbinomial")

         Goodness-of-fit test for nbinomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 23.01709  5 0.0003350356
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML'

 count observed      fitted
     0     1058 1051.085049
     1      223  257.118985
     2      127  101.585990
     3       50   45.231189
     4       20   21.273528
     5       16   10.325656
     6        2    5.115405
     7        1    2.570862

$param
$param$size
[1] 0.4483843

$param$prob
[1] 0.4544358


$p
[1] 0.0003350356
```



Pierwsze z tych ostrzeżeń brzmi: `1: In dnbinom\mu(x, size, mu, log) : NaNs produced` a reszta jest taka sama; są to ostrzeżenia pochodzące od `goodfit()` (faktycznie od `dnbinom()`).

Nasz rozkład ujemny dwumianowy (`"nbinomial"`) jest lepszy, niż rozkłady: Poissona (`"poisson"`) i dwumianowy (`"binomial"`) dla tych danych.

Teraz nadszedł czas na zaprezentowanie dobrze dopasowanego rozkładu.

4.2.2. Komórka „Corporate Finance/Execution, Delivery & Process Management”

Weźmy inne dane:

```
> x45 <- read.loss(b=4,r=5,loss.data.object)
> dim(x45)
```

```
[1] 38  2
```

Sprawdźmy parametry.

```
> fit <- {}
> i = 1
> for(period in c("days","weeks","months","quarters")){
```

```
+ dist <- c("poisson","binomial","nbinomial")
+ p1 <- root.period(x45,period,"poisson")$p
+ p2 <- root.period(x45,period,"binomial")$p
+ p3 <- root.period(x45,period,"nbinomial")$p
+ p <- c(p1,p2,p3)
+ number <- which(p == max(p))
+ fit[[i]] <- paste(dist[number],period); i <- i +1
+ }
```

Nasze `fit` jest listą rozkładów do wyboru. Potrzebujemy maksymalnej wartości `p`. Dla każdej wartości `period` jest wybierany jeden rozkład z `dist`.

```
> fit
```

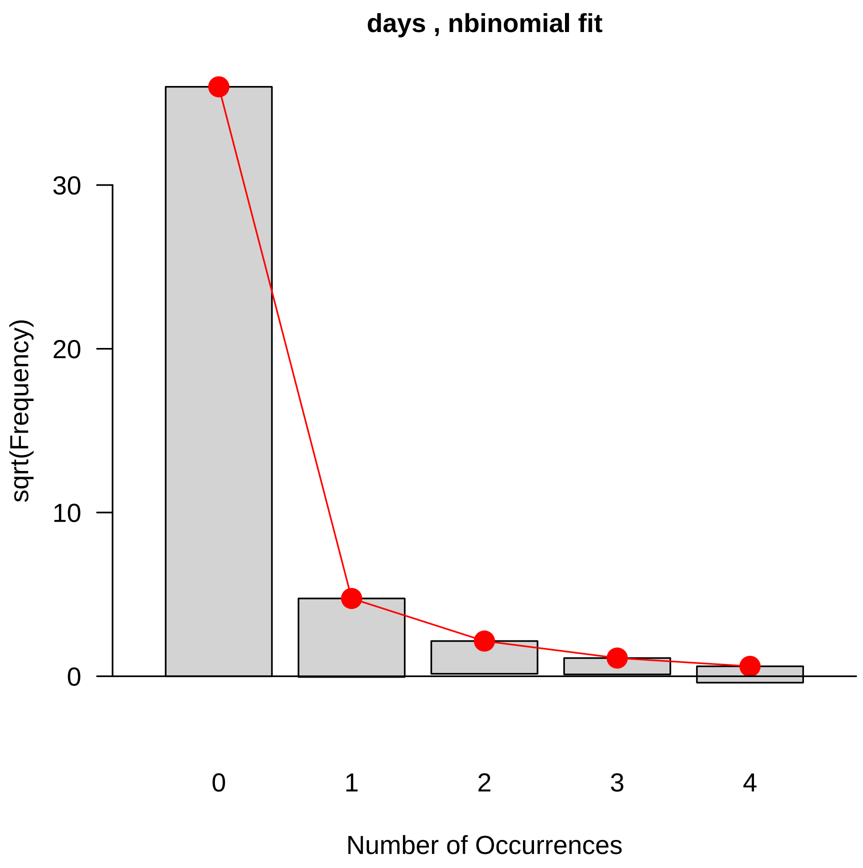
```
[1] "nbinomial days"   "nbinomial weeks"  "nbinomial months" "poisson quarters"
```

Jest teraz oczywiste, które rozkłady należy wybrać dla różnych okresów; `"nbinomial"` dla dni, tygodni i miesięcy; `"poisson"` dla kwartałów.
Dla dni:

Dni

```
> root.period(x45,"days","nbinomial")
```

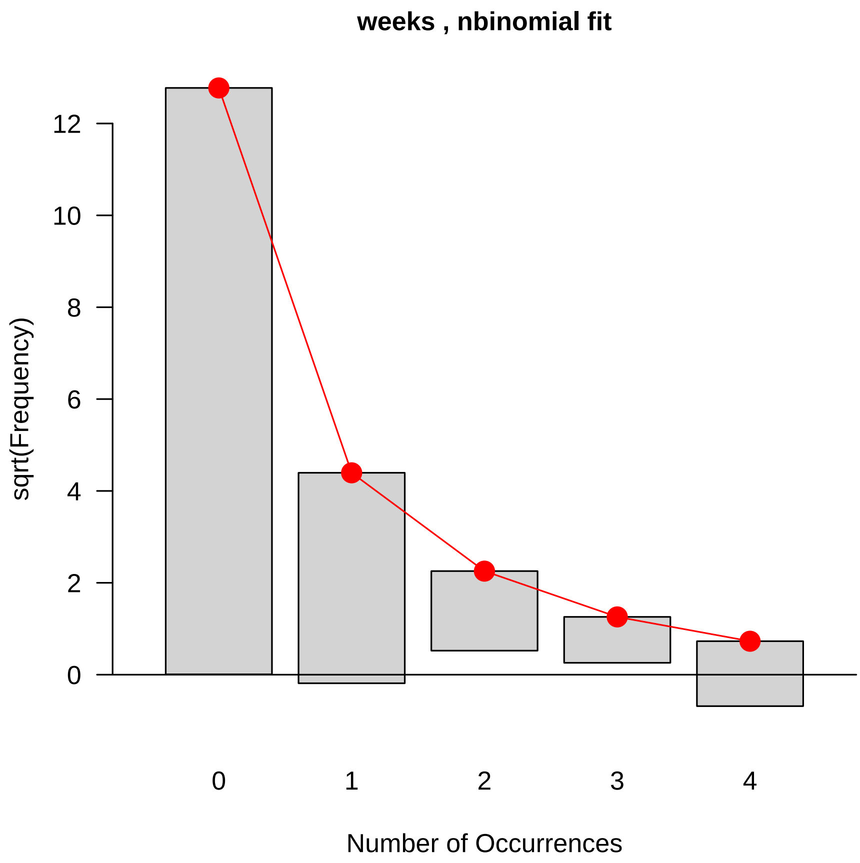
```
         Goodness-of-fit test for nbinomial distribution

                      X^2 df P(> X^2)
Likelihood Ratio 1.227877  2 0.541215
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by ‘ML’

 count observed       fitted
     0     1296 1296.0023507
     1       23   22.5832754
     2        4    4.6294474
     3        1    1.2384585
     4        1    0.3700255

$param
$param$size
[1] 0.04438856

$param$prob
[1] 0.6074363


$p
[1] 0.541215
```



Tygodnie

```
> root.period(x45,"weeks","nbinomial")

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 4.342171  2 0.1140537
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML'

 count observed      fitted
     0      163 163.1849680
     1       21  19.3219847
     2        3   5.0854239
     3        1   1.5838878
     4        2   0.5315334

$param
$param$size
```

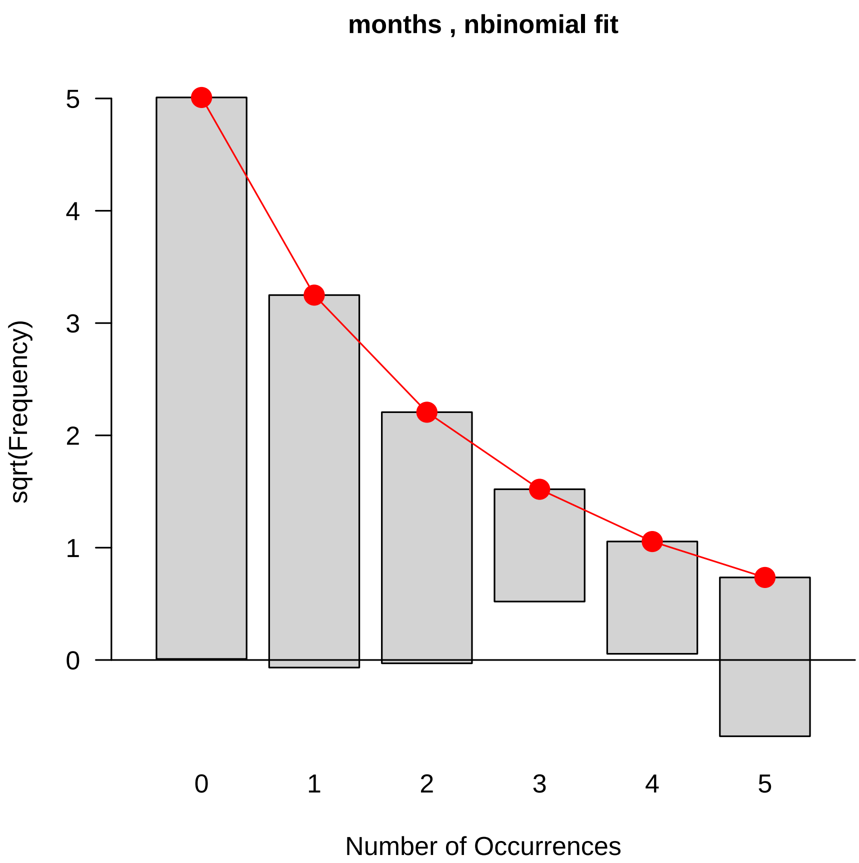
```
[1] 0.2902222

$param$prob
[1] 0.5920181


$p
[1] 0.1140537
```



Miesiące

```
> root.period(x45,"months","nbinomial")

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 4.333902  3 0.2275930
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by `ML'
```

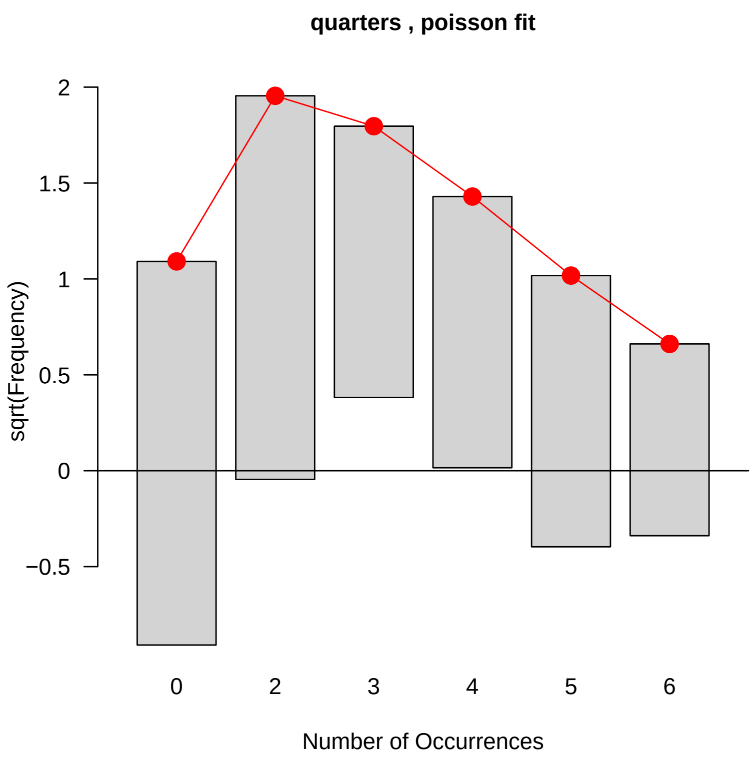
```
 count observed      fitted
     0       25 25.0902201
     1       11 10.5566972
     2        5  4.8692328
     3        1  2.3116422
     4        1  1.1130419
     5        2  0.5404305

$param
$param$size
[1] 0.838577

$param$prob
[1] 0.4982578


$p
[1] 0.2275930
```



Kwartały

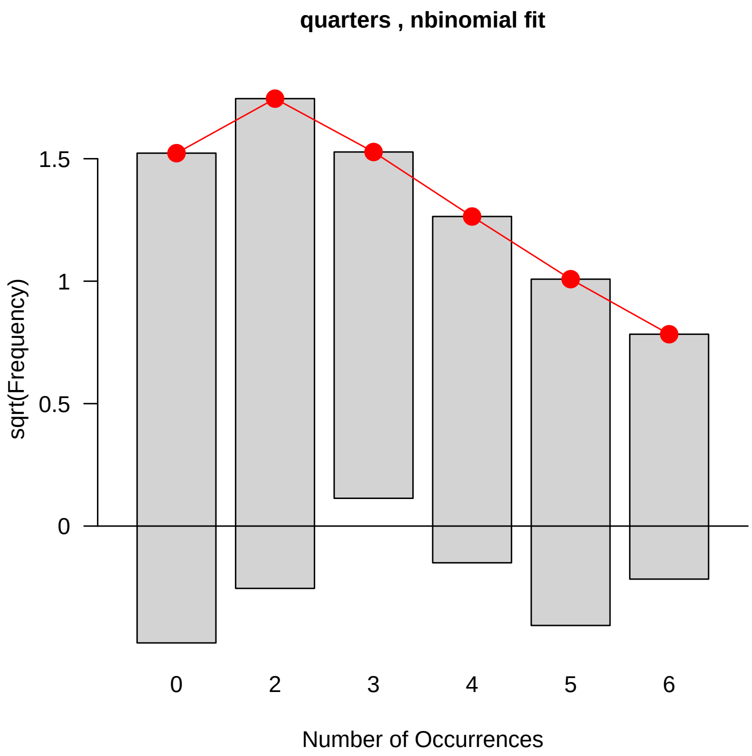
> root.period(x45,"quarters","poisson")

```
        Goodness-of-fit test for poisson distribution

                         X^2 df     P(> X^2)
Likelihood Ratio 12.34514  4 0.01496106
$table

Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed    fitted
     0        4 1.1909090
     2        4 3.8214946
     3        2 3.2270399
     4        2 2.0437919
     5        2 1.0355212
     6        1 0.4372201

$param
$param$lambda
[1] 2.533333


$p
[1] 0.01496106
```



Wartość `p` nie wydaje się być bardzo dobra w porównaniu z tymi dla dni, tygodni lub miesięcy, ale jak powiedzieliśmy, kwartały nie muszą dawać wiarygodnych dopasowań, gdyż jest ich zbyt mało, także dane powinny być podane dla pełnych okresów, a nie są.
Właściwie nie ma wielkiej różnicy pomiędzy dopasowaniami, gdy zamiast `"poisson"` spróbujemy dopasować `"nbiniomial"`:

```
> root.period(x45,"quarters","nbinomial")

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
Likelihood Ratio 10.50511  3 0.01472627
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by ‘ML’

 count observed    fitted
     0        4 2.3188191
     2        4 3.0463495
     3        2 2.3334455
     4        2 1.5980008
     5        2 1.0165374
     6        1 0.6136513

$param
$param$size
[1] 3.206557

$param$prob
[1] 0.5586445


$p
[1] 0.01472627
```



Dla Poissona `p` wyniosło 0.01496106, więc różnica to zaledwie 0.00023479.

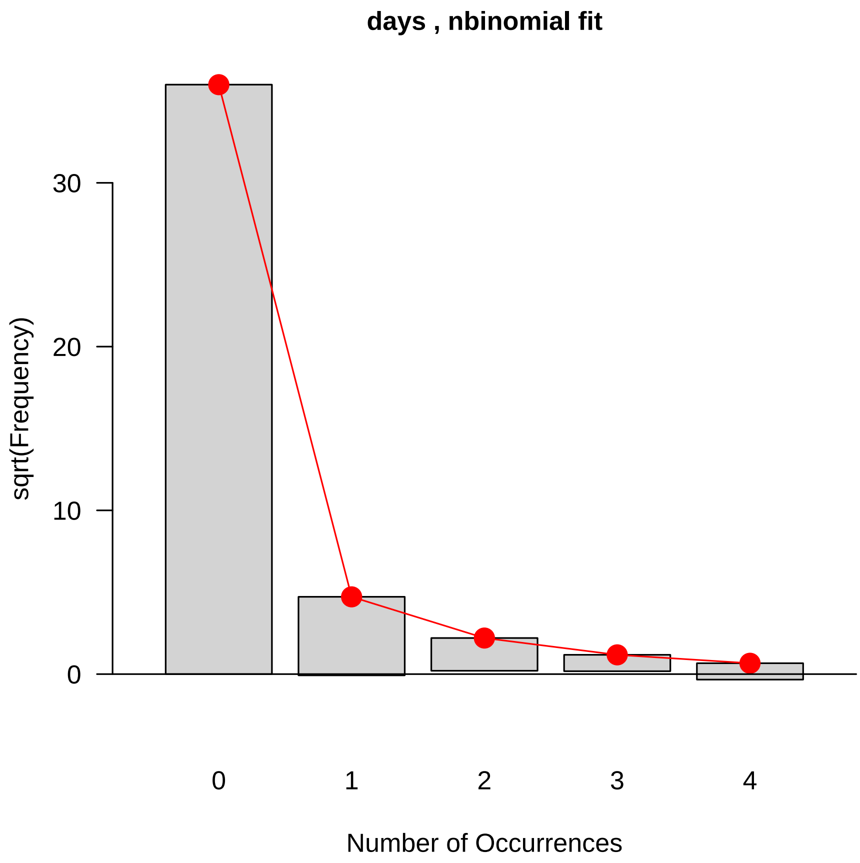
Metody „Maximum Likelihood" i „Minimum Chi-squared"

Jest jeszcze jedna rzecz, na którą należałoby zwrócić uwagę. Rozkłady są dopasowywane przy zastosowaniu metody największej wiarogodności - `"ML"` (Maximum Likelihood) lub przez minimalizowanie statystyki χ^2 - `"MinChisq"` (Minimum Chi-squared), przy czym domyślnie jest `"ML"`.

Zobaczmy częstość strat dla `x45` dopasowaną metodą `"MinChisq"`, gdzie `period` = `days`:

```
> root.period(x45,"days","nbinomial",method = "MinChisq")

         Goodness-of-fit test for nbinomial distribution

              X^2 df P(> X^2)
Pearson 0.4405477  2 0.802299
$table

Observed and fitted values for nbinomial distribution
with parameters estimated by 'MinChisq'

 count observed       fitted
     0     1296 1295.8037516
     1       23   22.2756990
     2        4    4.8601571
```

```
     3          1     1.3860155
     4          1     0.4416922

$param
$param$size
[1] 0.04101081

$param$prob
[1] 0.5808266


$p
[1] 0.802299
```



Wartość `p` jest większa niż przedtem i jest to różnica znaczna, ale tabela `table` jest niemal identyczna. Mamy też pewne ostrzeżenia.

Podsumowując, rozkład `"nbinomial"` raczej satysfakcjonująco dopasowuje się do naszych danych `x45`.

Rozdział 5

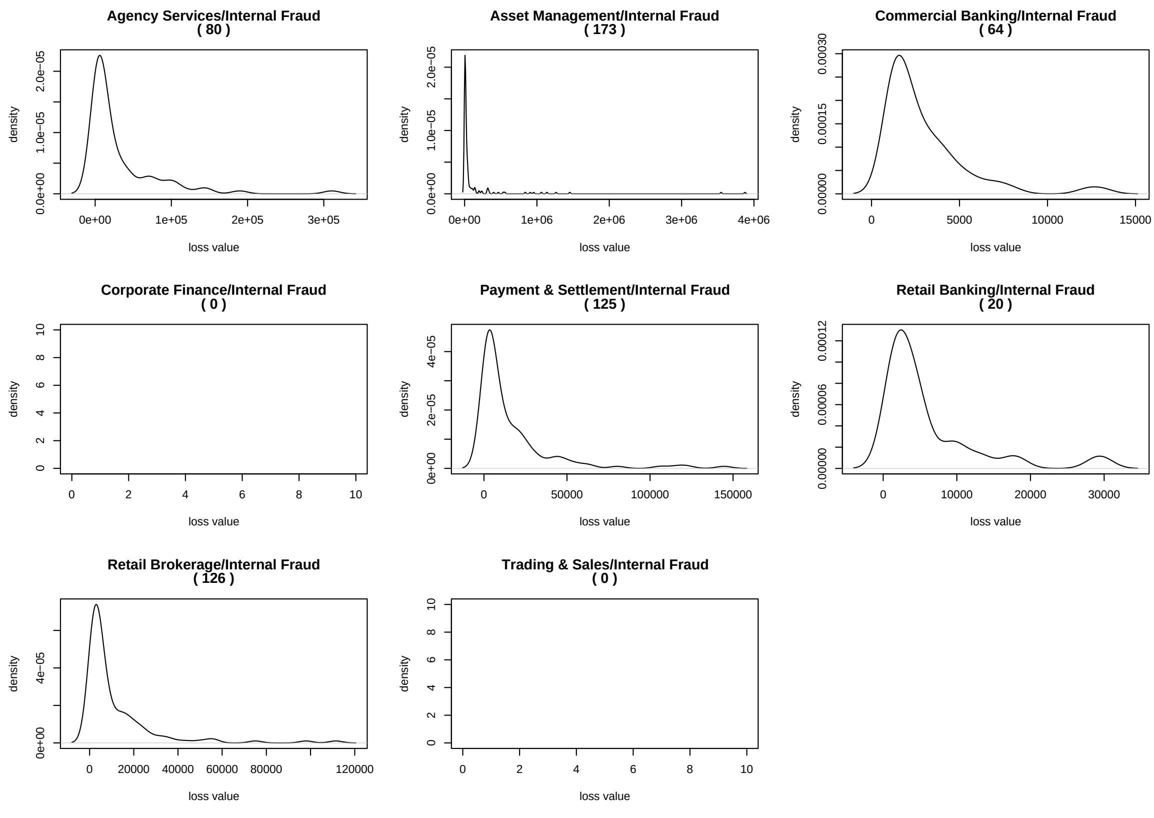
Dotkliwość strat

Mając rozkład częstości spróbujemy dopasować również rozkład dotkliwości strat.

5.1. Gęstość

Na początku, narysujmy gęstość dla pewnych komórek. Mamy funkcję `loss.denstiy()`, która rysuje wszystkie gęstości dla danej kategorii ryzyka (linii biznesowej) i wszystkich linii biznesowych (kategorii ryzyka).

```
> loss.density( a=1,b=7,loss.data.object)
```



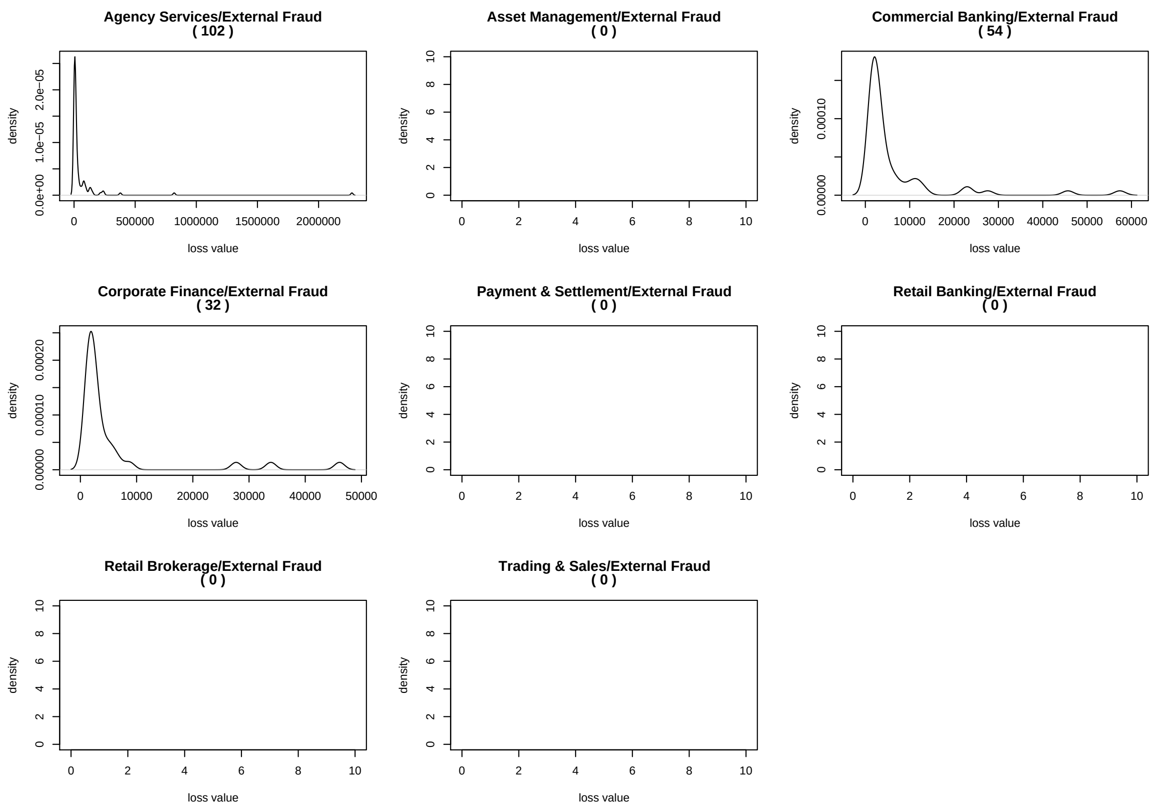
Rysunek 5.1: Gęstości strat dla 7 kategorii ryzyka i wszystkich linii biznesowych

`a` = 1 oznacza gęstość dla wszystkich linii biznesowych (`loss.data.object$blines`), a `b` = 7 oznacza kategorię ryzyka `"Internal Fraud"`; `period` = `"none"` (domyślny).

Jak łatwo sprawdzić, nie ma danych w niektórych komórkach „linia biznesowa/kategoria ryzyka” (patrz `loss.matrix.image(D,loss.data.object$blines,loss.data.object$rcateg)`); ilości strat są wyświetlane nad rysunkami.

Może jednak zdarzyć się tak, że będzie wiele pustych obrazków - np. `loss.density(1,1,loss.data.object)`: tu wcale nie ma danych, tak więc opcja `no` (nierysowanie pustych obrazków) może być użyteczna. Zobaczmy różnicę:

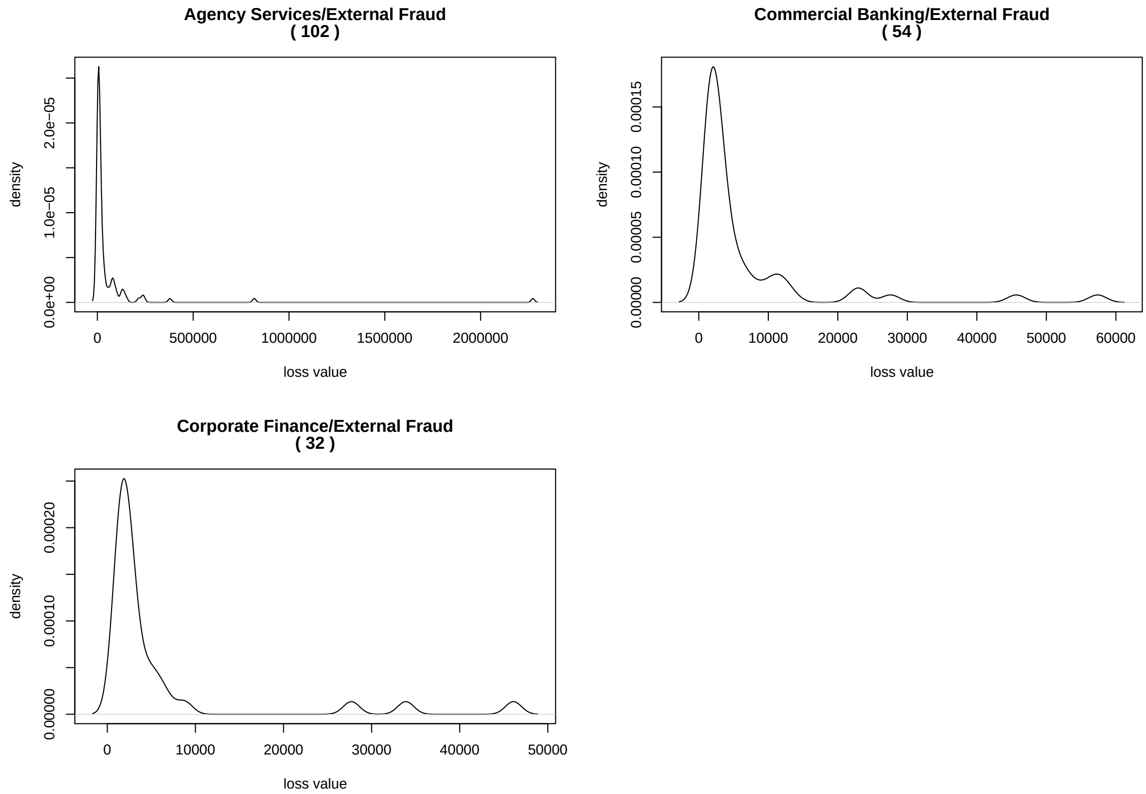
```
> loss.density(a=1,b=6,loss.data.object)
```



Rysunek 5.2: Gęstości strat dla szóstej kategorii ryzyka i wszystkich linii biznesowych

Kategoria ryzyka to `"External Fraud"`. Jest tu pięć pustych obrazków. Możemy je wyłączyć.

