

Proces ryzyka i Ruch Browna

Adam Wrzesiński, Joanna Kołaczek

26.06.2022

Spis treści

1 Zagadnienia wstępne	3
1.1 Generowanie wybranych rozkładów prawdopodobieństwa	3
1.1.1 Rozkład Poissona	3
1.1.2 Rozkład wykładniczy	4
1.1.3 Rozkład geometryczny	5
1.1.4 Rozkład normalny	6
1.2 Wybrane testy statystyczne zgodności	7
1.2.1 Test Kołmogorowa-Smirnova	7
1.2.2 Test Andersona-Darlinga	7
1.2.3 Symulacyjne oszacowanie p-wartości	8
1.3 Jednorodny Proces Poissona	9
1.4 Proces Ryzyka	10
1.5 Ruch Browna	11
2 Analiza wybranego klasycznego modelu procesu Ryzyka	13
2.1 Wizualizacja danych	13
2.2 Oszacowanie parametru intensywności procesu Poissona	13
2.3 Znalezienie rozkładu wypłat odszkodowań	15
2.4 Obliczenie wartości parametru narzutu	16
2.5 Sprawdzenie poprawności wyników	17
2.6 Prawdopodobieństwa bankructwa	18
2.6.1 W czasie skończonym	18
2.6.2 W czasie nieskończonym	19
3 Wybrane zależności dla trajektorii procesu Wienera	21
3.1 Średni czas wyjścia z danego przedziału	21
3.2 Prawdopodobieństwo wyjścia przez wybraną granicę	22
4 Podsumowanie	25

Wstęp

Niniejszy raport powstał na potrzeby realizacji laboratorium Symulacji Komputerowych, prowadzonych przez dr inż. Mateusza Świtałę, do wykładu prof. dr hab. inż. Marcina Magdziarza. Stanowi on niejako podsumowanie całego semestru, gdyż do jego realizacji są potrzebne zagadnienia poznane w ciągu ostatnich czterech miesięcy. Nie tylko z zakresu tego kursu lecz także ze Statystyki Stosowanej. Zanim przejdziemy do modelowania procesu Ryzyka lub procesu Browna, omówimy pokrótce właśnie te zagadnienia. Każde zadanie jest poprzedzone wstępem złożonym z wyjaśnienia owych zagadnień. Nie będziemy przy tym sprawdzać poprawności generatorów zmiennych losowych, gdyż uczyniliśmy już to w poprzedniej pracy. Przypomnimy natomiast krótko algorytmy, które przydadzą się do wykonania postawionych przed nami zadań. Omówimy pokrótce dwa testy zgodności - *Kołmogorowa-Smirnova* oraz *Andersona-Darlinga*. Przedstawimy ich statystyki testowe, algorytmy implementacji oraz pokażemy, jak za pomocą metody Monte Carlo znaleźć *p-wartość* danego testu. Następnie przejdziemy do *Jednorodnego procesu Poissona, ogólnego procesu Ryzyka* i ich własności. Zebrana w ten sposób wiedza pozwoli przeanalizować konkretny przypadek klasycznego modelu procesu Ryzyka - za pomocą 50 trajektorii zadanych na pewnym horyzoncie czasowym T i danym kroku h . Krok po kroku oszacujemy kolejne potrzebne nam wartości do osiągnięcia celu, jakim jest oszacowanie prawdopodobieństwa bankructwa danej spółki ubezpieczeniowej. W kolejnym rozdziale, będziemy badać zależność między punktem startowym procesu Wienera, a średnim czasem przekroczenia ustalonych granic, oraz prawdopodobieństwem przez którą granice nastąpi to przekroczenie. Później spróbujemy aproksymować funkcje, które odzwierciedlają uzyskane rezultaty. Do każdej metody zapiszemy algorytm, dzięki czemu w pewien sposób poniższa praca będzie niezależna i wystarczająca do odtworzenia rozwiązania zadań. Życzymy miłej lektury!

1 Zagadnienia wstępne

1.1 Generowanie wybranych rozkładów prawdopodobieństwa

W tej sekcji krótko opowiadamy o rozkładach, ich interpretacjach, a następnie wyznaczamy algorytm generowania zmiennej losowej z danego rozkładu. Notujemy, że wszystkie opierają się na *metodzie odwrotnej dystrybuanty*.

1.1.1 Rozkład Poissona

Rozkład Poissona interpretujemy jako narzędzie do wyznaczania liczby wystąpień jakiegoś zdarzenia w danym czasie. Jest to rozkład dyskretny, a prawdopodobieństwo zajścia k zdarzeń wyznaczamy ze wzoru

$$p_i = P(X = i) = e^{-\lambda} \frac{\lambda^i}{i!}, \quad i = 0, 1, 2, \dots, \quad (1.1.1)$$

gdzie $\lambda > 0$ - pewien parametr intensywności.

Ponieważ $k \in \mathbb{N}$, nie możemy bezpośrednio odwrócić (1.1.1). Wiążałoby się to z koniecznością obliczenia bardzo dużej ilości, a niestety w pewnym momencie zabraknie nam cyfr. Dlatego znajdziemy zależność pomiędzy kolejnymi prawdopodobieństwami i na jej podstawie sporządzimy algorytm:

$$p_{i+1} = e^{-\lambda} \frac{\lambda^{(i+1)}}{(i+1)!} = e^{-\lambda} \frac{\lambda^i}{i!} \cdot \frac{\lambda}{i+1} = p_i \frac{\lambda}{i+1}. \quad (1.1.2)$$

W związku z (1.1.2), wylosujemy