```
> loss.density(a=1,b=6,loss.data.object,no=TRUE)
```

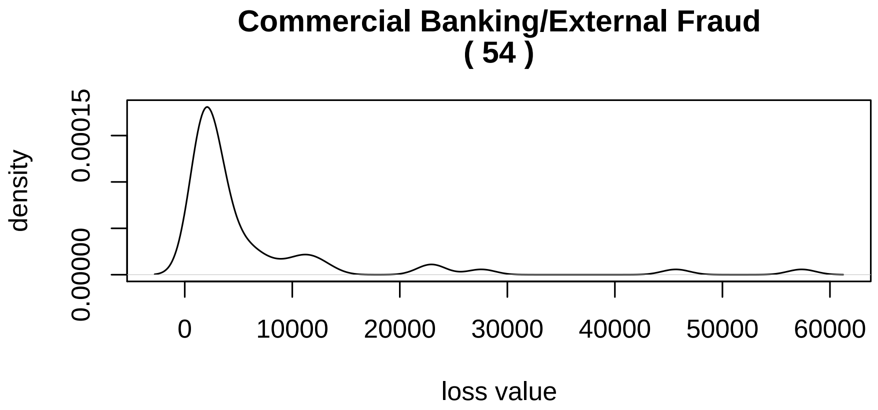


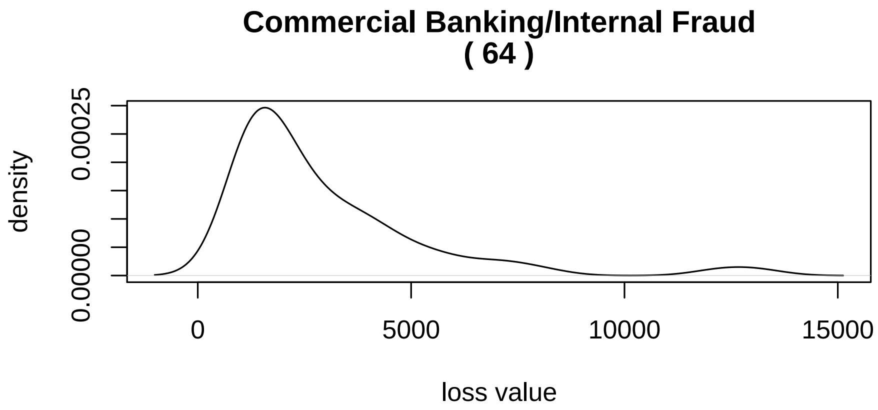
Rysunek 5.3: Gęstości strat dla szóstej kategorii ryzyka i wszystkich linii biznesowych z wyłączeniem pustych komórek

Teraz rysunek jest o wiele bardziej przejrzysty.

Jest także możliwość wywołania tej funkcji dla jednej kategorii ryzyka (linii biznesowej) i pewnych wybranych pozycji z linii biznesowych (kategorii ryzyka).
Np. weźmy trzecią linię biznesową (`"Commercial Banking"`) i wektor kategorii ryzyka składający się z szóstej i siódmej kategorii ryzyka (`"External Fraud"` i `"Internal Fraud"`).

```
> loss.density(a=2,b=3,loss.data.object,rnumb=c(6,7))
```





Rysunek 5.4: Gęstości strat dla trzeciej linii biznesowej i kategorii ryzyka z numerami z rnumb=c(6,7)

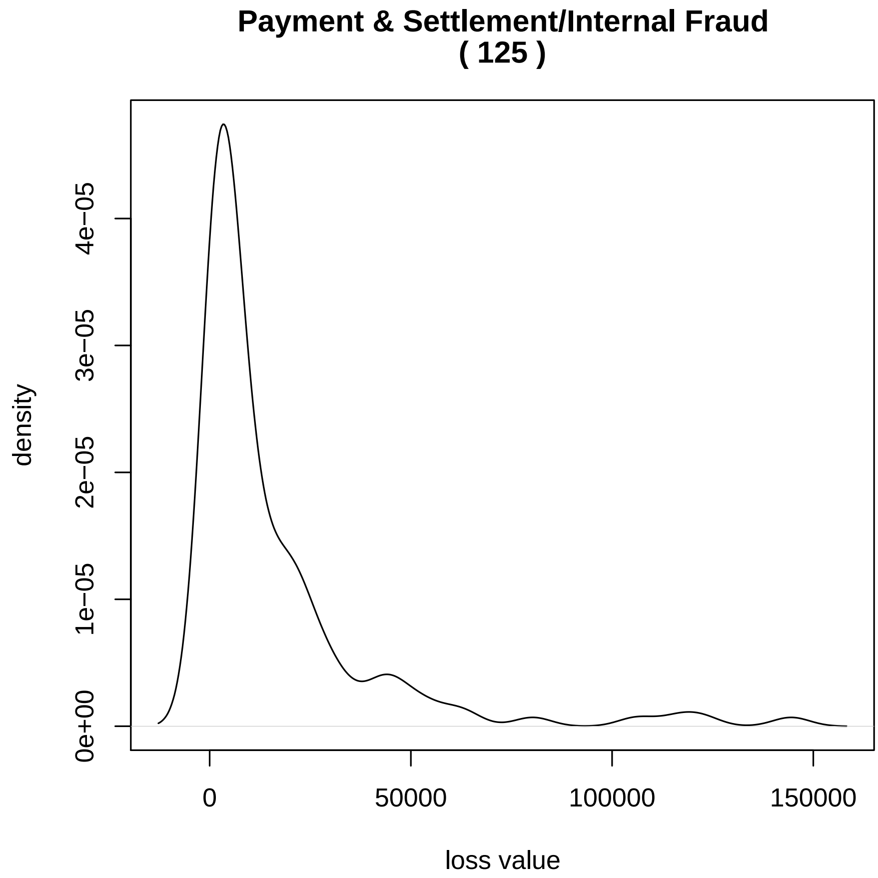
`a` = 2 oznacza, że rysujemy dla kategorii ryzyka (tu: `rnumb`) i wybieramy jedną linię biznesową, a jej numer to `b`=3.
Oczywiście bez sensu byłoby dawać jakiekolwiek `bnumb`, gdyż wybieramy tylko jedną linię biznesową i jest nią `b`. Instrukcje jak:

```
> loss.density(a=2,b=3,loss.data.object,rnumb=c(6,7),bnumb = c(11))
```

są formalnie niepoprawne (zauważmy, że nie ma w ogóle jedenastej linii biznesowej!), ale poprawne w wyniku (nie ma także ostrzeżenia związanego z nieistnieniem jedenastej linii biznesowej, ponieważ ta instrukcja zostaje pominięta). Otrzymujemy ten sam rysunek 5.4.

Teraz zobaczmy jak wygląda gęstość jednej tylko kategorii ryzyka i jednej linii biznesowej:

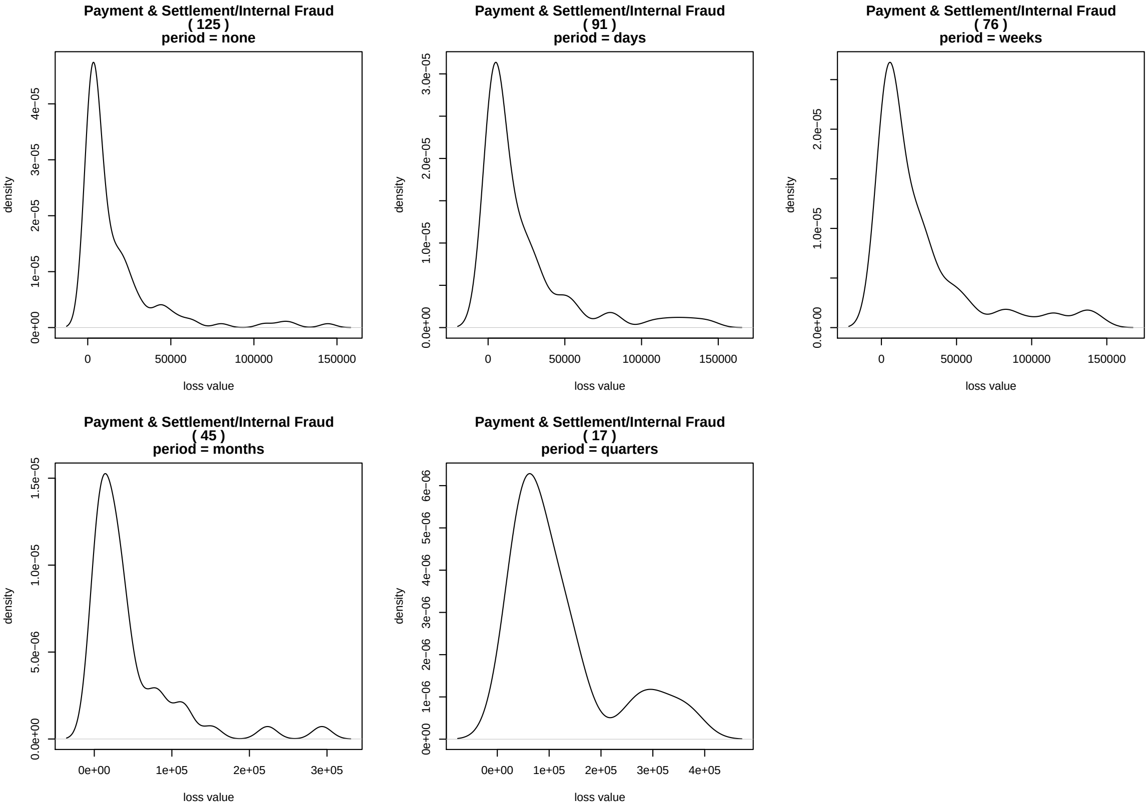
```
> loss.density(a=1,b=7,bnumb=c(5),loss.data.object)
```



Rysunek 5.5: Gęstość strat dla piątej linii biznesowej i siódmej kategorii ryzyka

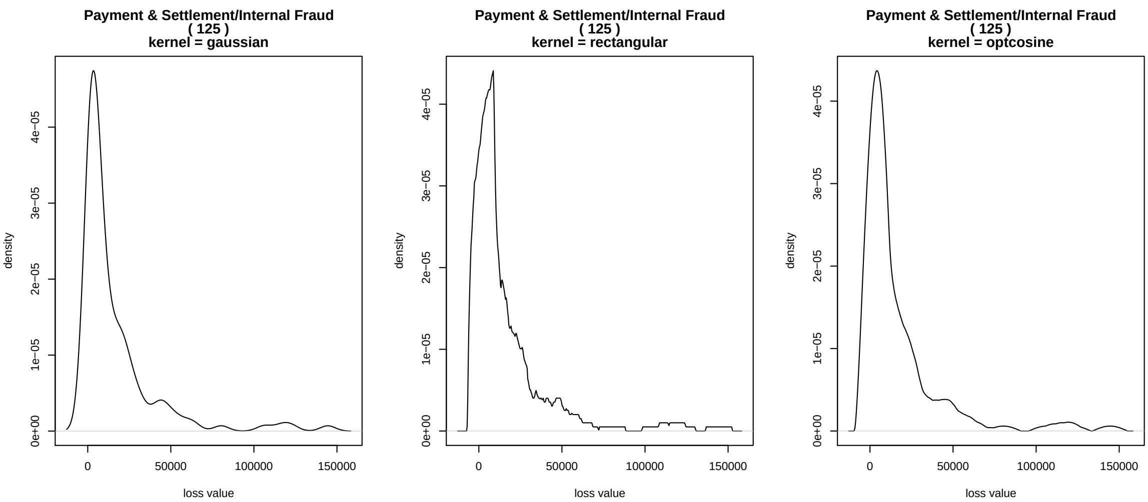
Kategoria ryzyka to `loss.data.object$rcateg[7]`, a ponieważ `bnumb=c(5)`; mamy tylko jedną linię biznesową, `loss.data.object$blines[5]`.

Teraz zmienimy `period` (który domyślnie jest `none`). Kod służący otrzymaniu tych rysunków otrzymano przez zmianę kodu `loss.density()` poprzez wstawienie „#” aby ukryć opcję `par(mfrow=c(n1,n2))` oraz dodanie `period` do `main`:

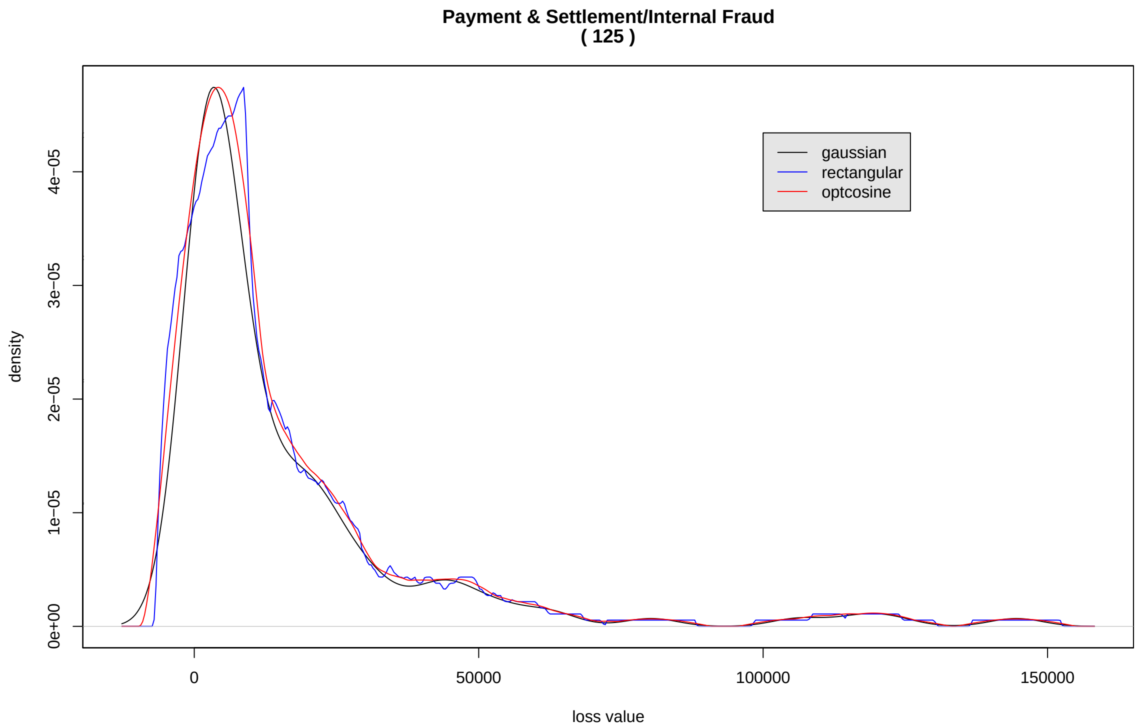


Rysunek 5.6: Gęstość strat dla piątej linii biznesowej i siódmej kategorii ryzyka; okresy to „none", „days", „weeks", „months" and „quarters"

Możemy także użyć pewnych opcji `density`, np. zmienić `kernel` - wybierzmy `"gaussian"` (domyślne), `"rectangular"` i `"optcosine"` (kod użyty do wygenerowania tych obrazków został otrzymany przez modyfikację kodu `loss.density`):



Na jednym obrazku:



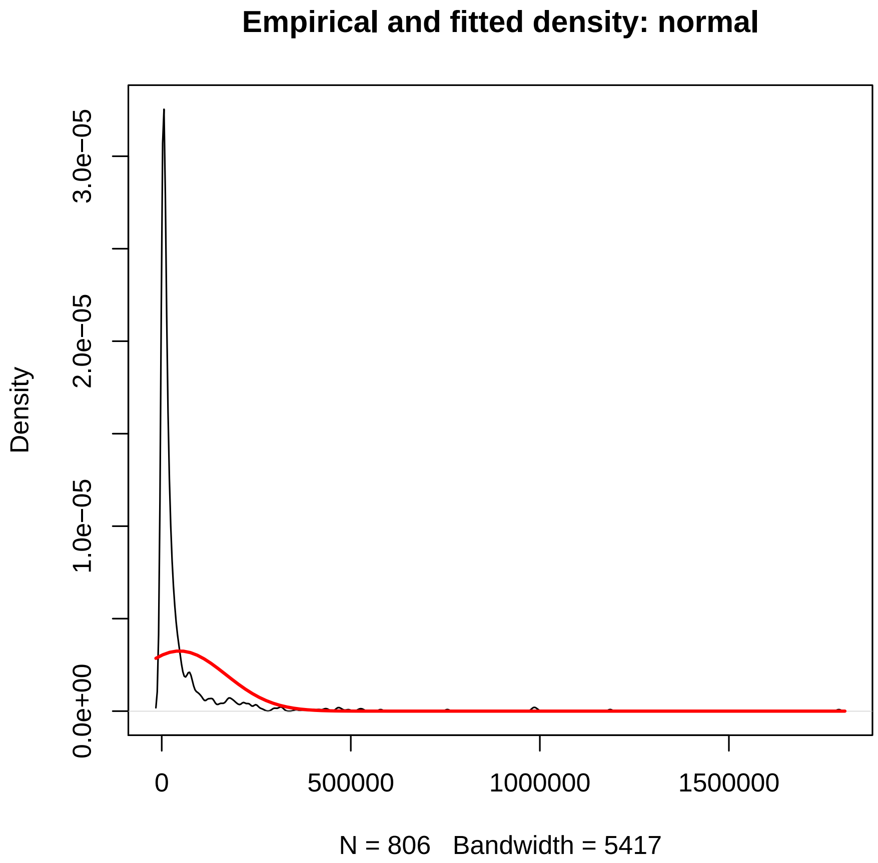
5.2. Dopasowywanie rozkładu dotkliwości strat

Teraz dopasujmy rozkłady dotkliwości.

5.2.1. Komórka „Agency Services/Clients, Products & Business Practices”

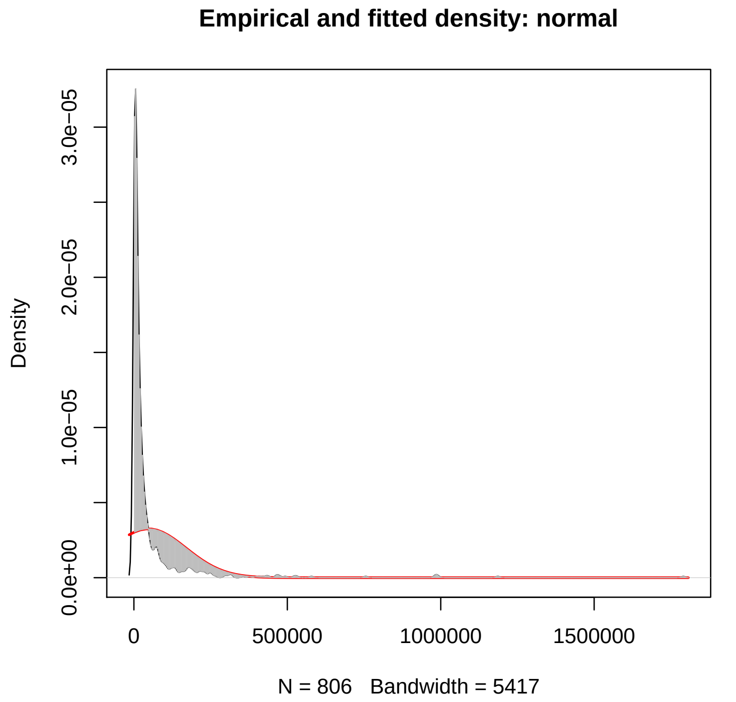
Użyjemy `loss.fit.dist()` aby dopasować rozkład `"normal"` do naszych danych:

```
> x12<- read.loss(b=1,r=2,loss.data.object)
> loss.fit.dist("normal",x12)

$mean
[1] 47079.52

$sd
[1] 122694.0

          ad
0.0002364375
```



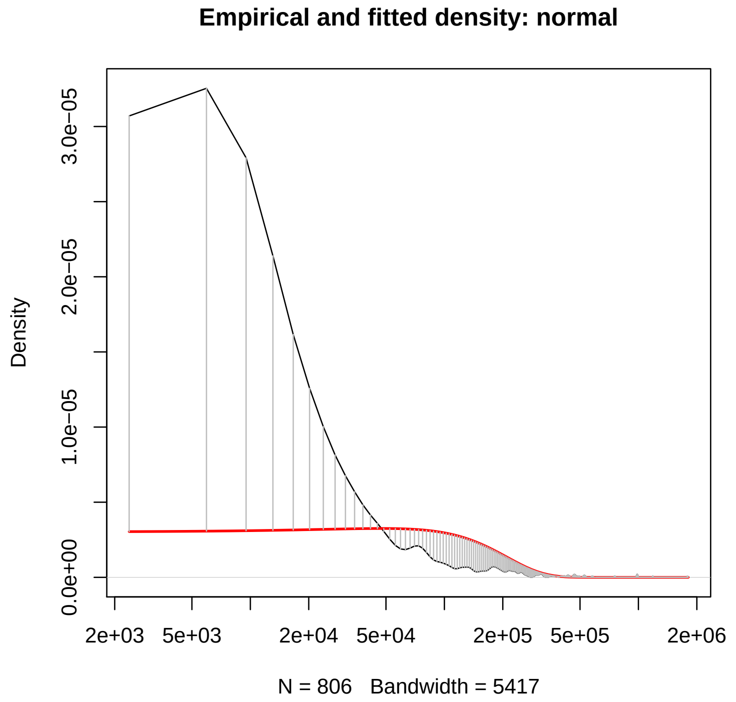
Jak widzimy, ten rozkład nie pasuje do naszych danych, ale chcielibyśmy mieć jednak jakieś liczbowe kryterium, a kryterium tym będzie `ad` (suma bezwzględnych wartości różnic pomiędzy gęstością empiryczną i dopasowaną). Aby zrozumieć to lepiej, użyjmy opcji `draw.diff` (wartość logiczna, domyślnie `FALSE`):

```
> x <- loss.fit.dist("normal",x12,draw.diff = T)
```



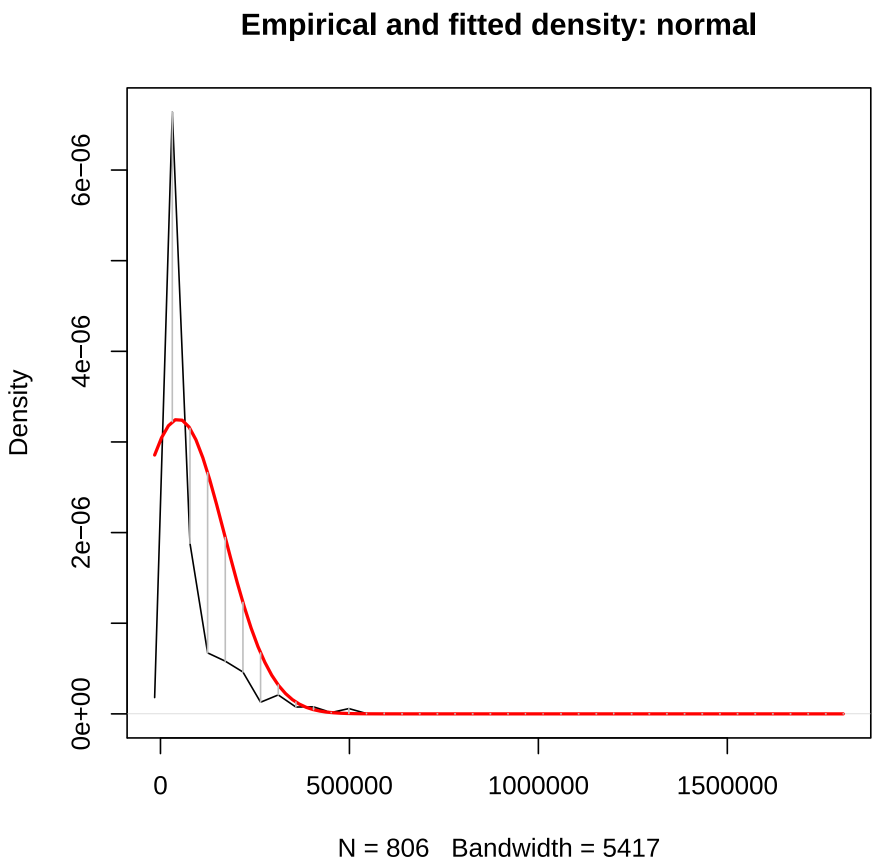
Prawdopodobnie będzie to bardziej czytelne przy użyciu `xlos.scale=T`:

```
> loss.fit.dist("normal", x12, draw.diff = T, xlog.scale = T)
```



Punkty y empirycznej i teoretycznej gęstości są połączone szarymi liniami; możemy zmienić ilość punktów użytych do narysowania tych gęstości i obliczyć dla nich te różnice, jednak nie wpływa to na estymację samych parametrów.

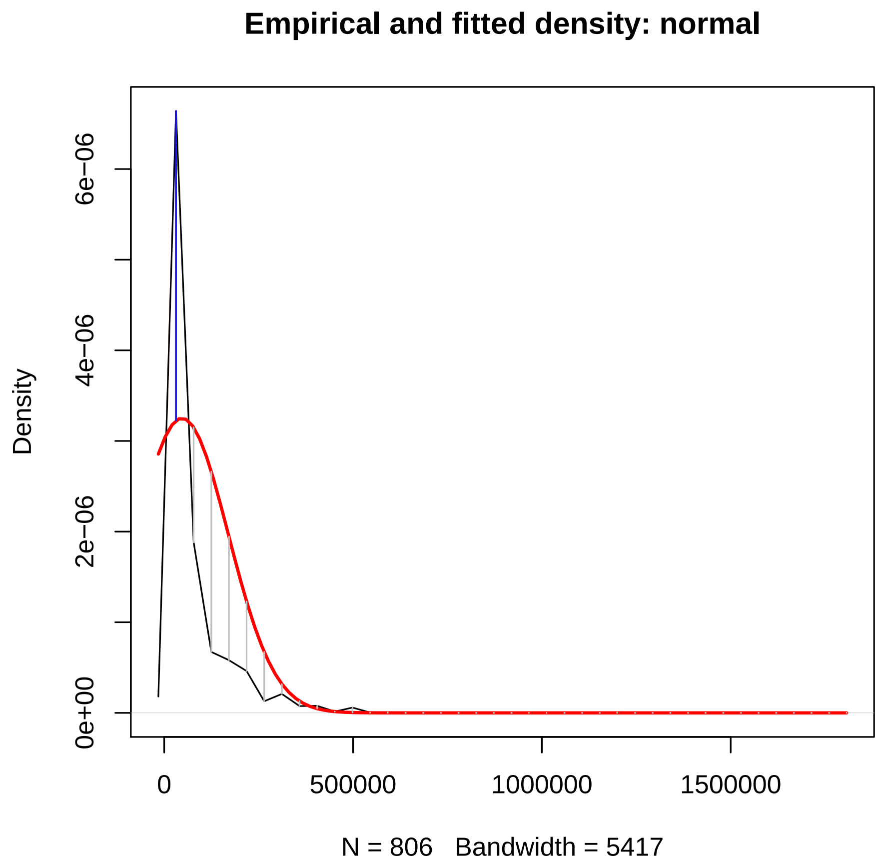
```
> loss.fit.dist("normal",x12,draw.diff = T,n = 40)

$mean
[1] 47079.52

$sd
[1] 122694.0

         ad
9.61874e-06
```



Oczywiście wzięcie zaledwie 40 punktów nie było dobrym pomysłem, ponieważ kryterium `ad` jest wtedy mniej wiarygodne; powinno być wiele punktów, gdyż jeśli mielibyśmy wiele niezachodzących na siebie linii (które na obrazku mają pewną szerokość), a między nimi brak (prawie) przerw, moglibyśmy pomnożyć `ad` przez tę „szerokość linii”, otrzymując coś przypominającego całkę.

```
> loss.fit.dist("normal",x12,draw.diff = T,n = 40,draw.max = T)
```

```
$mean
[1] 47079.52

$sd
[1] 122694.0

          ad
9.61874e-06
```



Teraz zobaczmy listę wszystkich `loss.fit.dist()` wartości

```
> summary(x)

           Length Class  Mode
loglik       1    -none- numeric
param        2    -none- list
sd           2    -none- numeric
q.t          0    -none- NULL
q.e          0    -none- NULL
ad           1    -none- numeric
teor.dens  507    -none- numeric
emp.dens   507    -none- numeric
maxdiff      1    -none- numeric
meandiff     1    -none- numeric
```

Gdzie: `loglik` (logarytm wiarogodności), `param` (estymowane parametry) and `sd` (estymowane błędy standardowe) są faktycznie wartościami `fitdistr()`; `q.t` i `q.e` to teoretyczne i empiryczne kwantyle liczone jedynie wówczas, gdy `qq = T` (domyślnie `FALSE`); `teor.dens` i `emp.dens` są wartościami teoretycznych i empirycznych wartości funkcji gęstości.
Dlaczego jednak mamy tylko 507 punktów w tych dwóch elementach, skoro `density()` domyślnie liczy `512` wartości? Możemy to sprawdzić:

```
> length(density(x12[,2])$y)
```

```
[1] 512
```

Jest to proste - liczenie bezwzględnych wartości różnic dla niedodatnich `x` nie wydaje się mieć wiele sensu, zatem:

```
> length(which(density(x12[,2])$x >0))
```

```
[1] 507
```

`fit.plot()` używany w `loss.fit.dist()` bierze tylko te dodatnie argumenty `x`.
Dwoma ostatnimi wartościami są `maxdiff` (maksymalna różnica bezwzględna) i `meandiff` (średnia różnica bezwzględna):

```
> x$maxdiff
```

```
[1] 2.946671e-05
```

```
> x$meandiff
```

```
[1] 4.663462e-07
```

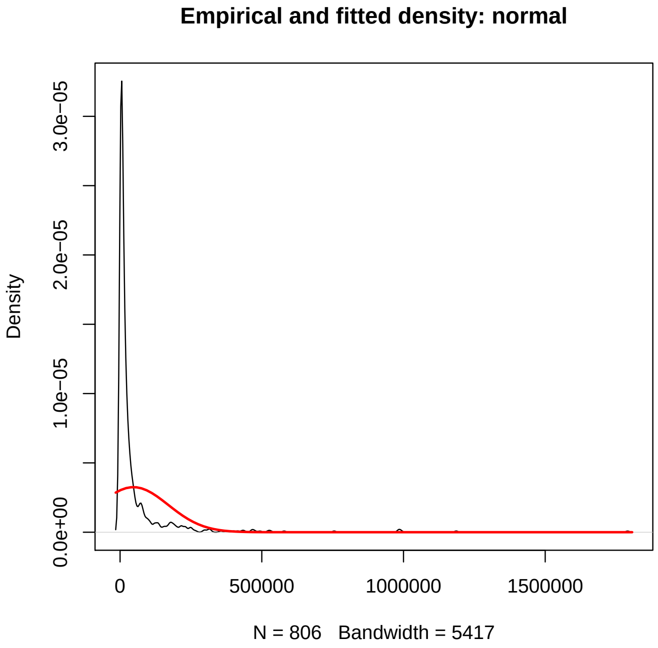
Teraz użyjmy opcji `qq`.

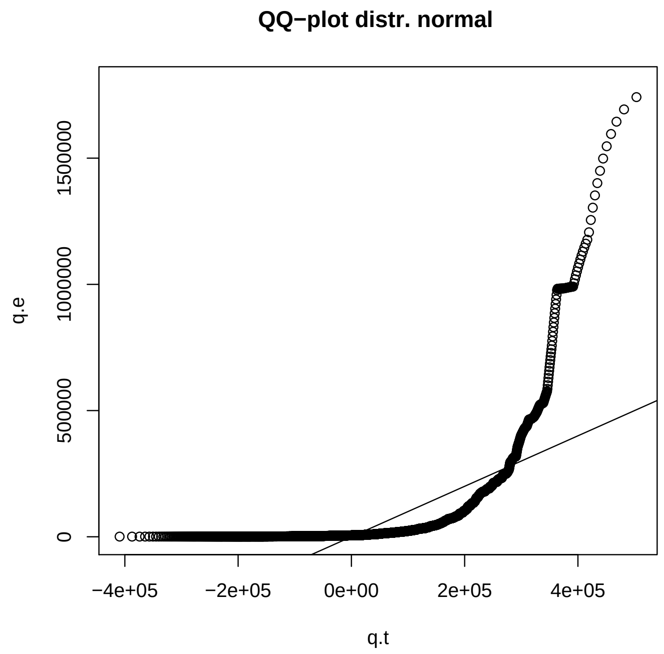
```
> z<- loss.fit.dist("normal",x12,qq=T)
> summary(z$q.e)
```

```
    Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
   794.2    3845.0   11180.0   46110.0   34600.0 1791000.0
```

```
> summary(z$q.t)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-927300  -35680   47080   47080  129800 1021000
```





Rysunek 5.7: Teoretyczne kwantyle dla dopasowanego rozkładu normalnego i empiryczne kwantyle dla danych x12

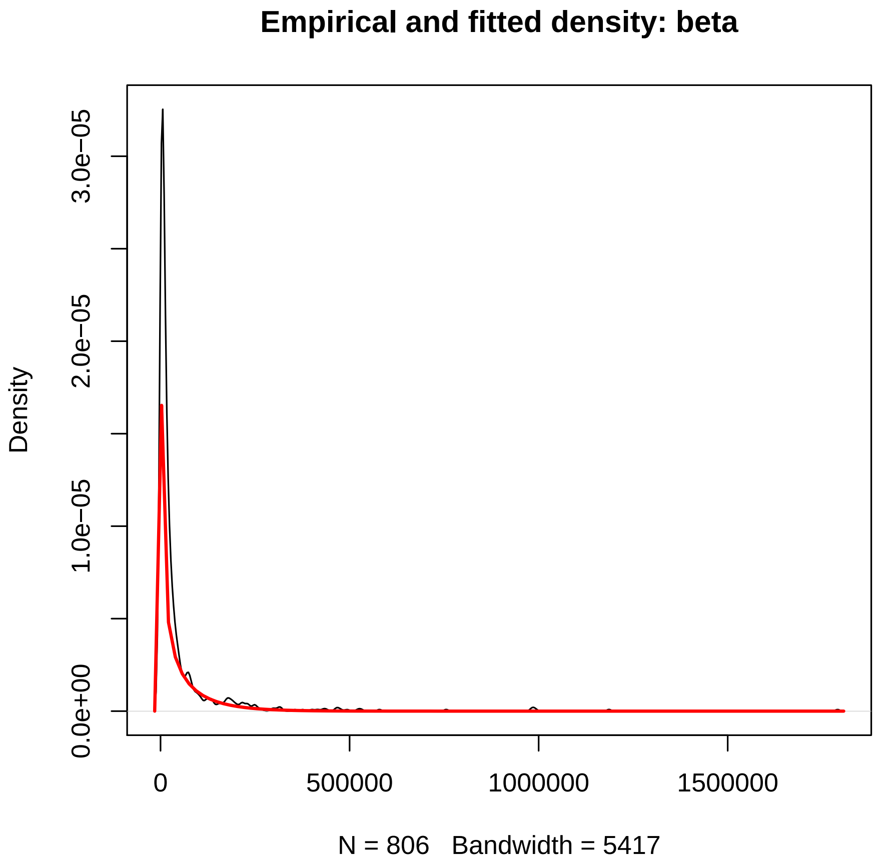
Oczywiście kwantyle nie układają się wzdłuż linii `y = x`, ale nie oczekujemy tego, wiedząc „z rysunku", że dopasowanie danych jest bardzo słabe. Mamy:

```
> head(cbind(z$q.e,z$q.t))

                    [,1]      [,2]
1.000000e-13% 794.2200 -927275.6
1.000100e-02% 807.3251 -409218.3
2.000200e-02% 820.4302 -387264.2
3.000300e-02% 833.5353 -373955.5
4.000400e-02% 846.6404 -364284.8
5.000500e-02% 859.7455 -356644.8
```

Jak widzimy, otrzymujemy pewne wartości ujemne w teoretycznych kwantylach, i to dość skrajne. To także potwierdza kiepskie dopasowanie rozkładu.
Zauważmy, że mamy opcję `length.out` (10000 domyślnie), dzięki której określamy długość sekwencji `p`. gdzie `p = seq(from, to, length.out, by)` jest wartością `probs` w `quantile`.

Na naszej liście rozpoznawalnych rozkładów niektóre są w pewien sposób specjalne. Mamy rozkład `"beta"`, ciągły rozkład prawdopodobieństwa zdefiniowany na przedziale (0,1). Oczywiście straty mogą zostać przeskalowane, ale pozostaje nadal znaczący problem: straty są ograniczone przez ich maksimum i tak samo ograniczone są ich symulacje. Zapewne nie byłoby dobrym pomysłem symulować straty za pomocą tego rozkładu, zwłaszcza nieoczekiwane.

```
> loss.fit.dist("beta",x12)

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 0.471598

$shape2
[1] 17.10989

           ad
0.0001251046
```



Mamy:

```
> max(x12[,2])
```

```
[1] 1790530
```

...więc nie może być straty większej niż ta. Choć `"beta"` wydaje się dobrze dopasowany, należy o tym pamiętać.

Ale `fitdistr()` (i w rezultacie `loss.fit.dist()`) może dopasowywać rozkłady niebędące na liście rozkładów rozpoznawalnych z nazwy. Spróbujemy to pokazać na `"dnorm"` (bez nazwy będzie on traktowany jak nieznany rozkład); będziemy potrzebować wartości startowych, więc:

```
> new.start = list(sd= sd(x12[,2]),mean = mean(x12[,2]))
```

```
> loss.fit.dist(dnorm,x12,start = new.start)
```

Daje to komunikat o błędzie:

```
Error in solve.default(res$hessian) :
Lapack routine dgesv: system is exactly singular
```

Wartości startowe były raczej dobre (metoda momentów):

```
> new.start
```

```
$sd
[1] 122770.1

$mean
[1] 47079.52
```

W rzeczy samej, porównajmy je z estymowanymi parametrami otrzymanymi dla `loss.fit.dist("normal",x12)`. Problem pochodzi z `fitdistr`:

```
> fitdistr(x12[,2],dnorm,start = new.start)
```

. . . i mamy to samo ostrzeżenie, co wcześniej. Jak możemy sprawdzić w `help(fitdistr)`, „direct optimization of the log-likelihood is performed using optim" - z `help(optim)` dowiadujemy się, że dla metody `"Nelder-Mead"` używanej w `fitdistr()`dokonuje się `500` iteracji (oraz „Defaults to 100 for the derivative-based methods"). Tylko dla metody `"SANN"` używamy `10000` ewaluacji (iteracji) - i nie ma innego kryterium stopu. Wypróbuj:

```
> fitdistr(x12[,2],dnorm,start = new.start,method = "SANN")
```

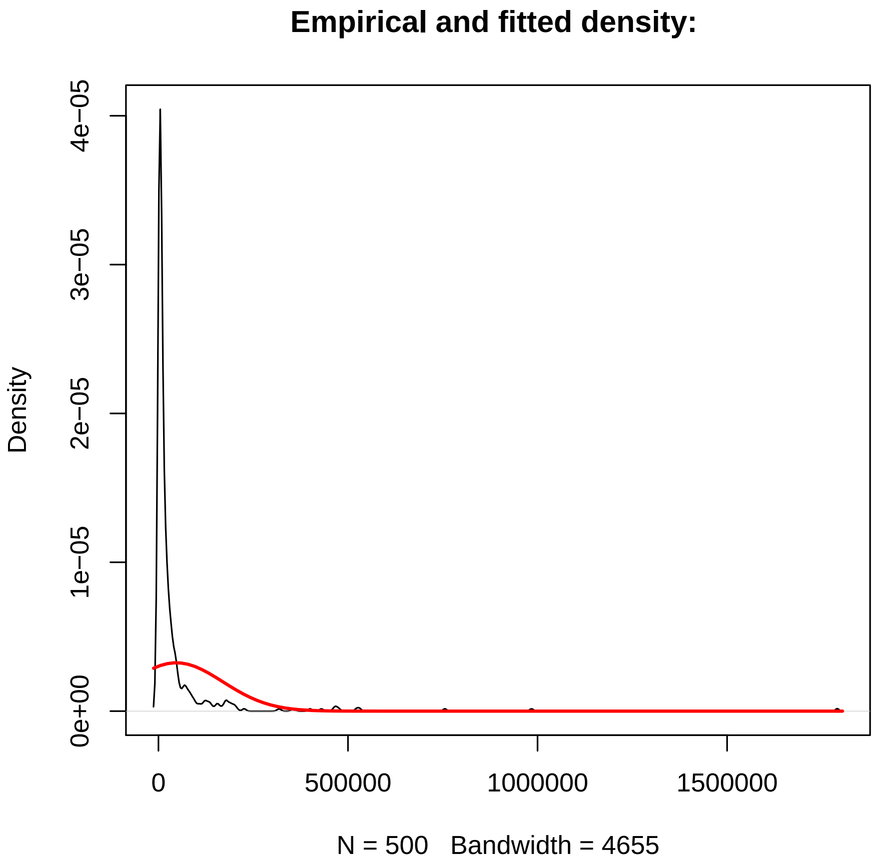
Metoda ta jest bazowana na symulacjach, stąd czasem daje wynik, a czasem nie. Spróbuj wywołać ją kilkakrotnie. Jeśli ma się szczęście, można otrzymać, prędzej czy później, pewne rezultaty (czasem dość dobre). Dla rozkładu `"normal"` traktowanego jako znany rozkład, wartości parametrów są po prostu obliczane jako:

```
> x<- x12[,2]
> n <- length(x)
>         sd0 <- sqrt((n - 1)/n) * sd(x)
>       mx <- mean(x)
>       estimate <- c(mx, sd0)
>       sds <- c(sd0/sqrt(n), sd0/sqrt(2 * n))
>       names(estimate) <- names(sds) <- c("mean", "sd")
> estimate

     mean        sd
 47079.52 122693.96
```

Można to łatwo sprawdzić wpisując `fitdistr` lub `fix(fitdistr)` aby zobaczyć kod funkcji.

Może dla dużych zbiorów danych i nierozpoznawalnych z nazwy rozkładów (w przypadku `loss.fit.dist()` będzie to także `"inverse gaussian"`, którego nie rozpoznaje `fitdistr()`) są potrzebne bardziej subtelne metody, gdyż `fitdistr()` nie wylicza żądanych wartości. Oczywiście zawsze można zmienić metodę, użyć metody momentów i/lub uciąć dane, by otrzymać pewne wyniki:

```
> loss.fit.dist(dnorm,x12[1:500,2],start = new.start)

$sd
[1] 122770.1

$mean
[1] 47079.52

          ad
0.0002690216
```

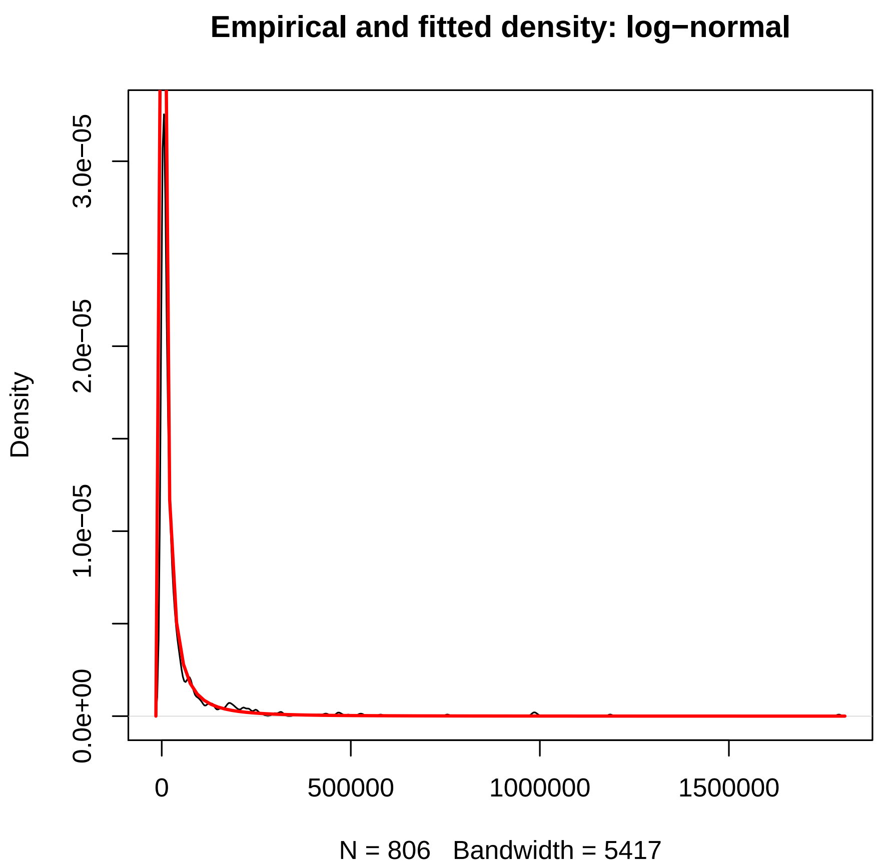


Rysunek 5.8: Empiryczna gęstość i dopasowana gęstość normalna dla pierwszych 500 wierszy z x12. Zauważmy, że argument distname nie został podany

Nie są to zbyt dobre wartości, ale nasze dane są inne - usunięto około 38 procent obserwacji.

Teraz nadszedł czas na zaprezentowanie dobrze dopasowanego rozkładu, a będzie to `"log-normal"`:

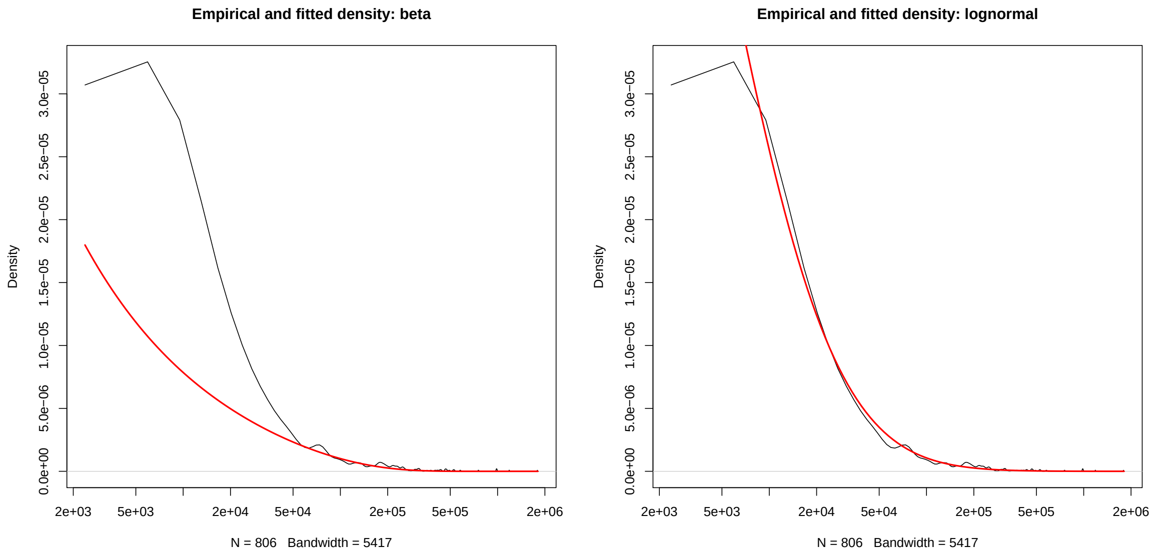
```
> loss.fit.dist("log-normal",x12)

$meanlog
[1] 9.455174

$sdlog
[1] 1.542905

          ad
6.158446e-05
```



Choć "beta" wydaje się raczej dobry, "log-normal" jest o wiele lepszy i zdaje się doskonały. Porównajmy te dwa rozkłady używając logarytmicznej skali x - będzie to TRUE dla opcji xlog.scale:

```
> par(mfrow = c(1,2))
>  loss.fit.dist("beta",x12,xlog.scale=T)

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 0.471598

$shape2
[1] 17.10989

          ad
0.0001251046

>  loss.fit.dist("lognormal",x12,xlog.scale=T)

$meanlog
[1] 9.455174

$sdlog
```

```
[1] 1.542905

          ad
6.158446e-05
```



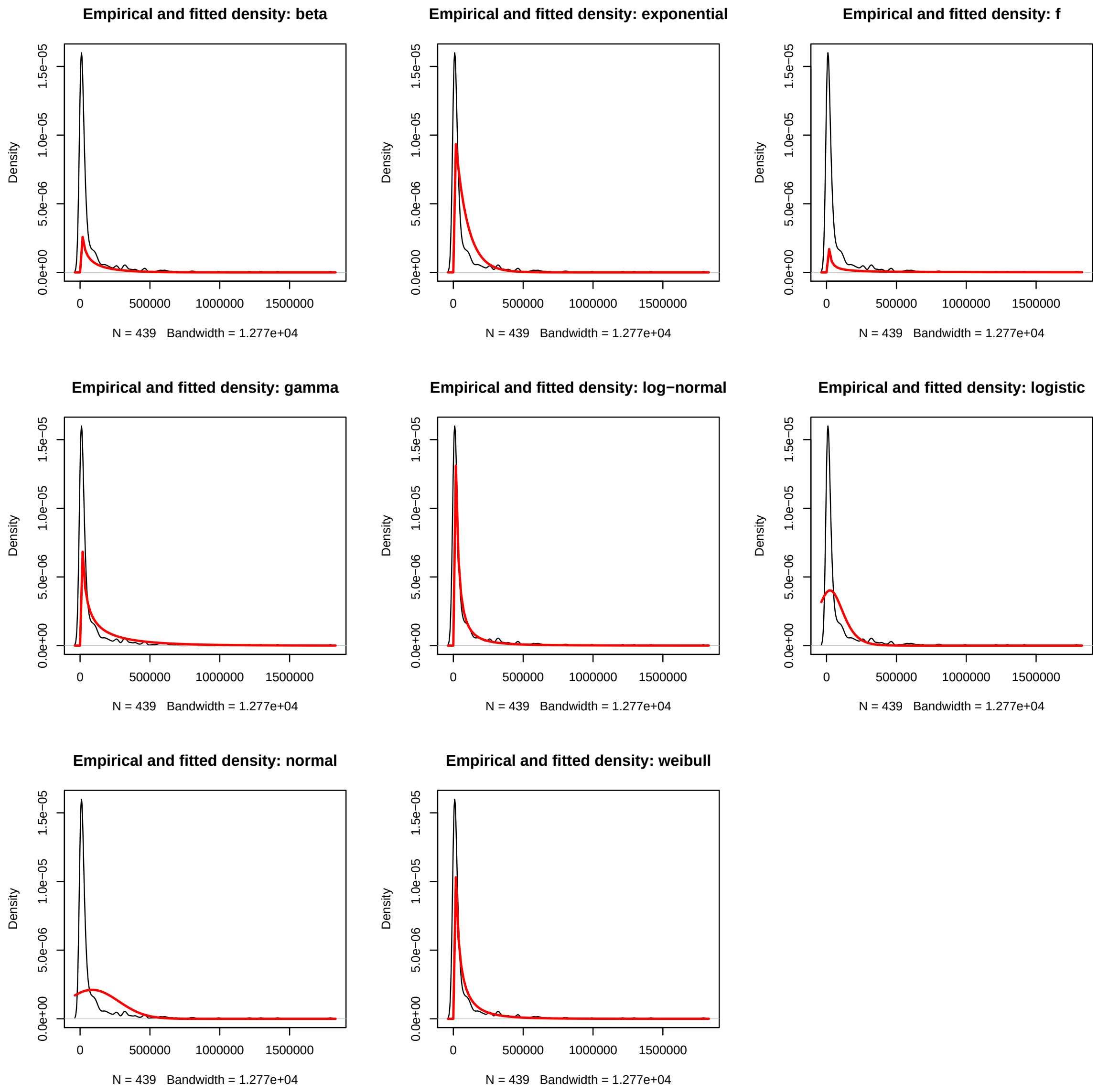
Rozkłady beta, wykładniczy, F, Gamma, log-normalny, logistyczny, normalny and Weibulla

Jak widać, "log-normal" jest niezaprzeczalnie lepszy od "beta". Ma również, rzecz jasna, mniejsze ad. Zobaczmy następujący prosty kod:

```
> dlist = c("beta", "exponential", "f", "gamma", "log-normal",
+                   "logistic", "normal", "weibull")
> par(mfrow = c(3,3))
> ad.days <- {}
> k <- 1
> for(i in dlist){
+ u<- loss.fit.dist(x=x12, densfun = i, period = "days")$ad
+ ad.days[[k]]<- u; k <- k+1
+ }

[1] "Argument scaled; x<- x/max(x)"

> names(ad.days)<- dlist
> ad.days<- sort(ad.days,decreasing = T); ad.days

      normal            f    logistic         beta  exponential        gamma
2.140493e-04 1.787725e-04 1.595774e-04 1.545514e-04 1.396397e-04 1.163744e-04
     weibull   log-normal
8.308964e-05 7.989927e-05

> names(which(ad.days ==min(ad.days)))
```

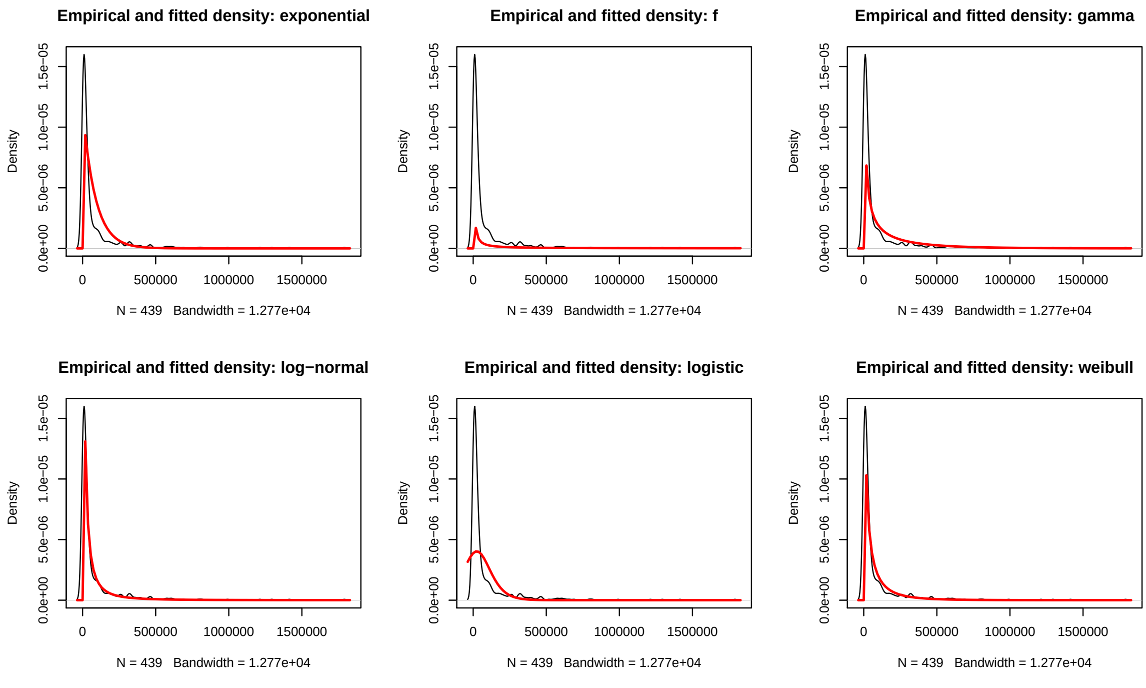
```
[1] "log-normal"
```

`dlist` to lista naszych rozkładów. Mamy tablicę 3 na 3 - `par(mfrow=c(3,3))` - i wyliczamy `ad` oraz rysujemy wykres dla każdego z tych rozkładów, otrzymując `ad.days` - listę naszych wartości `ad`, po czym wybieramy najmniejszą; `"log-normal"` jest najlepszy.



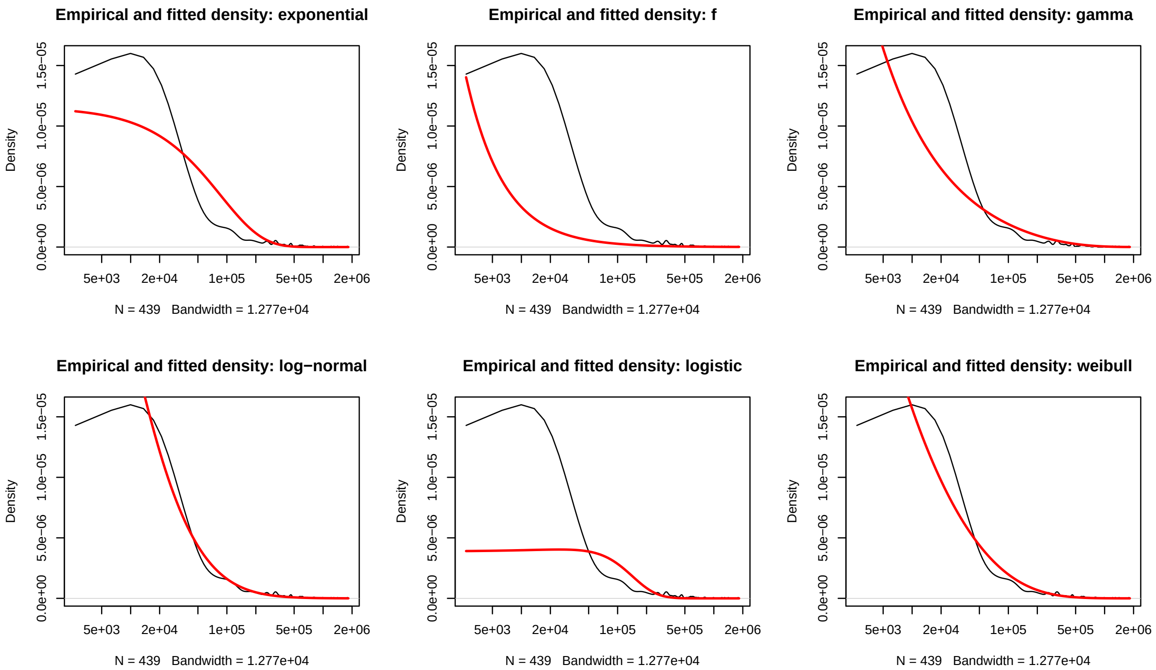
Możemy także od razu wykluczyć "beta" i "normal":

```
> dlist2 = c("exponential", "f", "gamma", "log-normal", "logistic", "weibull")
> par(mfrow = c(2,3))
> ad.days2 <- {}
> k <- 1
> for(i in dlist2){
+ u<- loss.fit.dist(x=x12, densfun = i, period = "days")$ad
+ ad.days2[[k]]<- u; k <- k+1
+ }
> names(ad.days2)<- dlist2
> ad.days2<- sort(ad.days2,decreasing = T); ad.days2

           f     logistic  exponential        gamma      weibull   log-normal
1.787725e-04 1.595774e-04 1.396397e-04 1.163744e-04 8.308964e-05 7.989927e-05

> names(which(ad.days2 ==min(ad.days2)))

[1] "log-normal"
```



I to samo, z użyciem `xlog.scale=T`:

```
> par(mfrow = c(2, 3))
> for (i in dlist2) {
+     u <- loss.fit.dist(x = x12, densfun = i, period = "days",
+         xlog.scale = T)$ad
+ }
```

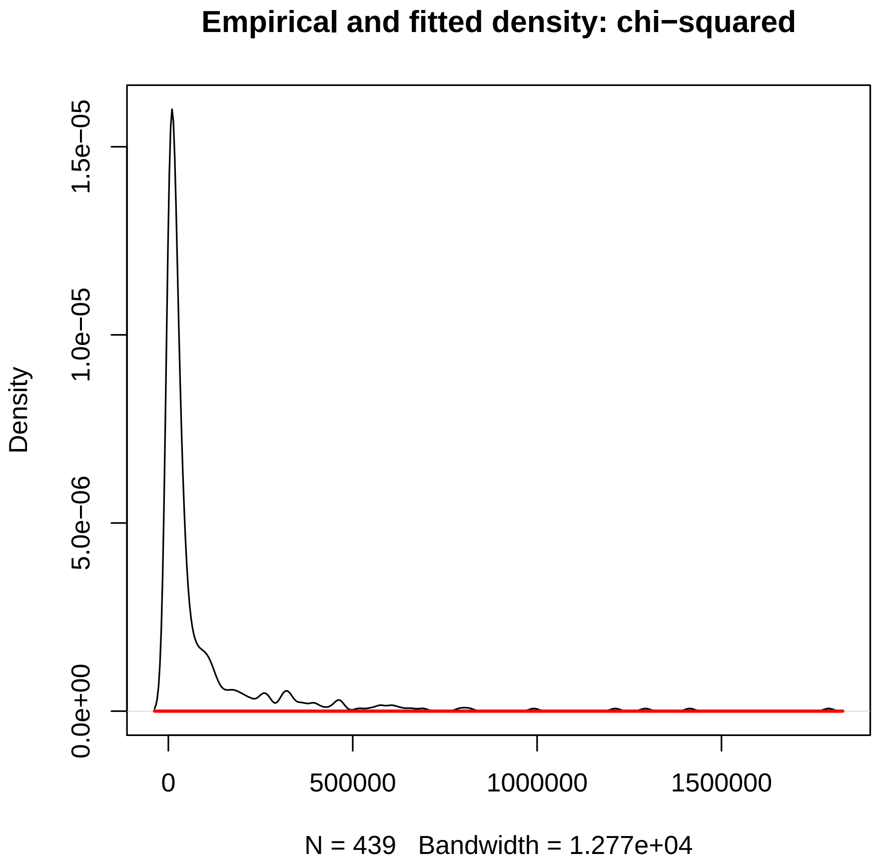


Zostają jednak jeszcze pewne rozkłady - `"cauchy"`, `"chi-squared"` i `"inverse gaussian"`. Powinny one zostać wywołane z dodatkowymi parametrami.

Rozkład Chi-kwadrat

Użyjemy `loss.fit.dist()` dla rozkładu `"chi-squared"`:

```
> k <- loss.fit.dist("chi-squared", x12, period = "days")
> k

$df
[1] 22521.04

          ad
0.0002301894
```



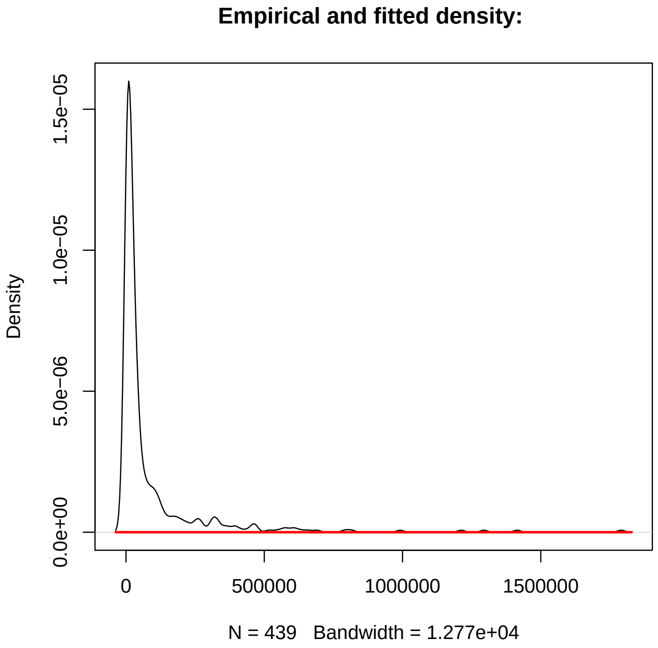
To nie wydaje się zbyt dobrym dopasowaniem. Może powinniśmy zmienić metodę?

```
> loss.fit.dist("chi-squared",x12,period="days",method = "BFGS")

$df
[1] 22504.64

          ad
0.0002301894
```

Bardzo podobny rysunek - niezałączony.

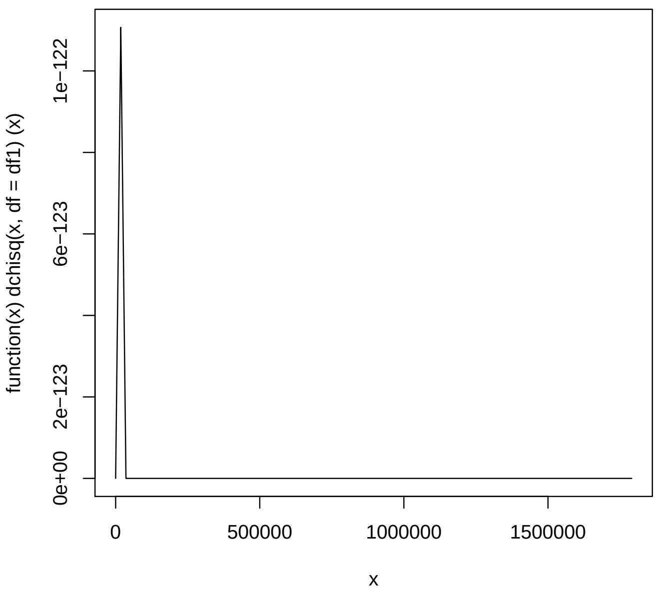
Dlaczego dla rozkładu `"chi-squared"` mamy narysowaną linię prostą? Dla `fit.plot()`, użytego w `loss.fit.dist()`, wywołanego z otrzymanymi wcześniej parametrami, mamy następujący rezultat:

```
> df1 <- as.numeric(k$param)
> z <- period.loss(x12, "days")
> fit.plot(z, dchisq, param = list(df = df1))

[1] 0.0002301894
```



Dlaczego wygląda to w ten sposób? Sprawdźmy:

```
> plot(function(x) dchisq(x,df = df1),xlim = c(0,max(z)))
```



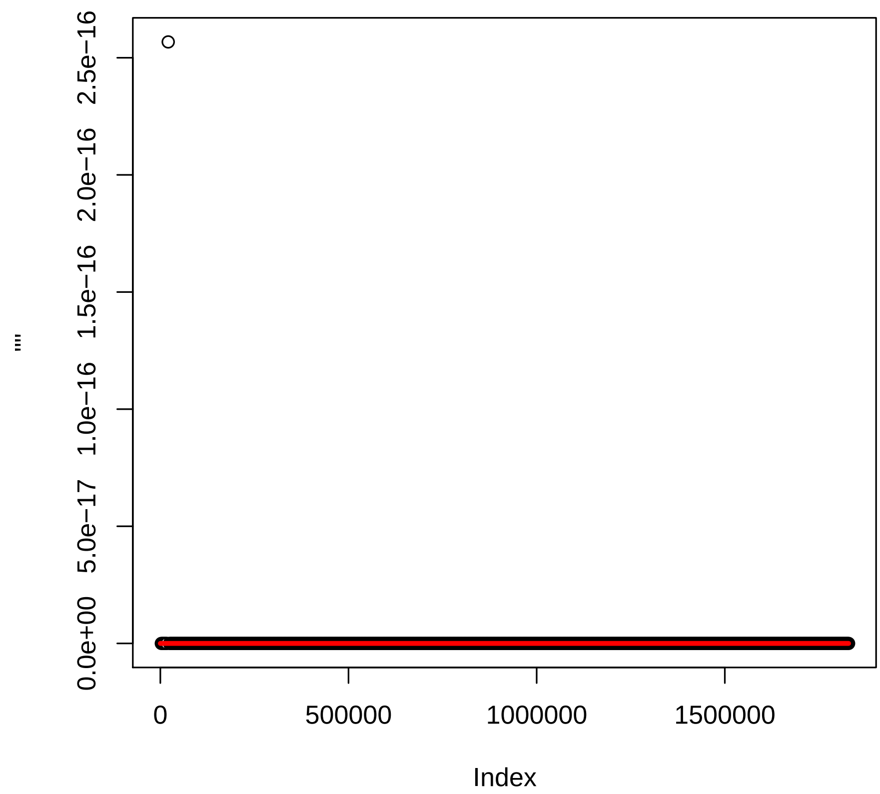
Oto gęstość rozkładu `"chi-squared"`.
Dlaczego jest tak rysowana? Sprawdźmy:

```
> nmbrs <- which(density(z)$x > 0)
> plus.nmbrs <- density(z)$x[nmbrs]
> plus <- which(dchisq(plus.nmbrs, df = df1) > 0)
```

```
> plus.values <- dchisq(plus.nmbrs[plus], df = df1)
> plus.values
```

```
[1] 3.219630e-161  2.567757e-16  1.037219e-22 2.640469e-138
```

```
> plot("", xlim = c(0, max(plus.nmbrs)), ylim = c(0, max(plus.values)))
> points(plus.nmbrs, dchisq(plus.nmbrs, df = df1))
> curve(dchisq(x, df = df1), add = T, col = "red", lwd = 3)
```

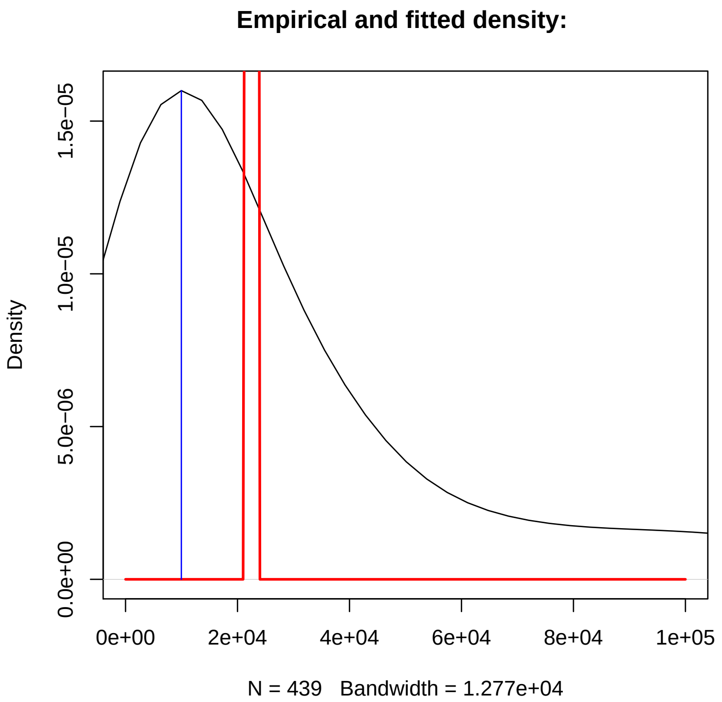


`nmbrs` są tymi z `density(z)$x`, które są ściśle dodatnie; sprawdzamy potem te z ściśle dodatnimi wartościami `density()` i rysujemy je (trzy z nich są tak małe w stosunku do czwartej, że na rysunku widzimy je jako zera). Potem używamy `curve()` i wydaje się, ze nie nie może ona połączyć wszystkich tych punktów - `max(plus.values)` jest wyłączony.

Możemy także sprawdzić, że pewna manipulacja `xlim` pozwala nam lepiej widzieć rysunek:

```
> fit.plot(z, dchisq, param = list(df = df1), draw.max = T, xlim = c(0,1e+05))
```

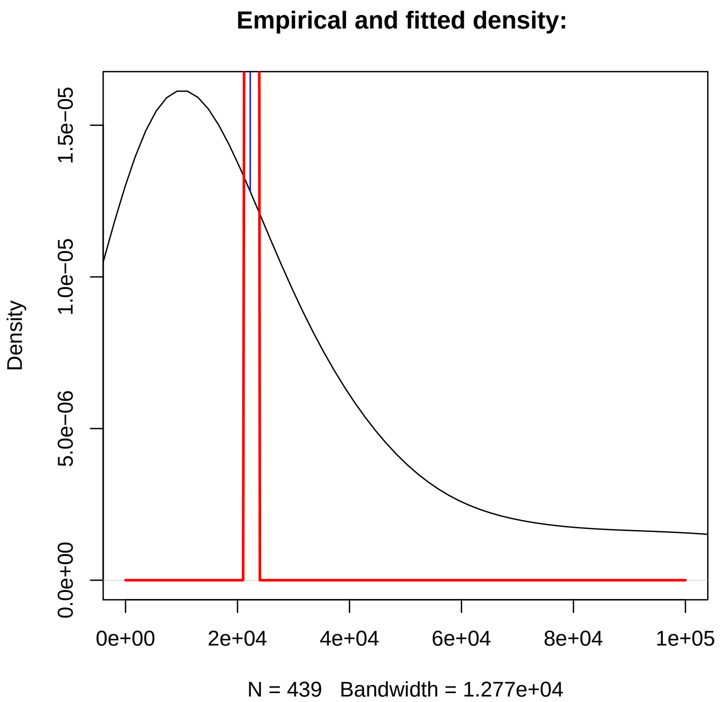
```
[1] 0.0002301894
```



Możemy także zmieniać `n` - ilość punktów w których oblicza się wartości:

```
> fit.plot(z, dchisq, param = list(df = df1), draw.max = T,
+     xlim = c(0,1e+05), n = 1000)
```

```
[1] 0.001365987
```



...jak jednak widzimy, ten rozkład nie pasuje dobrze do tych danych, zostawmy go więc.

Rozkład Cauchy

Chcielibyśmy dopasować także rozkład `"cauchy"` do `x12`, z `period = "days"`:

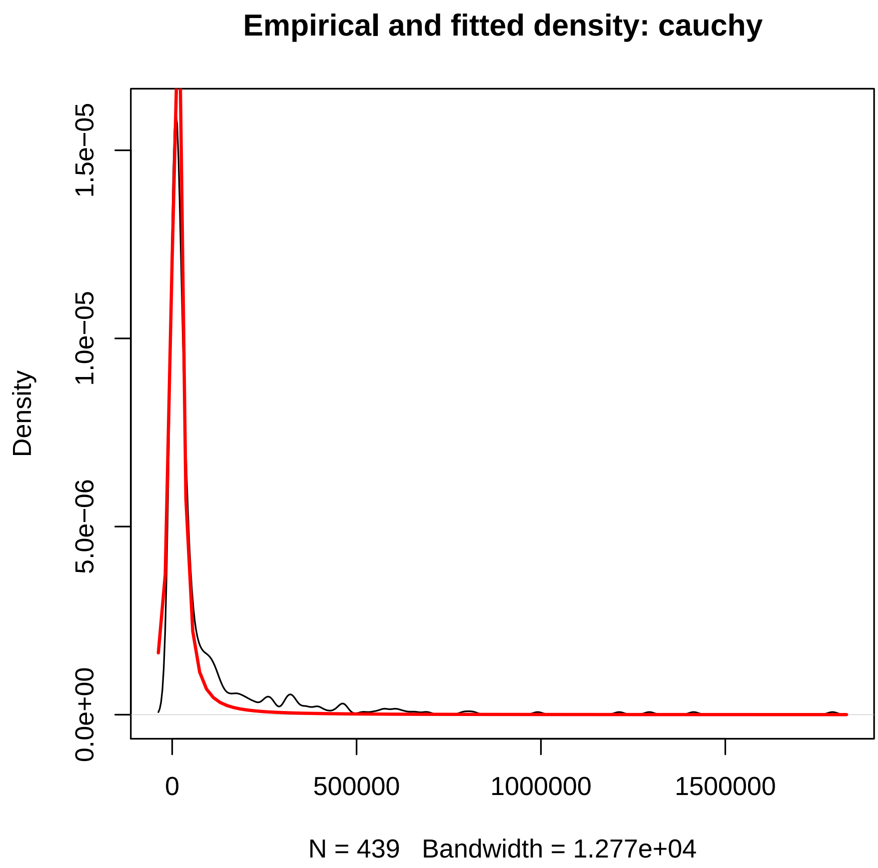
```
> loss.fit.dist("cauchy",x12,period = "days")
```

Daje to komunikat błędu:

```
Error in solve.default(res$hessian) :
Lapack routine dgesv: system is exactly singular
```

Możemy jednak zmienić metodę (`method`):

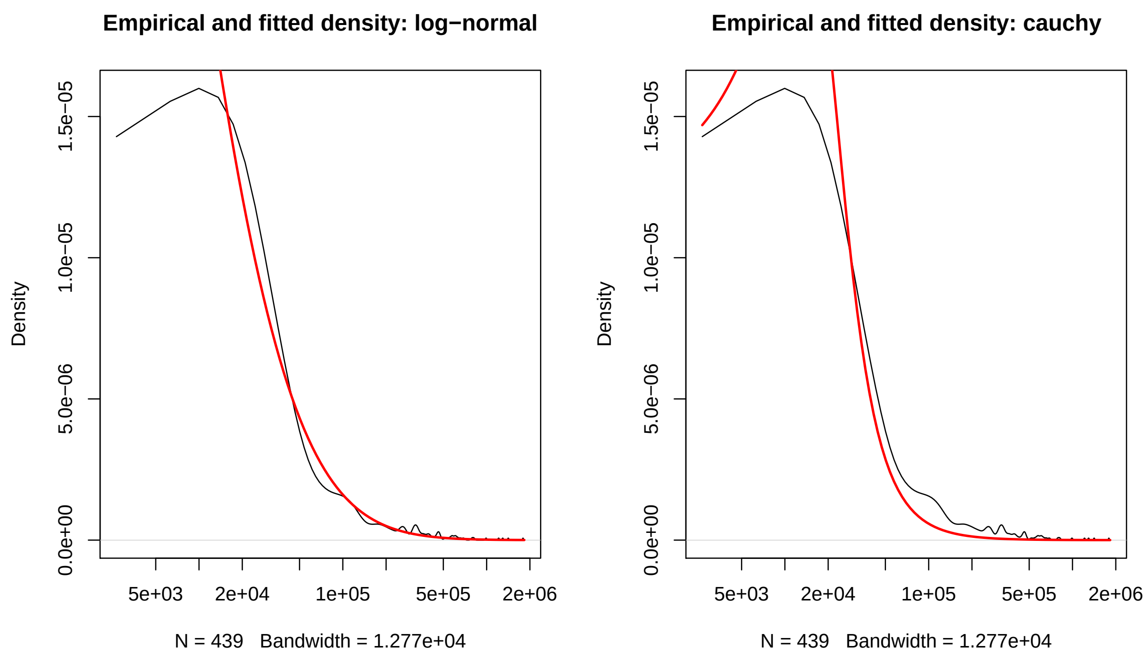
```
> loss.fit.dist("cauchy",x12,period = "days",method = "Nelder-Mead")

$location
[1] 12959.27

$scale
[1] 14188.55

          ad
8.679948e-05
```



Jest to dobre dopasowanie, ale `"log-normal"` nadal jest lepszy dla tych danych.

Możemy także porównać te dopasowania używając `xlog.scale=T`:

```
> par(mfrow = c(1, 2))
> loss.fit.dist("log-normal", x12, period = "days", xlog.scale = T)
> loss.fit.dist("cauchy", x12, period = "days", method = "Nelder-Mead",
+     xlog.scale = T)
```



Pozostaje tylko `"inverse gaussian"`:

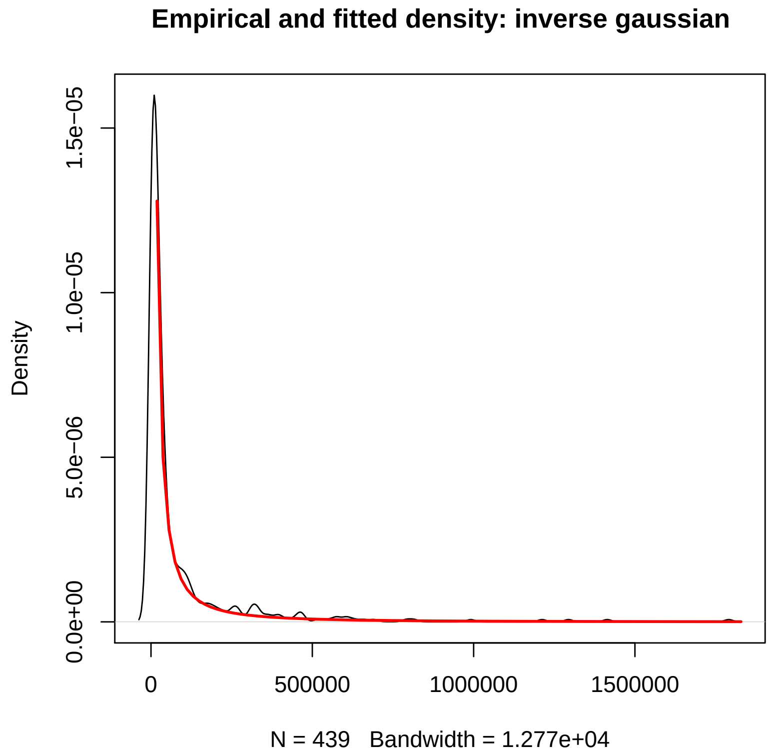
Rozkład odwrotny gaussowski

```
> loss.fit.dist("inverse gaussian",x12,period = "days")
```

Daje ten sam komunikat błędu jak dla rozkładu `"cauchy"`. Zmieniając `method` mamy:

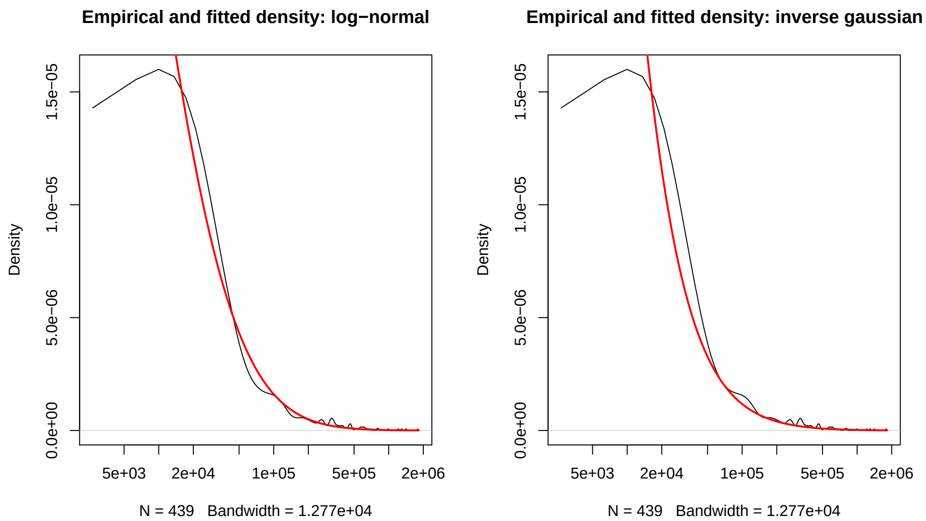
```
> loss.fit.dist("inverse gaussian",x12,period = "days",method = "Nelder-Mead")
```

```
$lambda
[1] 8653.568

$nu
[1] 86341.77

           ad
0.0001247034
```



Oczywiście `ad` dla `"inverse gaussian"` jest większe, niż dla `"log-normal"`. Możemy także porównać te dopasowania używając `xlog.scale=T`:

```
> par(mfrow = c(1, 2))
> loss.fit.dist("log-normal", x12, period = "days", xlog.scale = T)
> loss.fit.dist("inverse gaussian", x12, period = "days", method = "Nelder-Mead",
+     xlog.scale = T)
```

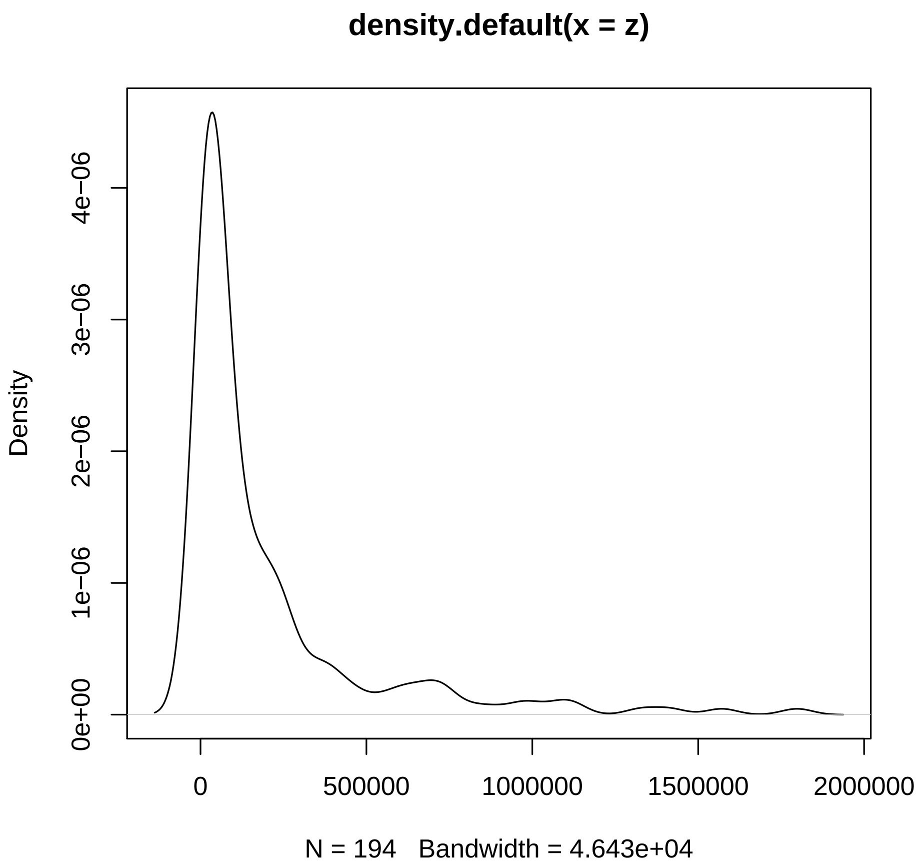


Także nie jest lepszy niż `"log-normal"`.

Inne okresy

Teraz zróbmy to samo dla okresów (`period`) `weeks`, `months` i `quarters`. Zobaczmy gęstość dla `weeks`:

```
> z<- period.loss(x12,"weeks")
> plot(density(z))
```

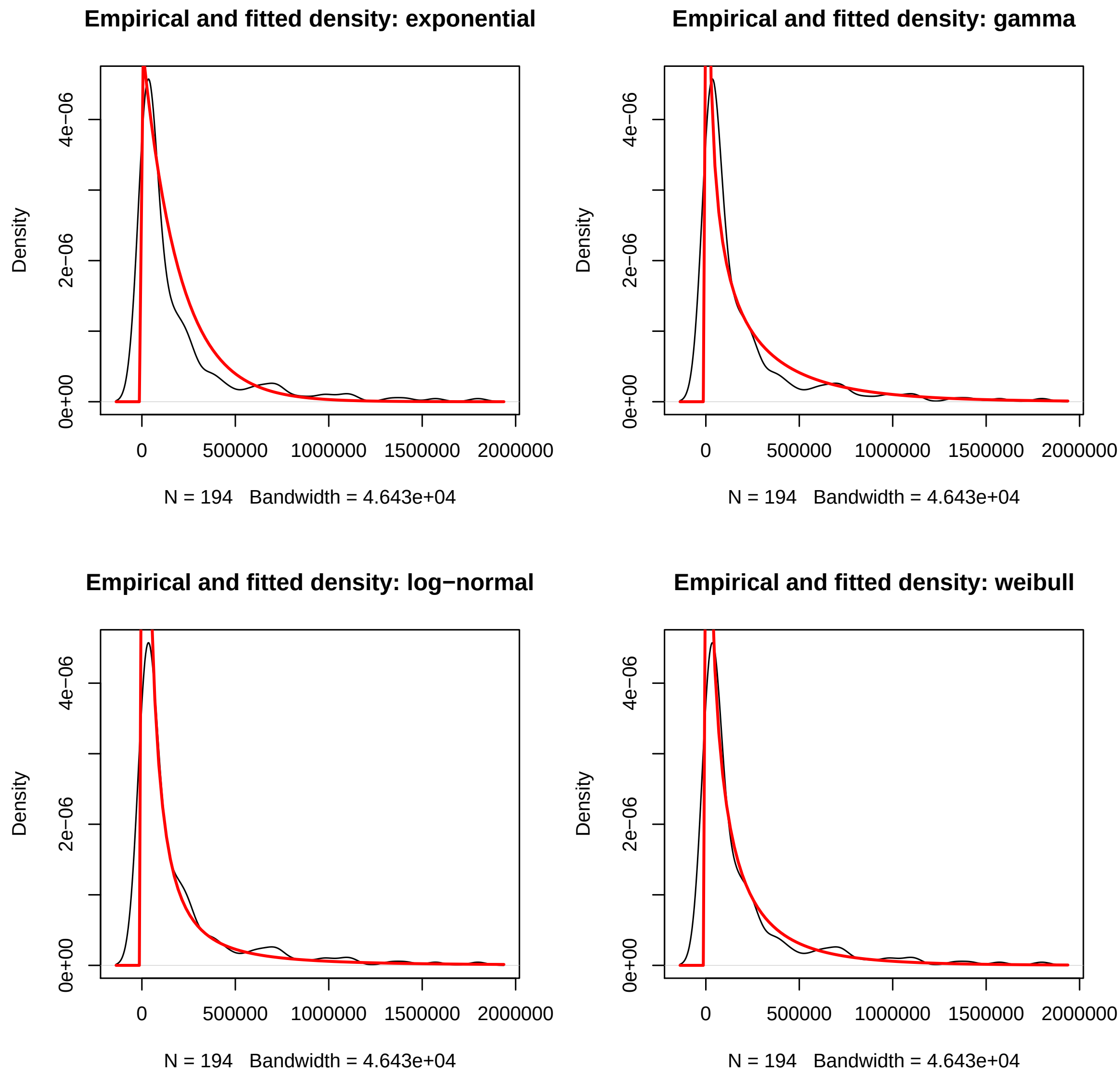


Mamy dobre dopasowanie dla `"weibull"`, `"gamma"`, `"log-normal"` i `"exponential"`, w tej kolejności. Różnice nie są wielkie:

```
> dlist3  =  c("exponential","gamma", "log-normal", "weibull")
> par(mfrow = c(2,2))
> ad.weeks <- {}
> k <- 1
> for(i in dlist3){
+ u<- loss.fit.dist(x=x12, densfun = i, period = "weeks")$ad
+ ad.weeks[[k]]<- u; k <- k+1
+ }
> names(ad.weeks)<- dlist3
> ad.weeks <- sort(ad.weeks ,decreasing = T); ad.weeks

 exponential   log-normal        gamma      weibull
7.216382e-05 6.797123e-05 6.345749e-05 5.706141e-05
```

```
> names(which(ad.weeks  == min(ad.weeks)))
```

```
[1] "weibull"
```



Wszystkie te rozkłady wydają się dobrze dopasowane; chcielibyśmy wybrać ten sam rozkład dla `days` i `weeks`. Powinniśmy być świadomi, że zmiana `n` może zmienić nasz wybór:

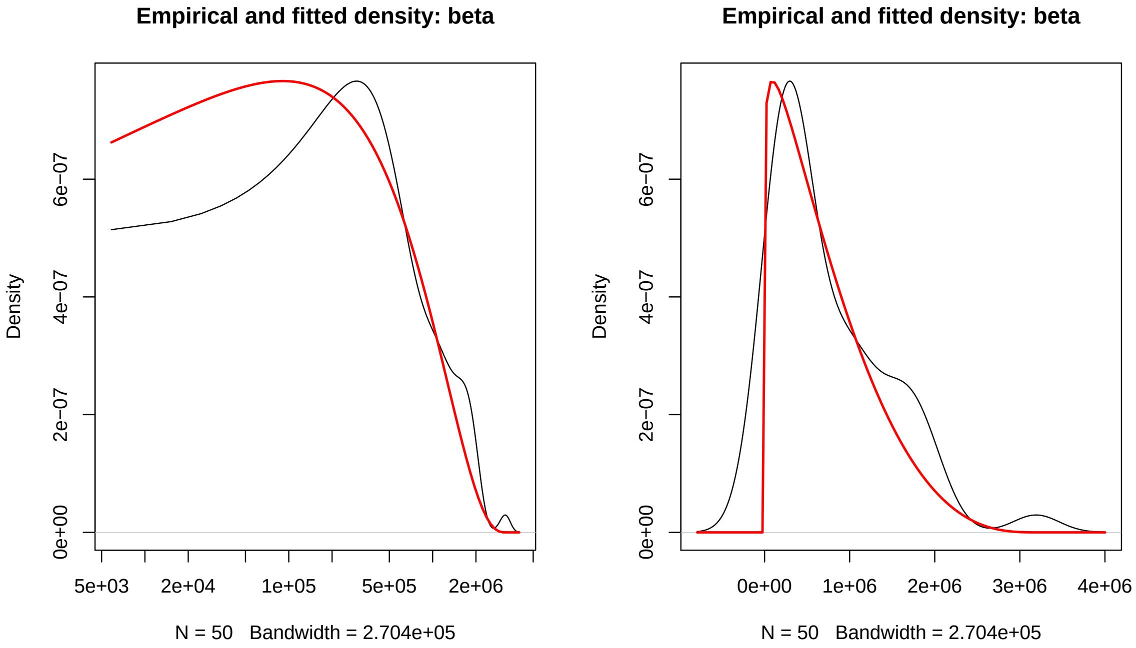
```
> dlist3  =  c("exponential","gamma", "log-normal", "weibull")
> par(mfrow = c(2,2))
> ad.weeks2 <- {}
> k <- 1
> for(i in dlist3){
+ u<- loss.fit.dist(x=x12, densfun = i, period = "weeks",n=10000)$ad
+ ad.weeks2[[k]]<- u; k <- k+1
+ }
> names(ad.weeks2)<- dlist3
> ad.weeks2 <- sort(ad.weeks2 ,decreasing = T); ad.weeks2
```

```
      gamma exponential     weibull  log-normal
0.001599348 0.001427725 0.001345542 0.001344964

> names(which(ad.weeks2==min(ad.weeks2)))

[1] "log-normal"
```

Jak widzimy, teraz `"log-normal"` jest najlepszym rozkładem. Oczywiście możemy zmienić `n` także dla `days`, ale wybierzemy po prostu rozkład `"log-normal"`.

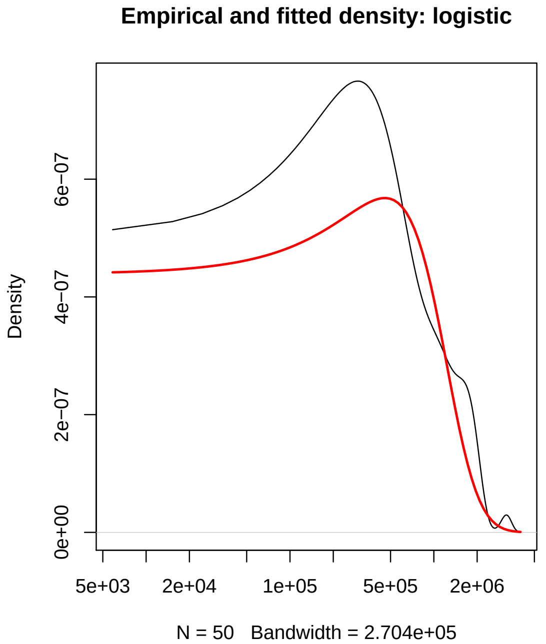
Dla miesięcy mamy rozkład `"beta"` lub `"logistic"`:

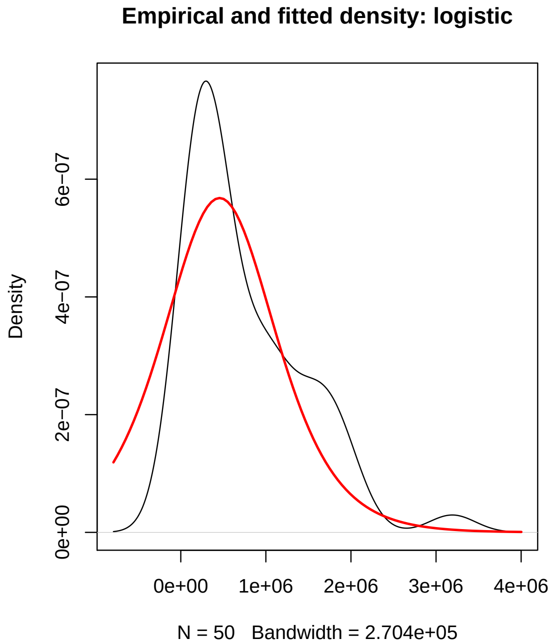
```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "beta", period = "months",xlog.scale=T)

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 1.080420

$shape2
[1] 3.753931

          ad
1.705283e-05

> loss.fit.dist(x=x12, densfun = "beta", period = "months")

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 1.080420

$shape2
[1] 3.753931

          ad
1.705283e-05
```



. . . oraz . . .

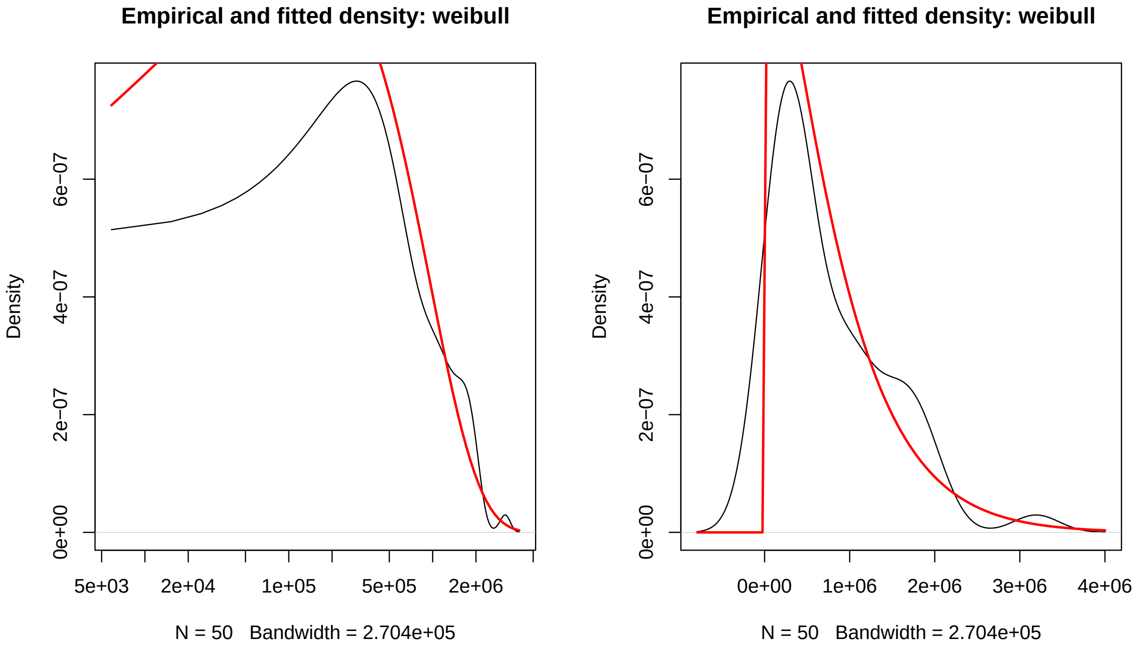
```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "logistic", period = "months",xlog.scale=T)

$location
[1] 456530.1

$scale
[1] 440127.1

          ad
2.350981e-05

> loss.fit.dist(x=x12, densfun = "logistic", period = "months")

$location
[1] 456530.1

$scale
[1] 440127.1

          ad
2.350981e-05
```





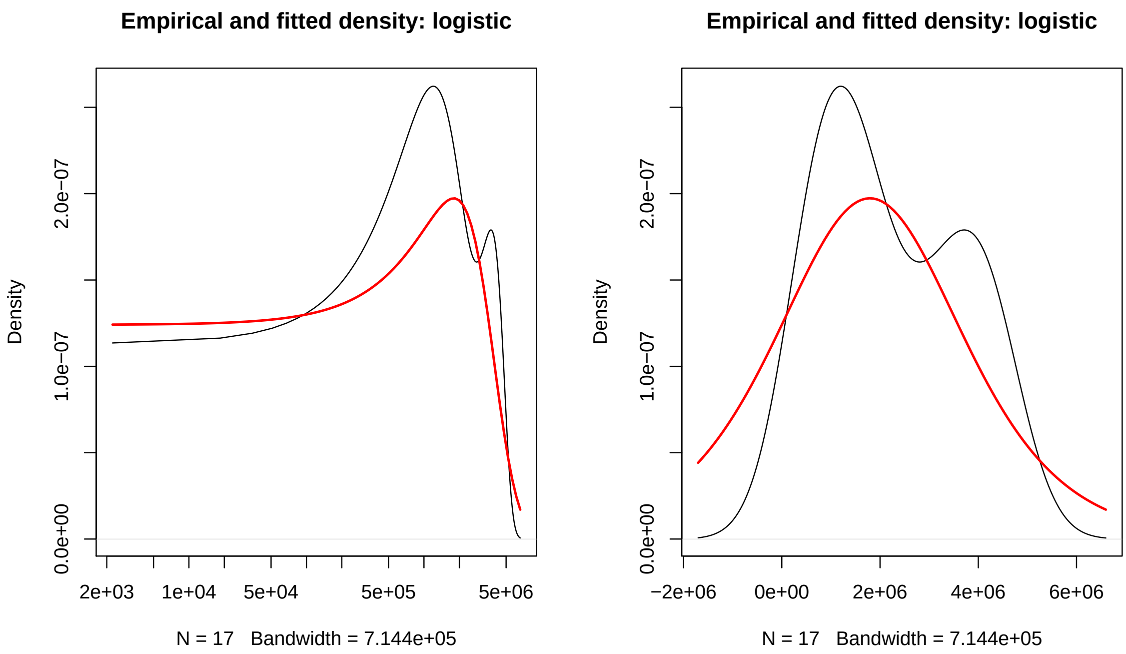
Także `"weibull"` zdaje się rozsądnym wyborem.

```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "weibull", period = "months",xlog.scale=T)

$shape
[1] 1.136458

$scale
[1] 797806.5

          ad
2.486692e-05

> loss.fit.dist(x=x12, densfun = "weibull", period = "months")

$shape
[1] 1.136458

$scale
[1] 797806.5

          ad
2.486692e-05
```



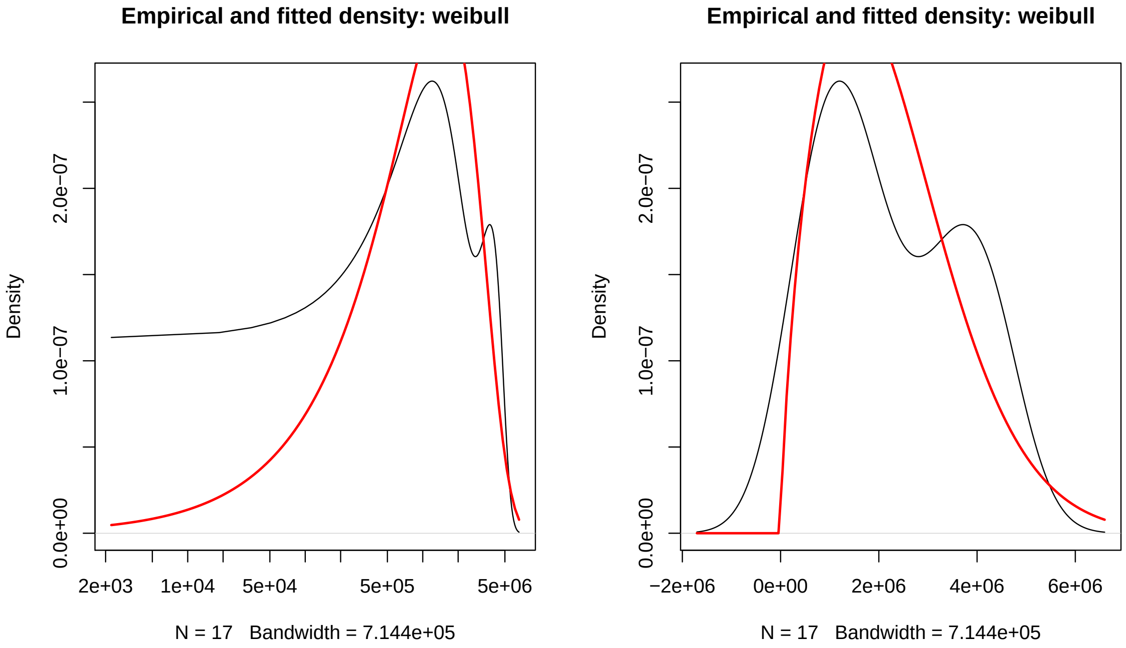
To samo dla kwartałów, tylko że teraz `"logistic"` jest najlepszy, a potem są `"weibull"` i `"beta"`.

```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "logistic", period = "quarters",xlog.scale=T)

$location
[1] 1793798

$scale
[1] 1266969

          ad
1.399808e-05

> loss.fit.dist(x=x12, densfun = "logistic", period = "quarters")

$location
[1] 1793798

$scale
[1] 1266969

          ad
1.399808e-05
```



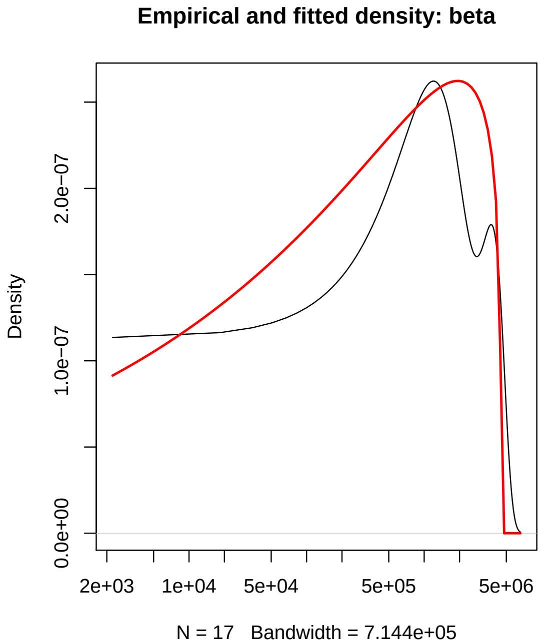
To samo dla rozkładu `"weibull"`:

```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "weibull", period = "quarters",xlog.scale=T)

$shape
[1] 1.705159

$scale
[1] 2528889

          ad
1.634660e-05

> loss.fit.dist(x=x12, densfun = "weibull", period = "quarters")

$shape
[1] 1.705159

$scale
[1] 2528889

          ad
1.634660e-05
```

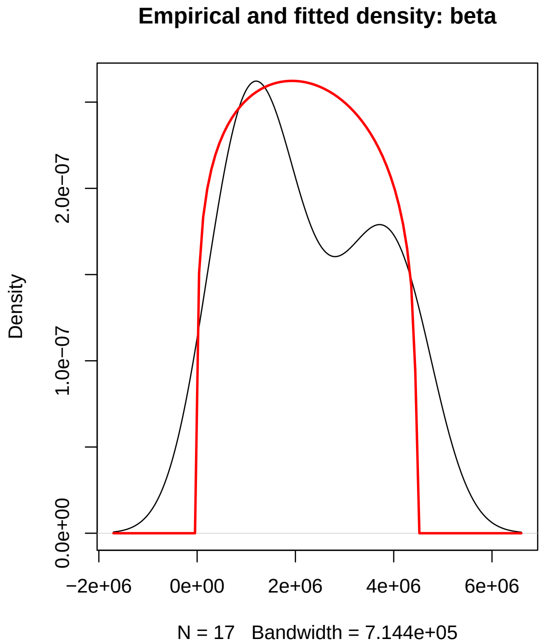


I to samo dla "beta":

```
> par(mfrow = c(1,2))
> loss.fit.dist(x=x12, densfun = "beta", period = "quarters",xlog.scale=T)

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 1.175090

$shape2
[1] 1.228670

          ad
1.741724e-05

> loss.fit.dist(x=x12, densfun = "beta", period = "quarters")

[1] "Argument scaled; x<- x/max(x)"
$shape1
[1] 1.175090

$shape2
[1] 1.228670

          ad
1.741724e-05
```



Empirical and fitted density: beta
Density
0.0e+00
1.0e−07
2.0e−07
2e+03
1e+04
5e+04
5e+05
5e+06
N = 17   Bandwidth = 7.144e+05



Empirical and fitted density: beta
Density
0.0e+00
1.0e−07
2.0e−07
−2e+06
0e+00
2e+06
4e+06
6e+06
N = 17   Bandwidth = 7.144e+05

Rozdział 6

Value at Risk - wartość zagrożona ryzykiem straty

Teraz wyliczymy ryzyko operacyjne dla naszych danych, używając funkcji `mc()`. Najpierw zróbmy to dla pewnych wybranych komórek.

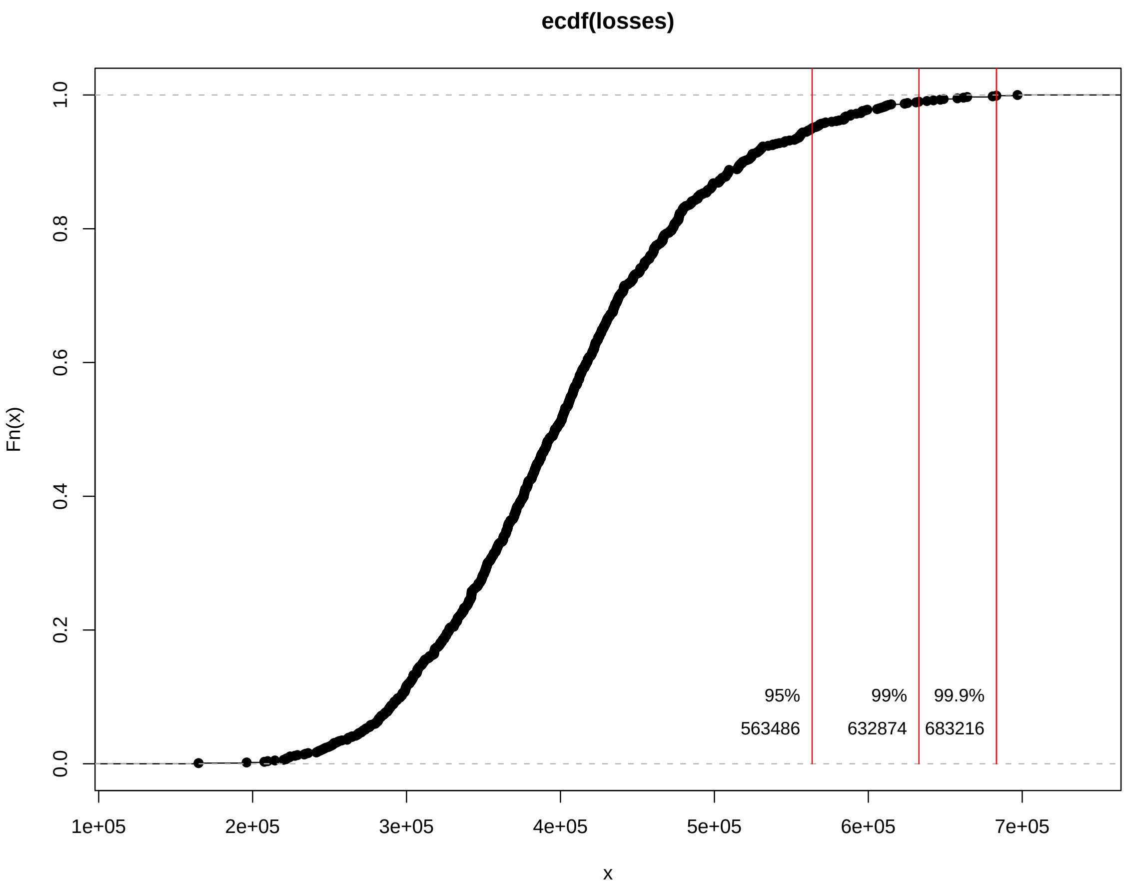
6.1. Value at Risk dla wybranych komórek

Zacznijmy od danych `x32`, na których pokażemy działanie funkcji `mc()`:

6.1.1. Komórka „Commercial Banking/Clients, Products & Business Practices"

```
> l1 = mc(x32)$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 124.9302  4 4.723739e-26

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 169.8749  4 1.112539e-35

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 3.08669  3 0.3784514
nbinomial
0.3784514

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 3.08669  3 0.3784514
$size
```

```
[1] 0.2834215

$prob
[1] 0.5978599

[1] "Argument scaled; x<- x/max(x)"
[1] "log-normal"
$meanlog
[1] 8.19784

$sdlog
[1] 0.9626437
```



Rysunek 6.1: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla x32

Oto kwantyle naszych strat:

```
> l1$q

     95%       99%     99.9%
563486.0 632874.1 683216.3
```

Oczywiście można zmienić kwantyle, domyślnie ustawione na `p = c(0.95, 0.99, 0.999)`.

Zostało zasymulowanych `1000` strat:

```
> length(l1$losses)

[1] 1000

> head(l1$losses)

[1] 376690.0 430624.6 384730.4 421828.0 358475.7 351422.3
```

Jeśli nie jest podany, `period` to `days`, a `iterate` to `years`.
Straty są symulowane w dla okresów `period` (tu: dni) a potem sumowane do rocznych strat, `iterate` to przedział czasowy (tu: lata), zaś `nmb` to liczba iteracji `iterate`.
W tym przykładzie mamy `nmb` (domyślnie `1000`) okresów `iterate`, co oznacza `365*nmb ==365000` strat dziennych do zasymulowania. Straty agregowane są po dniach, a potem po latach. Otrzymujemy `nmb` rocznych strat.

Teraz spróbujmy dla jedynie `10 nmb`, `iterate = "quarters"` i `period ="weeks"`. Oznacza to, że mamy `nmb=10` kwartałów (`iterate = "quarters"`), `13` tygodni na kwartał daje `10*13=130` tygodni, co, biorąc pod uwagę, że rok ma `52` tygodnie, jest równe `130/52=2.5` roku; symulujemy zawsze straty roczne. To nie daje pełnych lat, więc otrzymujemy ostrzeżenie:

```
> l2 = mc(x32, period = "weeks", iterate = "quarters", nmb = 10)$table

[1] "note that these are not full years"

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 28.42763  6 7.804346e-05

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 65.80124  6 2.959408e-12

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 4.644747  5 0.4607534
nbinomial
0.4607534

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 4.644747  5 0.4607534
$size
```

```
[1] 2.092517

$prob
[1] 0.6110669

[1] "Argument scaled; x<- x/max(x)"
[1] "exponential"
$rate
[1] 0.0001061786
```

Zobaczmy wartości funkcji `mc()`: symulowane straty, obliczone kwantyle i wartość `ad` dla dopasowanego rozkładu:

```
> l2

$losses
[1] 762077.6 585109.8 729420.7

$q
     95%      99%    99.9%
758811.9 761424.4 762012.2

$ad
         ad
0.001141128
```

. . . ale mamy 3 symulowane straty roczne. Jak to się dzieje? Zobaczmy:

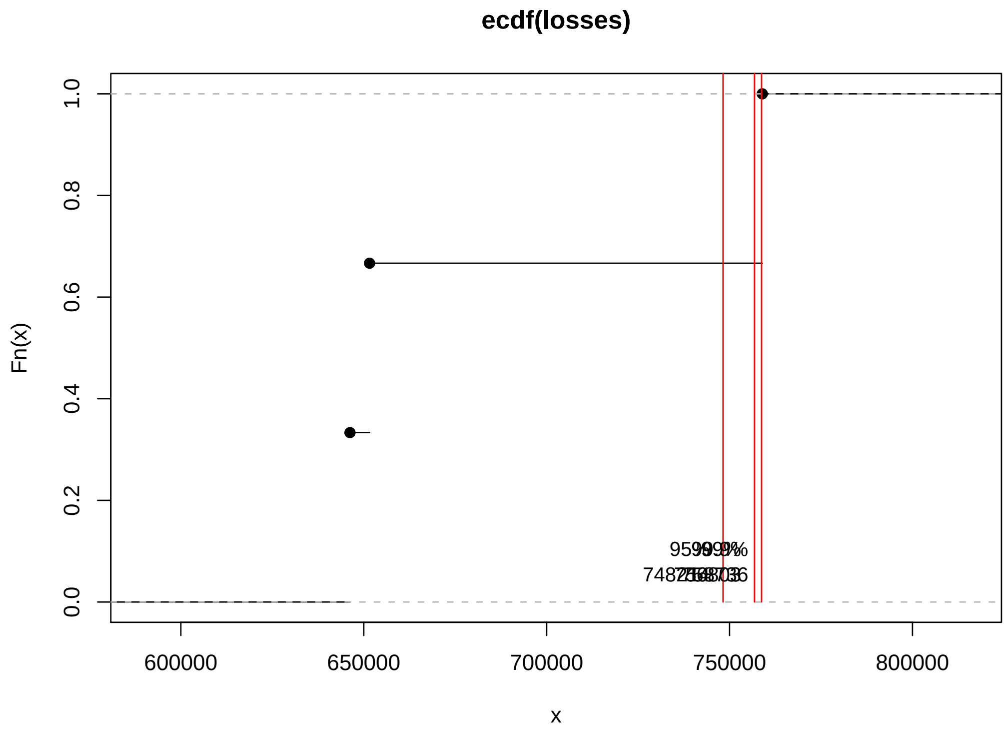
```
> matrix(c(1,2,3,4,4,4),3)

     [,1] [,2]
[1,]    1    4
[2,]    2    4
[3,]    3    4
```

To robi z `c(1,2,3,4,4,4)` macierz o 3 rzędach. A teraz:

```
> matrix(c(1,2,3,4,4,4,5),3)

     [,1] [,2] [,3]
[1,]    1    4    5
[2,]    2    4    1
[3,]    3    4    2
```

Otrzymujemy ostrzeżenie:
`In matrix(c(1, 2, 3, 4, 4, 4, 5), 3) :`
`data length [7] is not a sub-multiple or multiple of the number of rows [3]`
Siedmiu elementów nie da się rozpisać na trzy jednakowej długości wiersze, lecz macierz jest tworzona przez dodanie tylu elementów wektora, aby dopełnić ostatnią kolumnę (w tym wypadku są to dwa pierwsze elementy wektora).
Pokazuje to, że nasze dane mogą nie być całkowicie symulowane, jeśli używamy nieodpowiednich `nmb`, `iterate` lub `period`.

Obejrzyjmy jeszcze obrazek:



Rysunek 6.2: Empiryczna dystrybuanta dla trzech symulowanych rocznych strat dla x32

Oczywiście liczenie kwantyli z trzech obserwacji całkowicie mija się z celem, i prawdopodobnie najlepiej widać to na rysunku 6.2. Przykład ten zaprezentowano celem ostrzeżenia, natomiast do obliczania `VaR` stosowana jest symulacja dla `1000` lat, która wydaje się dawać dość przyzwoite wyniki, choć oczywiście ktoś dysponujący szybkim procesorem może na wszelki wypadek zwiększyć tę liczbę.

6.1.2. Komórka „Agency Services/Clients, Products & Business Practices”

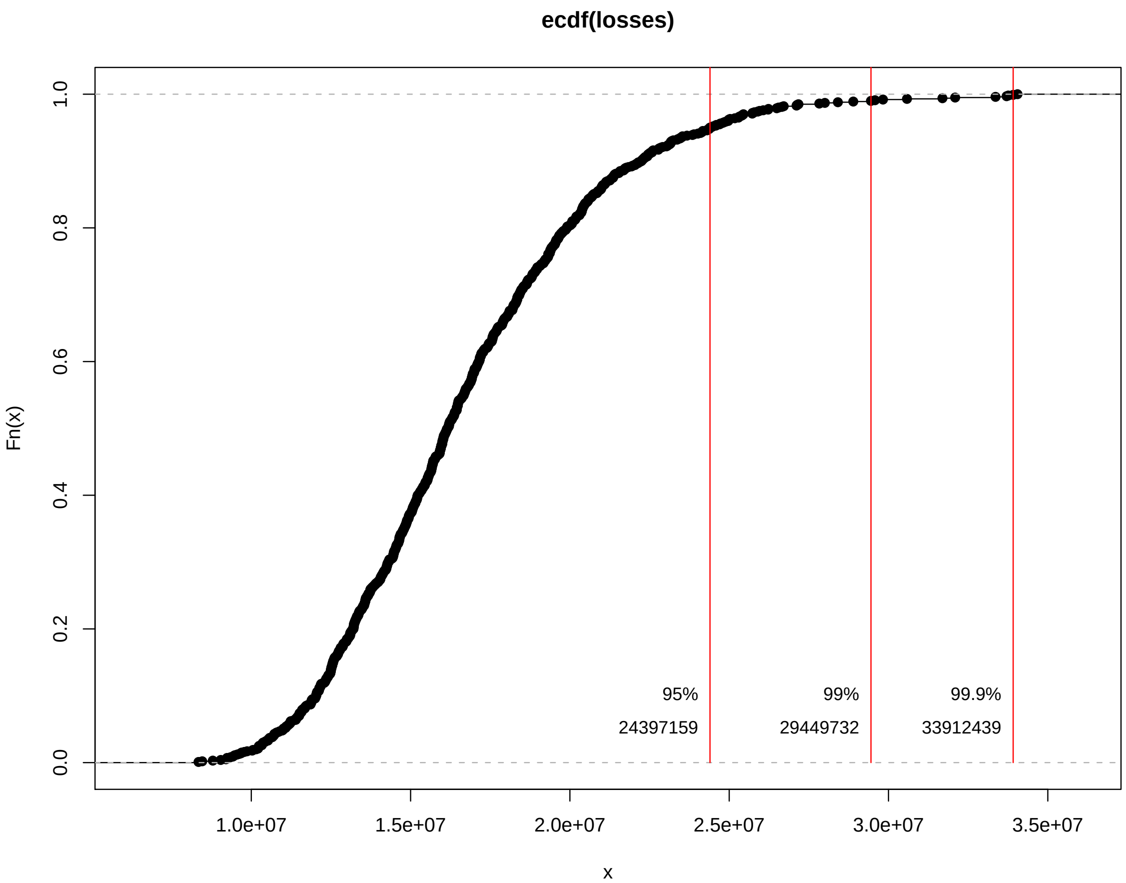
Teraz weźmy nasz `x12`. Znamy najlepiej do niego pasujące rozkłady częstości i dotkliwości, więc podamy je jako argumenty funkcji:

```
> l3 = mc(x12, rfun = "log-normal", type = "nbinomial")$table

         Goodness-of-fit test for nbinomial distribution

                      X^2 df      P(> X^2)
Likelihood Ratio 23.01709  5 0.0003350356
$size
[1] 0.4483843

$prob
[1] 0.4544358
```

```
$meanlog
[1] 10.02143

$sdlog
[1] 1.636008

> length(l3$losses)

[1] 1000
```

Zobaczmy kwantyle:

```
> l3$q

     95%      99%    99.9%
24397159 29449732 33912439
```



Rysunek 6.3: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla x32

6.2. Wszystkie linie biznesowe

Teraz dla wszystkich linii biznesowych. Najpierw przypiszemy straty do linii biznesowych:

```
> b.loss <- {}
> for(b in 1: length(loss.data.object$blines)){
+         b.loss[[b]]<- read.loss(b=b,r=1,loss.data.object)
+ for(r in 2: length(loss.data.object$rcateg)){
+         b.loss[[b]]<- rbind(b.loss[[b]],read.loss(b=b,r=r,loss.data.object))
+ }
+ }
```

`b[[i]]` to wszystkie straty przypisane do `loss.data.object$blines[i]`. To są całkowite liczby strat w liniach biznesowych - możemy porównać nasz wynik z `loss.matrix.image(data = loss.data.object)`:

```
> for(i in 1: length(loss.data.object$blines)){
+ print(paste(loss.data.object$blines[i],dim(b.loss[[i]])[1]))
+ }

[1] "Agency Services 988"
[1] "Asset Management 312"
[1] "Commercial Banking 464"
[1] "Corporate Finance 198"
[1] "Payment & Settlement 716"
[1] "Retail Banking 35"
[1] "Retail Brokerage 704"
[1] "Trading & Sales 192"
```

6.2.1. „Agency Services”

Dla `"Agency Services"` możemy sprawdzić wszystkie pozycje z `flist`, oprócz `"inverse gaussian"`, używając następującej komendy:

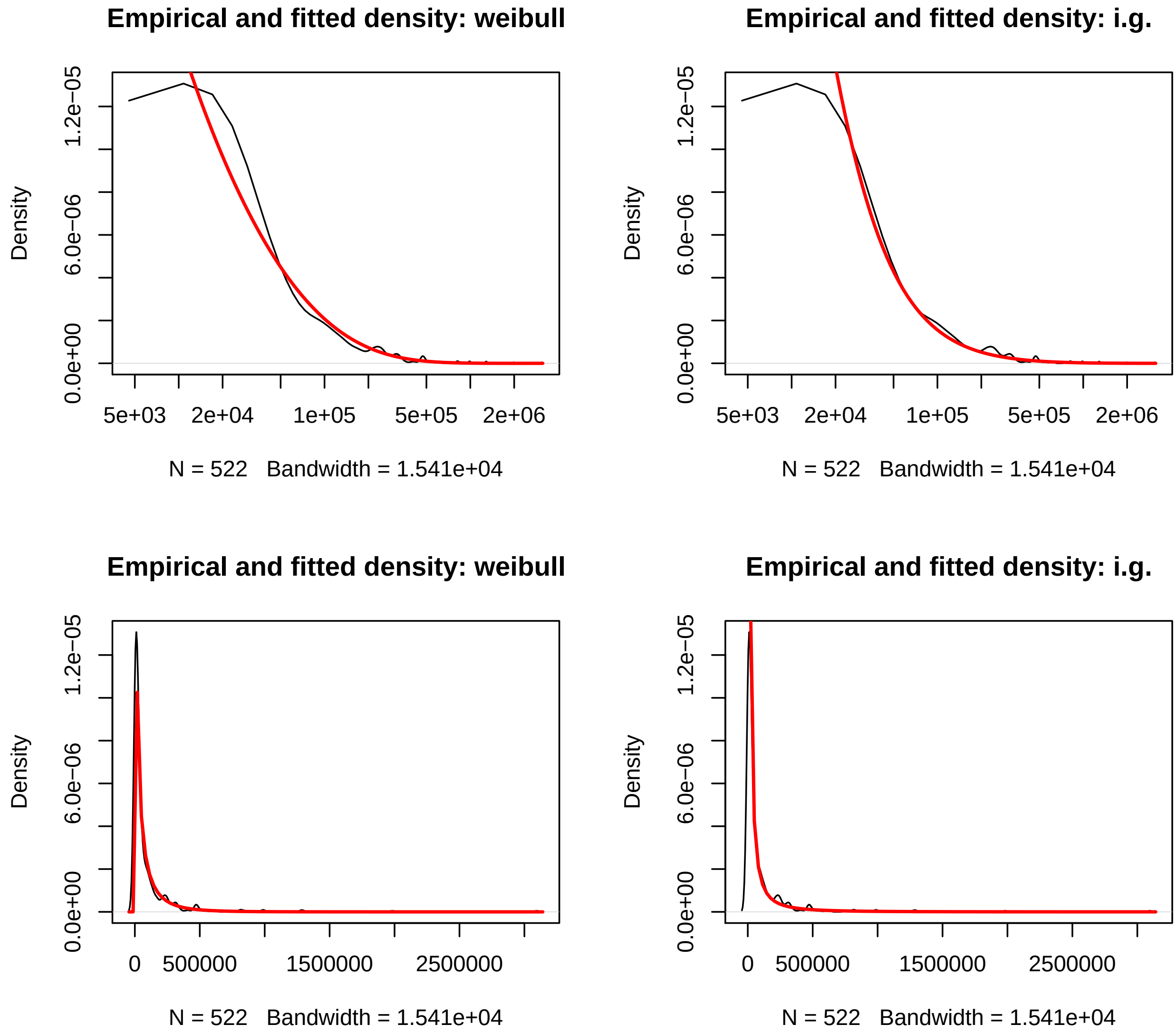
```
>  mc(b.loss[[1]], flist =  c("beta", "cauchy", "chi-squared", "exponential", "f",
+                    "gamma", "log-normal", "logistic", "normal", "weibull"), nmb=1)
```

...najlepszy jest `"weibull"`, od którego `"inverse gaussian"`, nie dający się wywołać z żadną z metod `fitdistr`, nie jest lepszy - używamy metody momentów.

Porównajmy `"weibull"` z `"inverse gaussian"` dla `b.loss[[1]]`:

```
> x<- period.loss(b.loss[[1]],"days")
> m <- mean(x)
> v <- var(x)
> lambda = max(m^3/v, 0.1^(100))
> nu = max(m, 0.1^(100))
> par(mfrow = c(2,2))
> loss.fit.dist("weibull",x,xlog.scale=T)
```

```
$shape
[1] 0.611712

$scale
[1] 53922.56

          ad
3.713546e-05

> fit.plot(x,dinvGauss,distname = "i.g.",param = list(lambda = lambda, nu = nu),
+ log="x")

[1] 5.641139e-05

> loss.fit.dist("weibull",x)

$shape
[1] 0.611712

$scale
[1] 53922.56

          ad
3.713546e-05

> fit.plot(x,dinvGauss,distname = "i.g.",param = list(lambda = lambda, nu = nu))

[1] 5.641139e-05
```

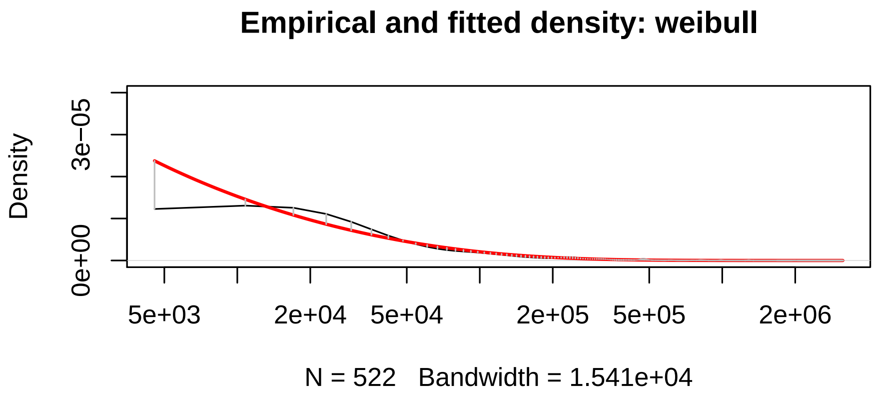


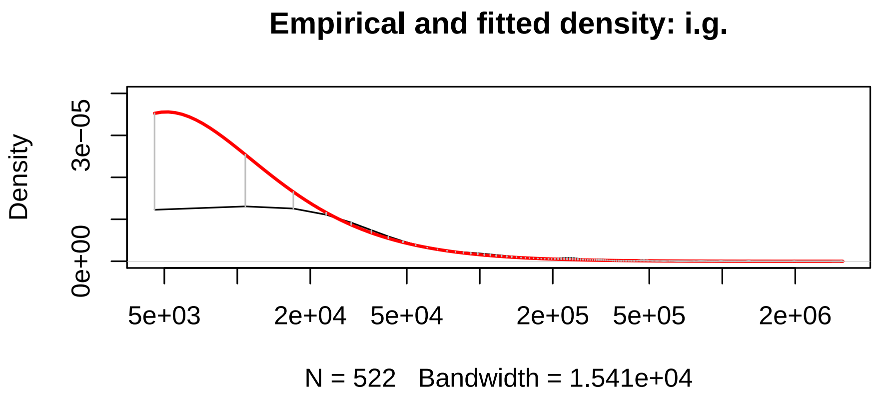
Rysunek 6.4: Dopasowania rozkładów: odwrotny gaussowski i Weibulla dla strat z „Agency Services”

Oba wydają się bardzo dobre, ale `"inverse gaussian"` wydaje się lepszy. Dlaczego nie został wybrany? Odpowiedź mamy na poniższym obrazku

```
> par(mfrow = c(2,1))
> loss.fit.dist("weibull",x,xlog.scale=T,draw.diff=T,ylim = c(0,40e-06))

$shape
[1] 0.611712

$scale
[1] 53922.56
```

```
          ad
3.713546e-05

> fit.plot(x,dinvGauss,distname = "i.g.",param = list(lambda = lambda, nu = nu),
+            log="x",draw.diff=T,ylim = c(0,40e-06))

[1] 5.641139e-05
```





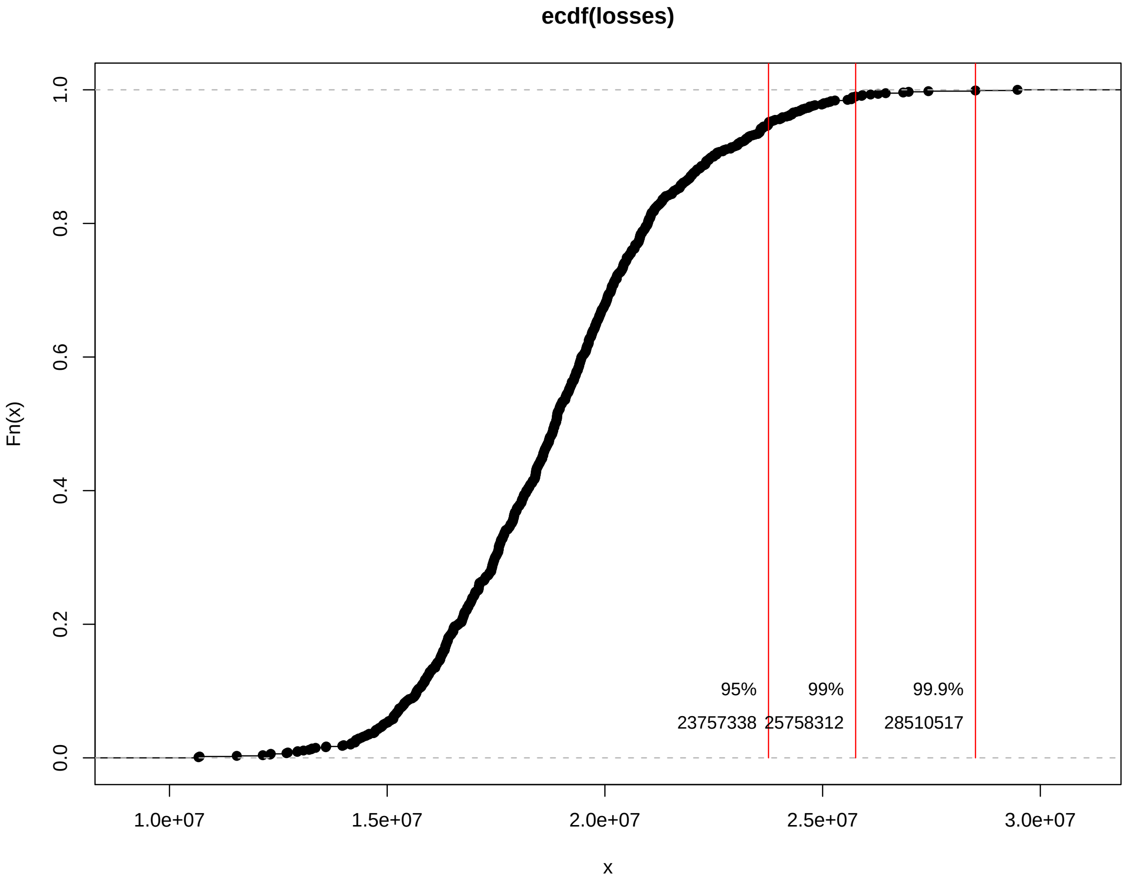
Jeśli chcemy lepszej estymacji ogona rozkładu, `"inverse gaussian"` mógłby być nieco lepszy. Mimo to tym razem wybieramy `"weibull"`:

```
> b1 <- mc(b.loss[[1]], rfun = "weibull")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 377.8318  7 1.348104e-77

         Goodness-of-fit test for binomial distribution

                      X^2 df      P(> X^2)
Likelihood Ratio 530.4053  7 2.319596e-110

         Goodness-of-fit test for nbinomial distribution
```

```
                      X^2 df     P(> X^2)
Likelihood Ratio 22.55354  6 0.0009606621
   nbinomial
0.0009606621

         Goodness-of-fit test for nbinomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 22.55354  6 0.0009606621
$size
[1] 0.5730788

$prob
[1] 0.464733

$shape
[1] 0.611712

$scale
[1] 53922.56
```

> b1$q

```
     95%      99%    99.9%
23757338 25758312 28510517
```



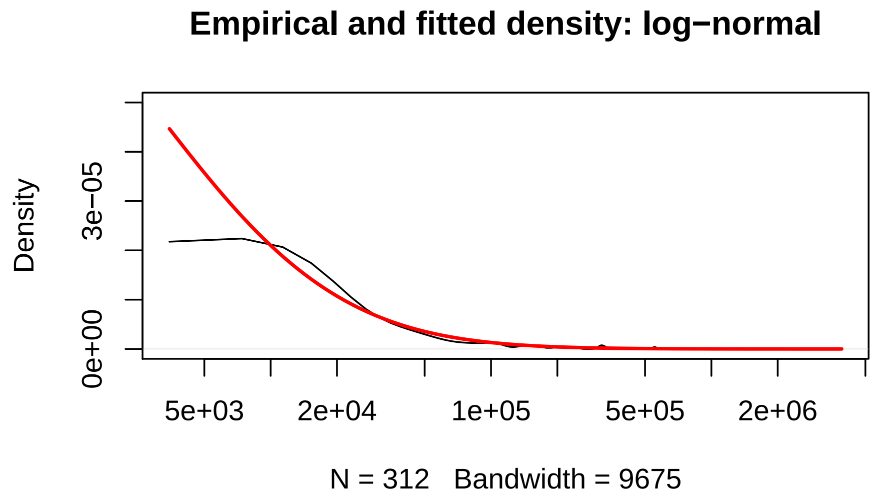
Rysunek 6.5: Empiryczna dystrybuanta dla tysiąca rocznych strat zasymulowanych dla „Agency Services”

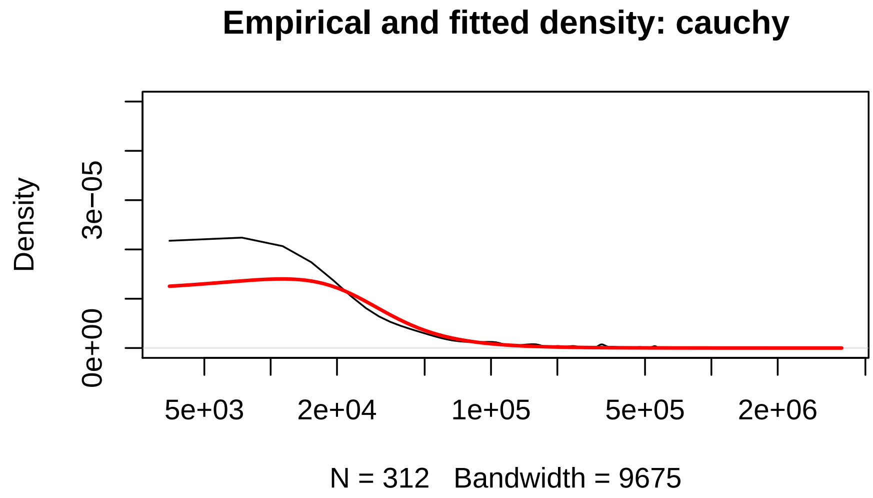
6.2.2. „Asset Management”

Dla `b.loss[[2]]` mamy rozkład `"cauchy"` jako najlepszy rozkład pod względem `ad`, ale `"log-normal"` mimo to może być lepszy. Użyjmy funkcji `loss.fit.dist()`:

```
> par(mfrow = c(2, 1))
> x <- period.loss(b.loss[[2]], "days")
> loss.fit.dist("log-normal", b.loss[[2]], xlog.scale = T,
+     n = 1000, ylim = c(0, 5e-05))

$meanlog
[1] 9.656914

$sdlog
[1] 1.844185

          ad
7.532465e-05
```

```
> loss.fit.dist("cauchy", b.loss[[2]], xlog.scale = T,
+     n = 1000, ylim = c(0, 5e-05))

$location
[1] 11273.96

$scale
[1] 22714.10

          ad
7.237527e-05
```

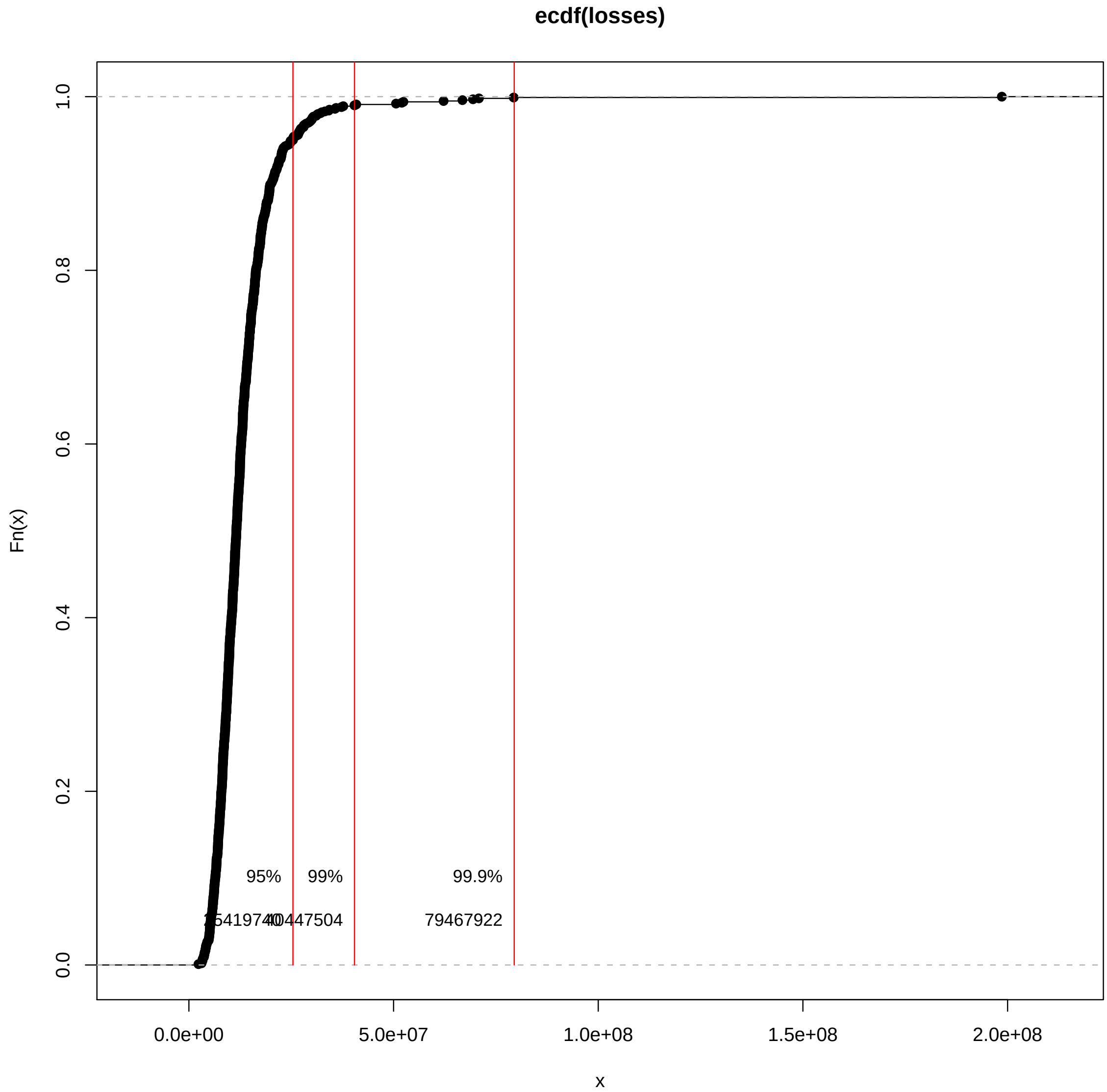




Rysunek 6.6: Dopasowania rozkładów: Cauchy i log-normal dla „Asset Management”

Jak widzimy, wartości gęstości rozkładu "log-normal" są raczej większe od empirycznych wartości gęstości, podczas gdy wartości gęstości rozkładu "cauchy" są raczej mniejsze. W związku z tym, być może bezpieczniej będzie wybrać rozkład "lognormal":

```
> b2 <- mc(b.loss[[2]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 208.9491  5 3.454384e-43

         Goodness-of-fit test for binomial distribution

                     X^2 df     P(> X^2)
Likelihood Ratio 266.269  5 1.770535e-55

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 6.64929  4 0.1556236
nbinomial
0.1556236

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 6.64929  4 0.1556236
$size
[1] 0.2164420

$prob
[1] 0.4979963

$meanlog
[1] 10.17238

$sdlog
[1] 1.902329

> b2$q

     95%      99%    99.9%
25419740 40447504 79467922
```



Rysunek 6.7: Empiryczna dystrybuanta dla tysiąca rocznych strat zasymulowanych dla „Asset Management”

Mamy duże różnice pomiędzy kwantylami - zobacz `b2$q` i rysunek 6.7. Wynik jest nie lepszy, a wręcz gorszy, dla rozkładu `"cauchy"`.

6.2.3. „Commercial Banking”

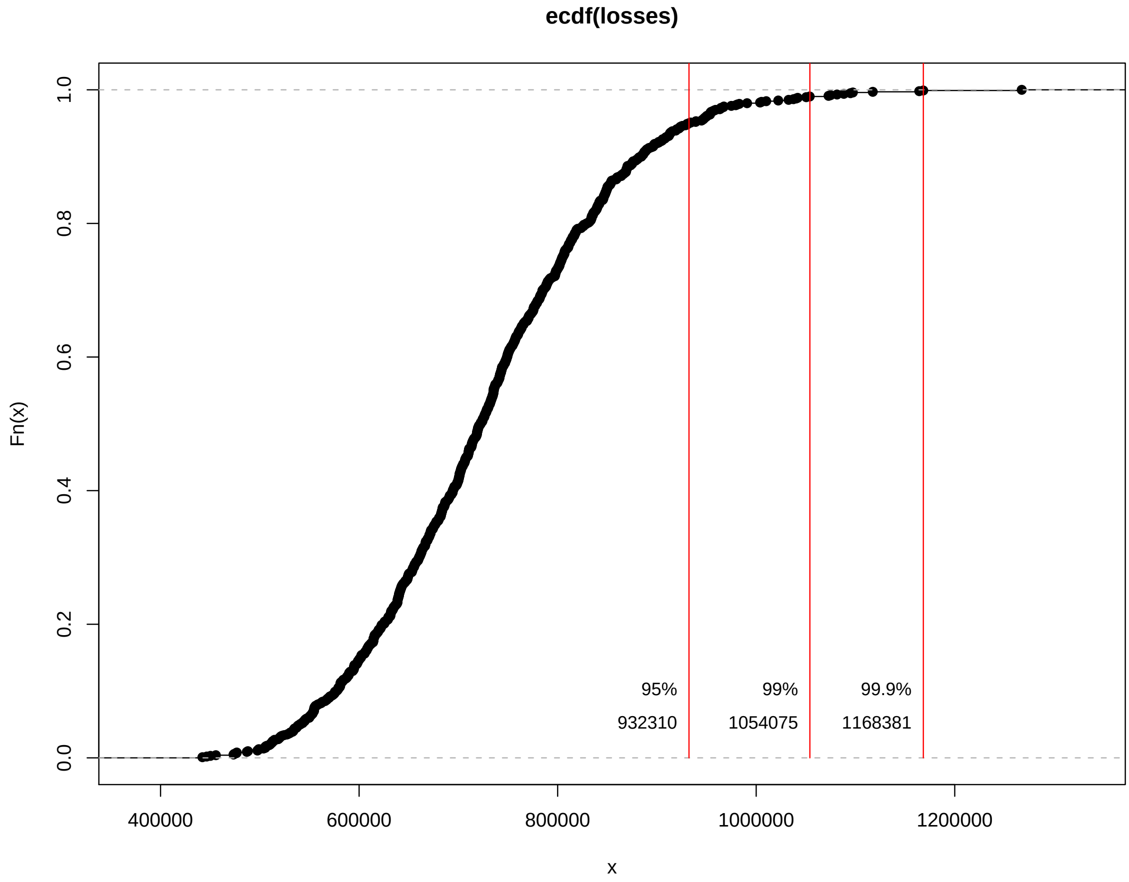
Dla `b.loss[[3]]` mamy rozkład `"log-normal"`; ale `"inverse gaussian"` też jest bardzo dobry.

```
> b3 <- mc(b.loss[[3]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df      P(> X^2)
```

```
Likelihood Ratio 173.4832  4 1.869881e-36

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 253.3477  4 1.236929e-53

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
Likelihood Ratio 8.722379  3 0.03321907
 nbinomial
0.03321907

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
Likelihood Ratio 8.722379  3 0.03321907
$size
[1] 0.4113998

$prob
[1] 0.5711386

$meanlog
[1] 8.331045

$sdlog
[1] 0.9357786
```

> b3$q

```
     95%       99%     99.9%
 932310.2 1054075.5 1168380.9
```



Rysunek 6.8: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Commercial Banking"

6.2.4. „Corporate Finance"

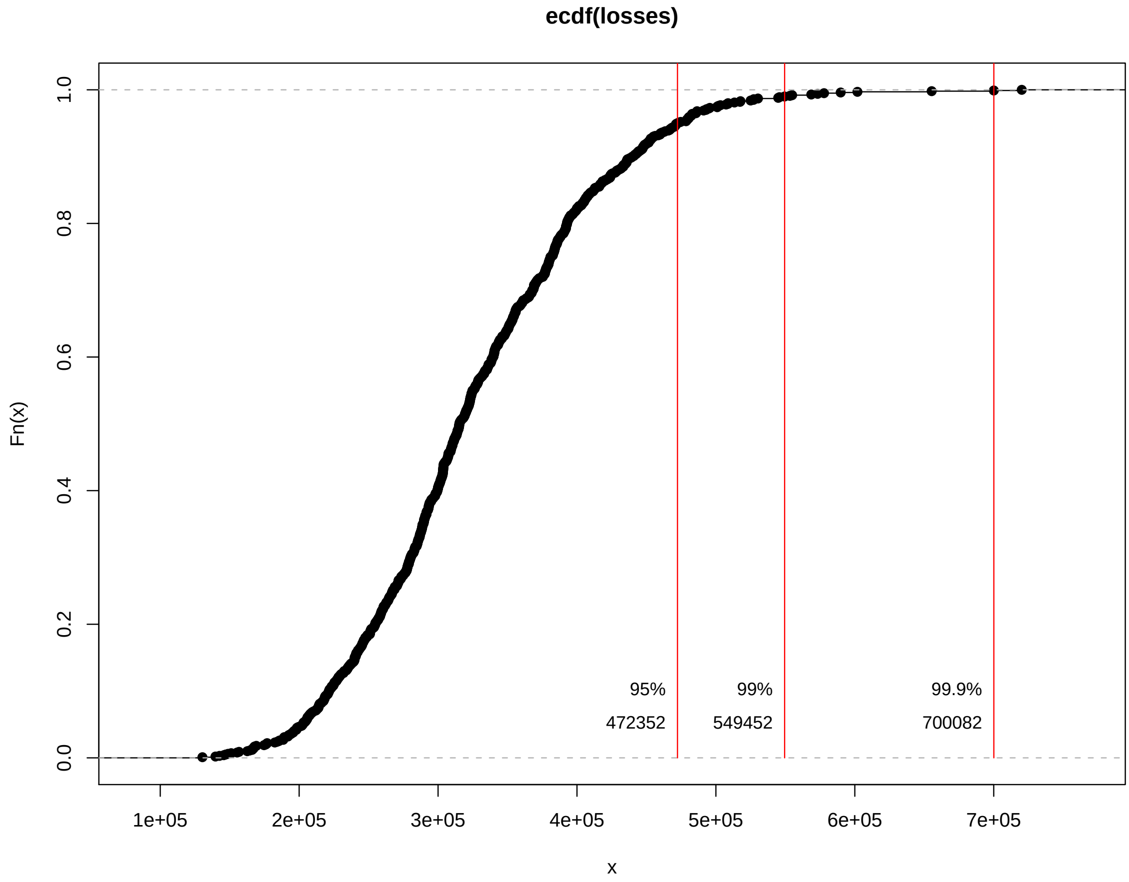
Dla `b.loss[[4]]` mamy rozkład `"log-normal"`, ale znów `"inverse gaussian"` też jest bardzo dobry.

```
> b4 <- mc(b.loss[[4]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 95.11894  3 1.740823e-20

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 130.0241  3 5.344058e-28

         Goodness-of-fit test for nbinomial distribution
```

```
                     X^2 df  P(> X^2)
Likelihood Ratio 3.90774  2 0.1417245
nbinomial
0.1417245

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 3.90774  2 0.1417245
$size
[1] 0.2181131

$prob
[1] 0.6194936

$meanlog
[1] 8.306907

$sdlog
[1] 0.9849417
```

> b4$q

```
     95%      99%    99.9%
472351.9 549452.4 700082.3
```



Rysunek 6.9: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Corporate Finance”

6.2.5. „Payment & Settlement”

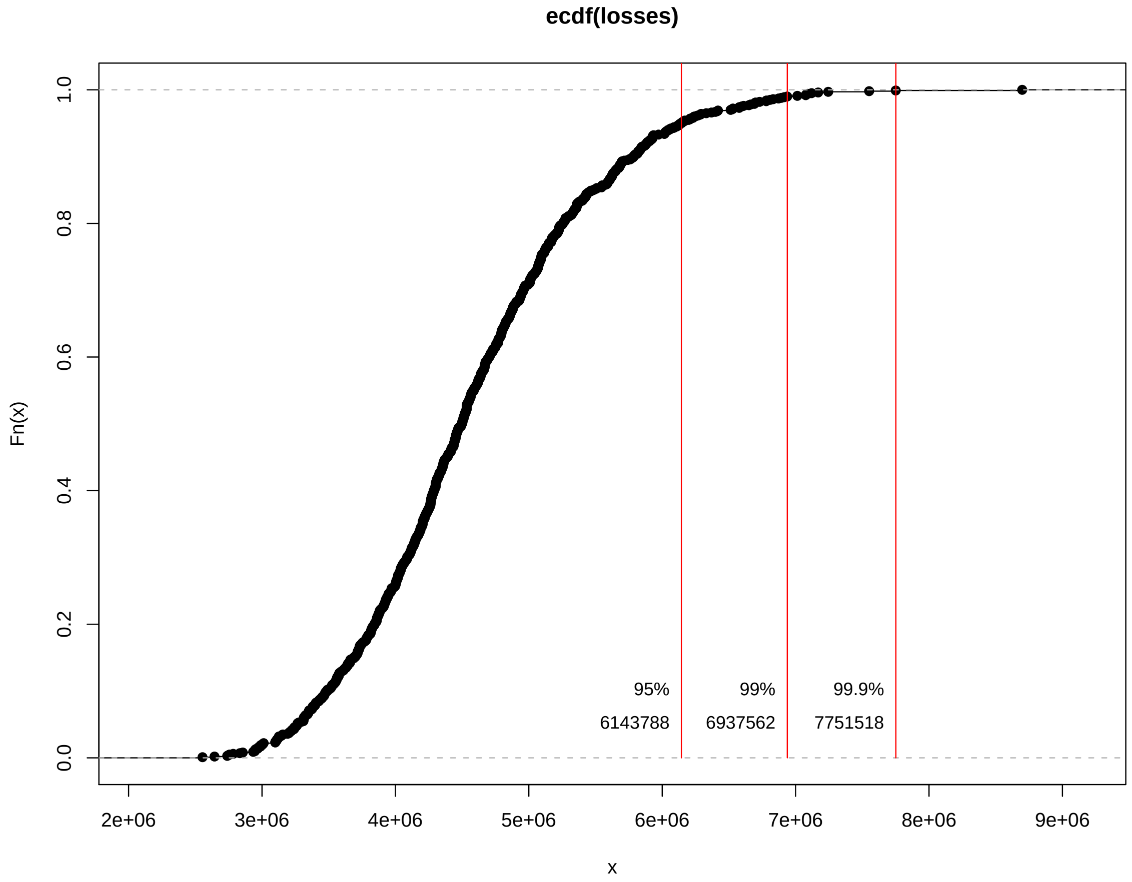
To samo dla `b.loss[[5]]`:

```
> b5 <- mc(b.loss[[5]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 280.0119  5 1.978837e-58

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 418.9128  5 2.485152e-88

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
```

```
Likelihood Ratio 10.18445  4 0.03743263
 nbinomial
0.03743263

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
Likelihood Ratio 10.18445  4 0.03743263
$size
[1] 0.4899842

$prob
[1] 0.5052239

$meanlog
[1] 9.325801

$sdlog
[1] 1.296436
```

> b5$q

```
    95%     99%   99.9%
6143788 6937562 7751518
```



Rysunek 6.10: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Payment & Settlement”

6.2.6. „Retail Banking”

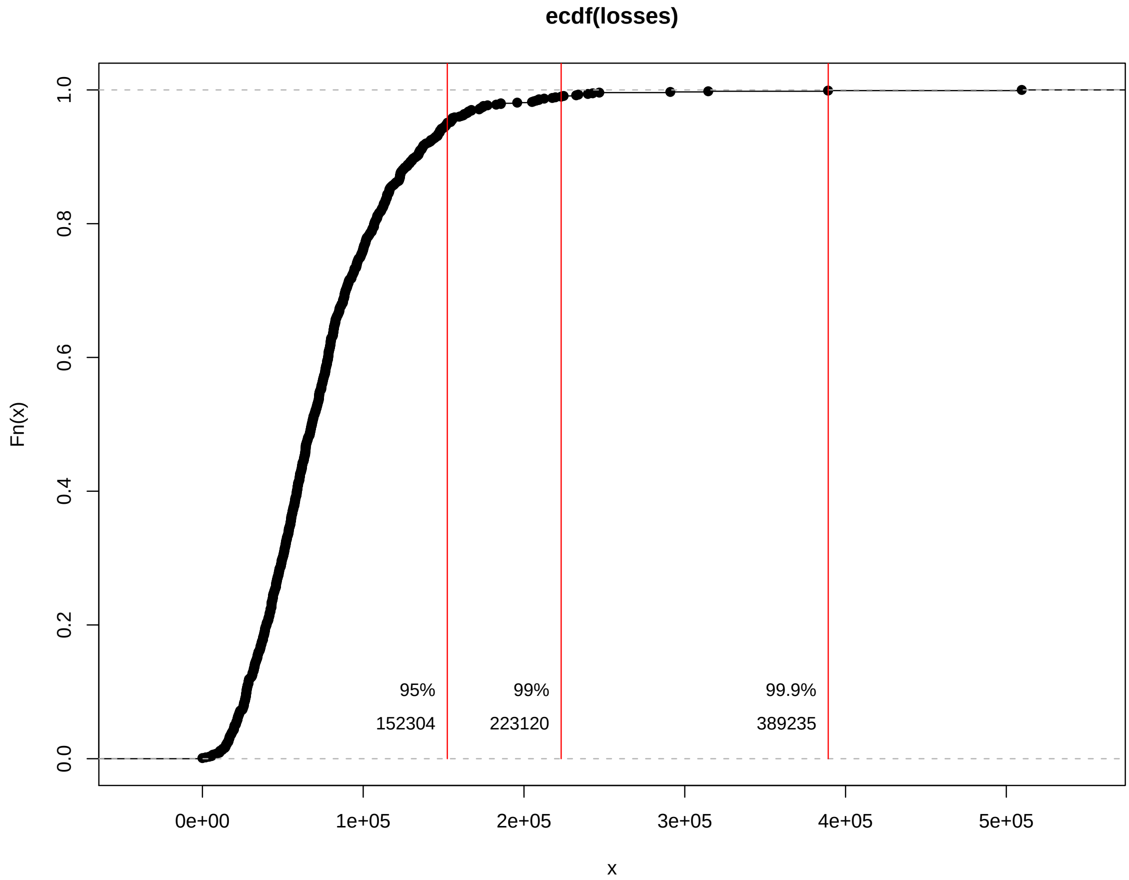
Mamy podobne problemy z rozkładem `"cauchy"` dla `b.loss[[6]]`, jak w przypadku linii biznesowej `"Asset Management"` i danych `b.loss[[2]]`, więc znów wybierzemy rozkład `"log-normal"`:

```
> b6 <- mc(b.loss[[6]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 12.57757  1 0.0003904038

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 17.69507  1 2.592976e-05

         Goodness-of-fit test for nbinomial distribution
```

```
                      X^2 df P(> X^2)
Likelihood Ratio 1.295086  0        0
     poisson
0.0003904038

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 12.57757  1 0.0003904038
$lambda
[1] 0.02433936

$meanlog
[1] 8.502203

$sdlog
[1] 1.027541
```

> b6$q

```
     95%      99%    99.9%
152303.6 223119.8 389235.0
```



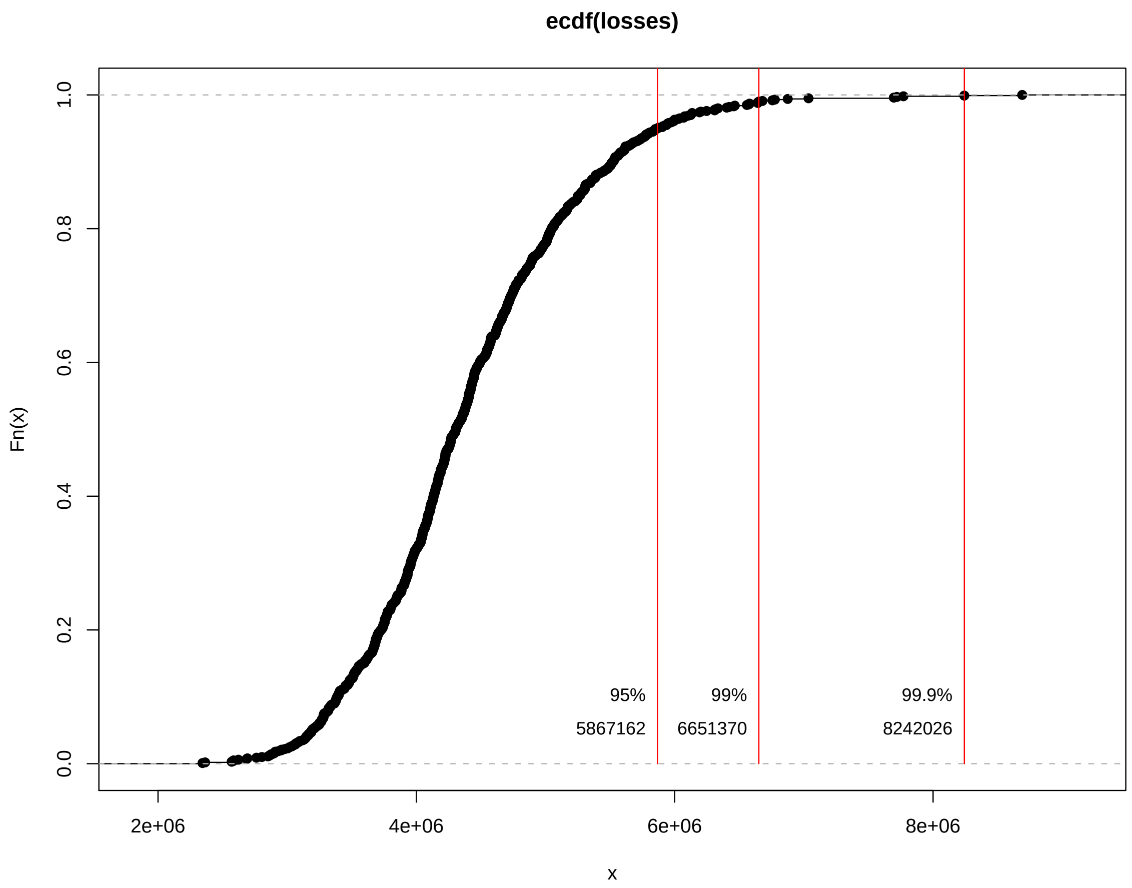
Rysunek 6.11: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Retail Banking”

6.2.7. „Retail Brokerage”

Znów `"log-normal"` dla `b.loss[[7]]`:

```
> b7 <- mc(b.loss[[7]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 223.5155  5 2.623081e-46

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 335.5067  5 2.306178e-70

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
```

```
Likelihood Ratio 12.11575  4 0.01651093
 nbinomial
0.01651093

         Goodness-of-fit test for nbinomial distribution

                      X^2 df   P(> X^2)
Likelihood Ratio 12.11575  4 0.01651093
$size
[1] 0.5650693

$prob
[1] 0.5454443

$meanlog
[1] 9.351597

$sdlog
[1] 1.265011

> b7$q

    95%     99%   99.9%
5867162 6651370 8242026
```



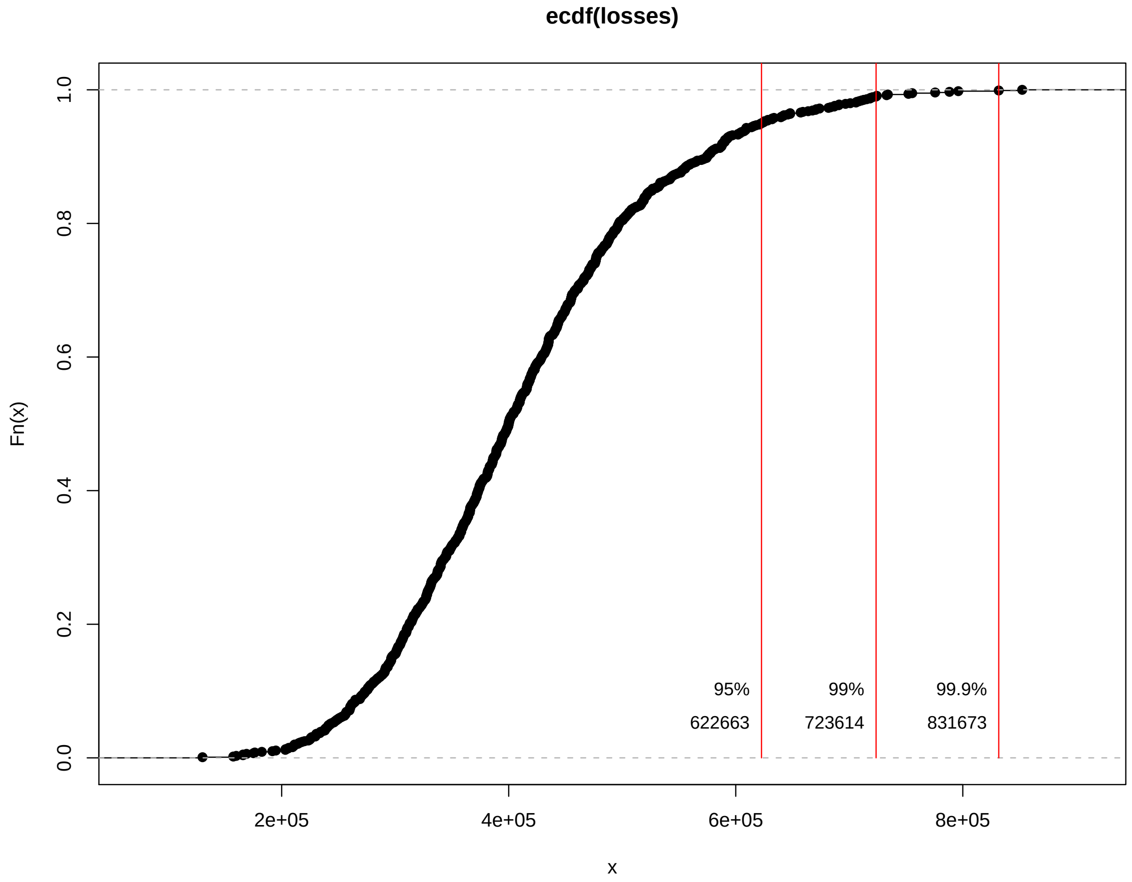
Rysunek 6.12: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Retail Brokerage”

6.2.8. „Trading & Sales”

. . . i tak samo dla `b.loss[[8]]`:

```
> b8 <- mc(b.loss[[8]], rfun = "log-normal")$table

         Goodness-of-fit test for poisson distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 116.2838  3 4.871768e-25

         Goodness-of-fit test for binomial distribution

                      X^2 df     P(> X^2)
Likelihood Ratio 157.1107  3 7.702578e-34

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
```

```
Likelihood Ratio 4.099084  2 0.1287939
nbinomial
0.1287939

         Goodness-of-fit test for nbinomial distribution

                      X^2 df  P(> X^2)
Likelihood Ratio 4.099084  2 0.1287939
$size
[1] 0.1796925

$prob
[1] 0.5794828

$meanlog
[1] 8.490445

$sdlog
[1] 1.065124

> b8$q

     95%      99%    99.9%
622662.9 723614.4 831673.0
```



Rysunek 6.13: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla „Trading & Sales”

6.2.9. Podsumowanie

Teraz krótkie podsumowanie:

```
> b <- rbind(b1$q, b2$q, b3$q, b4$q, b5$q, b6$q, b7$q,
+     b8$q)
> b

            95%        99%      99.9%
[1,] 23757338.3 25758312.0 28510517.4
[2,] 25419739.6 40447503.7 79467922.3
[3,]   932310.2  1054075.5  1168380.9
[4,]   472351.9   549452.4   700082.3
[5,]  6143787.8  6937561.6  7751517.9
[6,]   152303.6   223119.8   389235.0
[7,]  5867162.1  6651369.7  8242026.3
[8,]   622662.9   723614.4   831673.0
```

To są nasze kwantyle dla linii biznesowych, złączone po wierszach. Teraz zsumujmy dane strat po liniach biznesowych:

```
> b.loss.sum <- NULL
> for (i in 1:length(loss.data.object$blines)) {
+     b.loss.sum[i] <- sum(b.loss[[i]][, 2])
+ }
```

Mamy `b.loss.sum/4` - średnie straty roczne dla linii biznesowych dla naszego okresu około czterech lat:

```
> yearly.b.loss<- b.loss.sum/4
> yearly.b.loss

[1] 11766481.26  8901222.83   517504.47   261595.07
[5]  2617273.66    69289.07  2976987.77   343875.76
```

Jakie są kwoty, które potrzebujemy trzymać w stosunku do tych średnich rocznych strat dla linii biznesowych?

```
> fraction1 <- b[, 1]/yearly.b.loss
> fraction1

[1] 2.019069 2.855758 1.801550 1.805660
[5] 2.347400 2.198090 1.970838 1.810721
```

To oznacza, że dla uzyskania 95 procentowej pewności dla `"Agency Services"` potrzebujemy kwoty dwa razy większej, niż `yearly.b.loss[1]`. Ta kwota wynosi:

```
> fraction1[1]*yearly.b.loss[1]

[1] 23757338
```

Zobaczmy prawdziwe roczne straty dla pierwszej linii biznesowej; y będzie danymi strat dla `"Agency Services"`; y2 jest sumą strat z 2002; `min(as.Date(y[,1]) = "2002-01-08"`.

```
> y<- b.loss[[1]]
> y2<- sum(y[as.Date(y[,1])<as.Date("2003-01-01"),2])
> y2

[1] 3390290
```

. . . y3 jest sumą strat z 2003:

```
> y3<- sum(y[as.Date("2003-01-01")<=as.Date(y[,1]) &
+                    as.Date(y[,1])<as.Date("2004-01-01"),2])
> y3

[1] 8792141
```

. . . y4 jest sumą strat z 2004:

```
> y4<- sum(y[as.Date("2004-01-01")<=as.Date(y[,1]) &
+                    as.Date(y[,1])<as.Date("2005-01-01"),2])
> y4
```

```
[1] 12663557
```

...oraz y5 jest sumą strat z 2005:

```
> y5<- sum(y[as.Date("2005-01-01")<=as.Date(y[,1]) &
+                    as.Date(y[,1])<as.Date("2006-01-01"),2])
> y5
```

```
[1] 19903260
```

Zostaje jeszcze rok 2006, ale `max(as.Date(y[,1])) = "2006-02-12"`, więc dane z tego roku - 44 straty - są niekompletne.

Porównajmy wynik `fraction1[1]*yearly.b.loss[1] = 23757338` z:

```
> yearly.b.loss[1] + 2*sd(c(y2,y3,y4,y5))
```

```
[1] 25655443
```

Dla rozkładu normalnego to dałoby około 96 procentowy kwantyl. Mamy dość duże odchylenie standardowe dla naszych rocznych strat:

```
> sd(c(y1,y2,y3,y4))
```

```
[1] 6944481
```

...a w stosunku do średniej rocznej straty `"Agency Services"` mamy:

```
>100*sd(c(y1,y2,y3,y4))/yearly.b.loss[1]
```

```
[1] 59.01918
```

...zatem mogliśmy oczekiwać wartości `fraction1[1]*yearly.b.loss[1]` dość różnej od `yearly.b.loss[[1]]`.
Zauważmy jeszcze tylko, że straty z lat 2002-2005 przejawiają tendencję wzrostową, co może być związane np. z rozrostem banku i co powinno być w takim przypadku uwzględnione (należałoby jakoś przeskalować wyniki). Nie będziemy się tu jednak zajmować tym zagadnieniem.
Wykonajmy te same obliczenia dla kwantyli 99 i 99.9 - procentowych:

```
> fraction2 <- b[, 2]/yearly.b.loss
> fraction2
```

```
[1] 2.189126 4.544039 2.036843 2.100393
[5] 2.650683 3.220130 2.234262 2.104290
```

```
> fraction3 <- b[, 3]/yearly.b.loss
> fraction3
```

```
fraction3
[1] 2.423028 8.927753 2.257721 2.676206
[5] 2.961677 5.617553 2.768579 2.418528
```

Zauważmy, że dla drugiej i szóstej linii biznesowej wymagane kwoty rosną raczej szybko przy zmianie kwantyla. Były to linie biznesowe z gęstościami, do których dobrze pasował rozkład `"cauchy`, nie posiadający średniej ani odchylenia standardowego.

Oczywiście VaR nie jest koherentną miarą ryzyka i wartości `b` nie powinny być sumowane i porównywane z sumą strat bez dodatkowych założeń.

6.3. Wszystkie komórki

Teraz policzymy VaR dla każdej niepustej komórki „linia biznesowa/kategoria ryzyka". Potrzebujemy wybrać najlepsze dopasowanie dla każdej komórki. Utwórzmy macierz zerową `m`:

```
> m <- matrix(0, length(loss.data.object$blines), length(loss.data.object$rcateg))
```

Dla każdej niepustej komórki numer najlepiej dopasowanego rozkładu z `flist` będzie przypisany do `m[i,j]` gdzie `i`, `j` to numery linii biznesowej i kategorii ryzyka tej komórki. Lista `flist` składa się z zaledwie `5` rozkładów, ale są to rozkłady najlepiej pasujące do tych danych. Dobre dopasowania dostajemy również dla rozkładu `"cauchy"`, ale wyłączamy go z podobnych jak wcześniej powodów.

```
> flist = c("log-normal", "weibull", "gamma", "beta",  "exponential")
>  for (i in 1:8) {
+  for (j in 1:7) {
+         y <- read.loss(b = i, r = j, loss.data.object)
+         if (dim(y)[1] != 0) {
+         ad <- 100
+         value <- 1
+                 for (k in flist) {
+                 ad.new <- loss.fit.dist(densfun = k,
+                 x = y, period = "days", n = 10000)$ad
+                 if (ad.new < ad) {
+                   ad <- ad.new
+                   value <- which(flist == k)
+                 }
+             }
+             m[i, j] <- value
+         }
+ }
+ }
```

Otrzymana macierz:

```
> m

     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    0    1    0    0    0    1    3
[2,]    0    0    0    0    2    0    1
[3,]    0    1    0    4    0    1    3
[4,]    0    0    4    0    1    1    0
[5,]    0    0    1    5    5    0    2
[6,]    0    0    0    0    1    0    4
[7,]    0    1    0    0    5    0    5
[8,]    0    0    1    0    0    0    0
```

Przykładowo: `m[5,7]=2` oznacza, że dla piątej linii biznesowej i siódmej kategorii ryzyka do strat w komórce dopasowano rozkład `flist[2] = "weibull"`.

Możemy także posortować wartości tej macierzy ...

```
> sort(as.vector(m))

 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[29] 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 2 2 3 3 4 4 4 5 5 5 5
```

. . . aby zobaczyć wyraźnie, że mamy 34 puste komórki oraz że rozkład `"lognormal"` mamy 11 razy, rozkład `"weibull"` 2 razy, rozkład `"gamma"` 2 razy, rozkład `"beta"` 3 razy i rozkład `"exponential"` 4 razy.

Dla każdej niepustej komórki zostaną wyliczone trzy kwantyle. Użyjemy funkcji `mc()` z argumentem `rfun =flist[m[i,j]]`, gdzie `i`, `j` są odpowiednio numerami linii biznesowej i kategorii ryzyka dla danej komórki. Otrzymamy sumę wszystkich kwantyli dla niepustych komórek względem kwantyli.

```
> q <- c(0, 0, 0)
> for (i in 1:length(loss.data.object$blines)) {
+     for (j in 1:length(loss.data.object$rcateg)) {
+         if (m[i, j] != 0) {
+             y <- read.loss(b = i, r = j, loss.data.object)
+             q.new <- mc(y, rfun = flist[m[i, j]])$table$q
+             q <- apply(rbind(q, q.new), 2, sum)
+         }
+     }
+ }

> sum.all.rect <- q
```

Otrzymujemy:

```
> sum.all.rect

      95%       99%     99.9%
 65876010  92248519 153280063
```

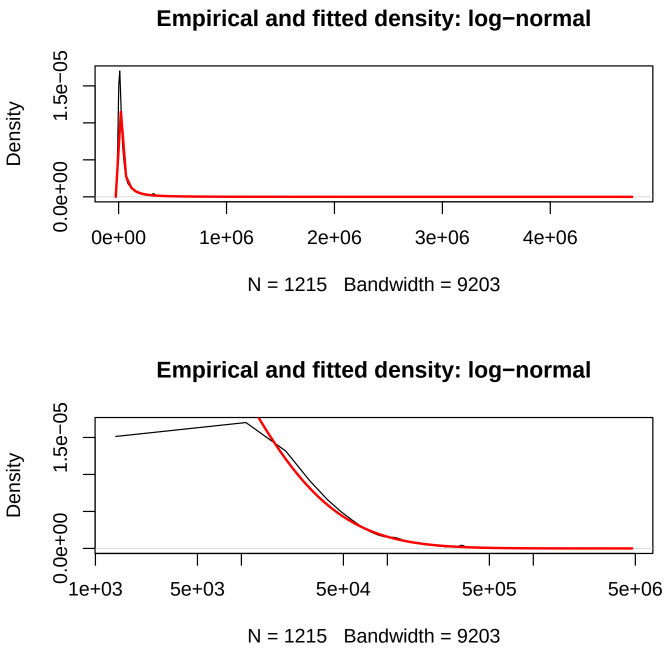
6.4. Wszystkie straty

A teraz dla wszystkich strat razem, bez podziału na kategorie:

```
> all.losses <- loss.data.object$losses[, 3:4]
> head(all.losses)

  First_Date_of_Event Gross_Loss_Amount
1          2002-01-03           1642.26
2          2002-01-06           2498.33
3          2002-01-08           7420.72
4          2002-01-09          27019.26
5          2002-01-10           1829.98
6          2002-01-11          12164.67
```

Dla tych danych najlepszy jest rozkład `"lognormal"`:

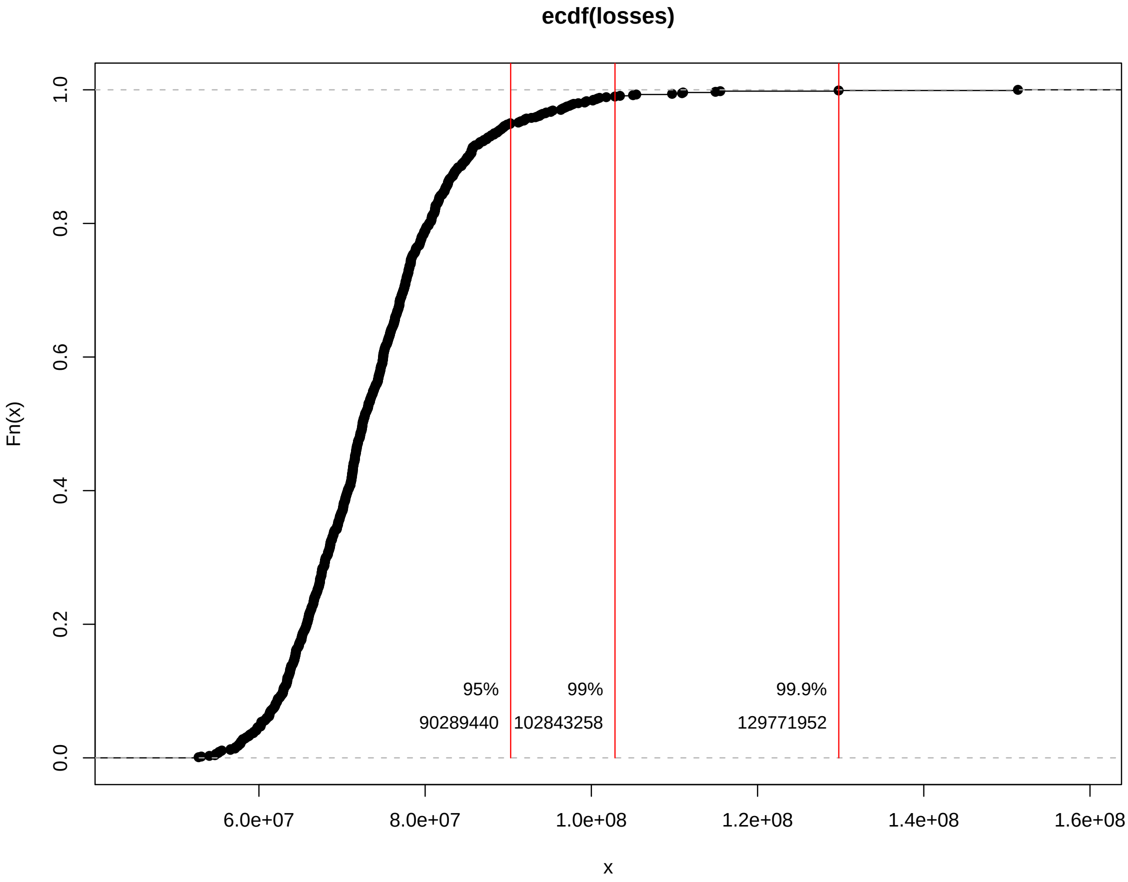
```
> par(mfrow = c(2, 1))
> loss.fit.dist("log-normal", all.losses, period = "days")

$meanlog
[1] 9.99138

$sdlog
[1] 1.638526

          ad
4.036907e-05

> loss.fit.dist("log-normal", all.losses, period = "days",
+     xlog.scale = T)

$meanlog
[1] 9.99138

$sdlog
[1] 1.638526

          ad
4.036907e-05
```



Rysunek 6.14: Dopasowanie rozkładu log-normal do wszystkich strat; skale normalna i logarytmiczna

A teraz obliczymy VaR:

```
> sum.all <- mc(all.losses, "log-normal")$table$q

         Goodness-of-fit test for poisson distribution

                     X^2 df     P(> X^2)
Likelihood Ratio 415.5948 11 2.983464e-82

         Goodness-of-fit test for binomial distribution

                     X^2 df      P(> X^2)
Likelihood Ratio 879.2554 11 1.784866e-181

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 12.80476 10 0.2347937
nbinomial
0.2347937

         Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 12.80476 10 0.2347937
$size
[1] 2.389882

$prob
[1] 0.4987338

$meanlog
[1] 9.99138

$sdlog
[1] 1.638526
```



Rysunek 6.15: Empiryczna dystrybuanta dla tysiąca rocznych strat symulowanych dla wszystkich danych

6.5. Value at Risk dla komórek, linii biznesowych i dla wszystkich strat

Podsumujmy:

```
> sum.all.blines

     95%       99%     99.9%
 63367656  82345009 127061355

> sum.all.rect

     95%       99%     99.9%
 65876010  92248519 153280063

> sum.all

     95%       99%     99.9%
 90289440 102843258 129771952
```

Różnice między odpowiednimi kwantylami nie są bardzo duże, zważywszy, że przez `VaR` nie jest podaddytywną miarą ryzyka, więc sumowanie strat dla poszczególnych komórek lub linii biznesowych nie tylko nie musi dawać oszacowania z góry wartości `VaR` dla wszystkich danych łącznie, ale i może drastycznie się różnić tej wartości, jak pamiętamy z `Przykładu` 1.0.1.

Dla wiarygodnej estymacji powinniśmy mieć więcej informacji o tych danych, aby móc poczynić pewne założenia. Dla lepszej estymacji VaR-u powinniśmy dokonać jeszcze przynajmniej estymacji częstości i dotkliwości dla tygodni, po czym zasymulować VaR, jednak spowodowałoby to znaczne powiększenie objętości tej pracy, więc zostawmy to w tym punkcie.

6.6. Podsumowanie

Głównym wynikiem pracy jest napisany przeze mnie pakiet `opVaR`, którego działanie przedstawiłam w pracy na wybranych przykładach. Zostały zaprezentowane wszystkie kluczowe dla pakietu funkcje, mianowicie: `read.loss()`, `period.loss()`, `loss.density()`, `loss.matrix()`, `loss.matrix.image()`, `hist.period()`, `root.period()`, `fit.plot()`, `loss.fit.dist()` i wreszcie funkcja `mc()`, służąca do przeprowadzania symulacji metodą Monte Carlo. Końcowe obliczenia są zwieńczeniem procesu wczytywania i przetwarzania danych o stratach. Po dopasowaniu odpowiednich rozkładów częstości i dotkliwości strat w niepustych komórkach oraz w liniach biznesowych, metodą symulacji Monte Carlo, wykonanej dla `1000` lat, wyliczyłam `VaR` dla wszystkich komórek i dla każdej linii biznesowej. Następnie przeprowadziłam takie same obliczenia dla wszystkich strat, bez podziału na jakiekolwiek kategorie.

W pracy przedstawiłam także pewne definicje i założenia dotyczące ryzyka operacyjnego oraz opisałam wybrane głównie pod kątem obliczania `VaR` dla ryzyka operacyjnego punkty *Bazylei II* - zbioru rekomendacji odnośnie zarządzania owym ryzykiem.

Dodatek A

Lista rozkładów

Poniżej przedstawiam rozkłady prawdopodobieństwa wykorzystane w pakiecie `opVaR`.

A.1. Rozkłady dyskretne

A.1.1. Rozkład Poissona z parametrem λ

$$P(X=k)=\frac{e^{-\lambda}\lambda^k}{k!}$$
$$E(X)=\lambda$$
$$Var(X)=\lambda$$

A.1.2. Rozkład dwumianowy z parametrami q, n $(0<q<1, n\ naturalne)$

$$P(X=k)=\binom{n}{k}q^k(1-q)^{n-k},\quad k=0,1,...,n$$
$$E(X)=nq$$
$$Var(X)=nq(1-q)$$

A.1.3. Rozkład ujemny dwumianowy z parametrami $q,\ r$

$$P(X=k)=\binom{r+k-1}{k}q^k(1-q)^r,\quad k=0,1,...$$
$$E(X)=r\frac{q}{1-q}$$
$$Var(X)=r\frac{q}{(1-q)^2}$$

A.2. Rozkłady ciągłe

A.2.1. Rozkład beta

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \quad 0 \leqslant x \leqslant 1, \; a, b > 0$$

$$E(x) = \frac{a}{a+b}$$

$$Var(X) = \frac{ab}{(a+b)^2(a+b+1)}$$

A.2.2. Rozkład Cauchy'ego

$$f(x) = \frac{h}{\pi((x-l)^2 + h^2)}, \quad h > 0$$

Wartość oczekiwana nie istnieje.

A.2.3. Rozkład χ^2

$$f(x) = \begin{cases} \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} & x > 0 \\ 0 & x \leqslant 0 \end{cases}$$

$$E(X) = n$$

$$Var(X) = 2n$$

A.2.4. Rozkład wykładniczy

$$f(x) = \lambda e^{-\lambda x}, \quad x \geqslant 0, \quad \lambda > 0$$

$$E(X) = \frac{1}{\lambda}$$

$$Var(X) = \frac{1}{\lambda^2}$$

A.2.5. Rozkład Fishera

$$f(x) = \begin{cases} (\frac{n_1}{2})^{\frac{n_1}{2}} (\frac{n_2}{2})^{\frac{n_2}{2}} \frac{\Gamma(\frac{n_1}{2}+\frac{n_2}{2})}{\Gamma(\frac{n_1}{2})\Gamma(\frac{n_2}{2})} \frac{x^{\frac{n_1}{2}-1}}{(\frac{n_1}{2}x+\frac{n_2}{2})^{\frac{n_1}{2}+\frac{n_2}{2}}} & x > 0 \\ 0 & x \leqslant 0 \end{cases}$$

$$E(X) = \frac{n_2}{n_2 - 2}$$

$$Var(X) = \frac{2n_2^2(n_1 + n_2 - 2)}{n_1(n_2-2)^2(n_2-4)}$$

A.2.6. Rozkład gamma

$$f(x) = \frac{1}{s^a\Gamma(a)} x^{a-1} e^{-\frac{x}{s}}, \quad x \geqslant 0, \quad a, s > 0$$

$$E(X) = as$$

$$Var(X) = as^2$$

A.2.7. Rozkład odwrotny gaussowski

$$f(x) = (\frac{\lambda}{2\pi x^3})^{\frac{1}{2}} exp(-\frac{\lambda(x-\mu^2)}{2x\mu^2}), \quad \nu, \ \lambda > 0$$

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}} exp(-\frac{\lambda(x-\mu^2)}{2x\mu^2}), \quad \nu, \lambda > 0$$

$$E(X) = \mu$$

$$Var(X) = \frac{\mu^3}{\lambda}$$

A.2.8. Rozkład logarytmiczny normalny

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} exp(-\frac{(log(x)-\mu)^2}{2\sigma^2}), \quad \sigma \geqslant 0, x > 0$$

$$E(X) = exp(\mu + \frac{1}{2}\sigma^2)$$

$$Var(X) = (exp(\sigma^2) - 1)(2\mu + \sigma^2)$$

A.2.9. Rozkład logistyczny

$$f(x) = \frac{1}{s} exp(\frac{x-m}{s})(1 + exp(\frac{x-m}{s}))^2$$

$$E(X) = m$$

$$Var(X) = \frac{\pi^2}{3} s^2$$

A.2.10. Rozkład normalny

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2\sigma^2}), \quad \sigma > 0$$

$$E(X) = \mu$$

$$Var(X) = \sigma^2$$

A.2.11. Rozkład Weibulla

$$f(x) = \frac{a}{b}(\frac{x}{b})^{a-1}exp(-(\frac{x}{b})^a), \quad x \geqslant 0$$

$$E(X) = b\Gamma(1 + \frac{1}{a})$$

$$Var(X) = b^2(\Gamma(1 + \frac{2}{a}) - (\Gamma(1 + \frac{1}{a}))^2)$$

Bibliografia

[King] Jack L.King, *Operational Risk. Measurement and Modelling.*, Wiley Finance Series, John Wiley & Sons, LTD, 2001 Great Britain

[Kl& Pa & Wi] Stuart A. Klugman, Harry H. Panjer, Gordon E. Willmot, *Loss models. From data to decision. Third Edition*, Wiley Series in Probability and Statistics, A John Wiley & Sons, INC., Publication, 2008 New Jersey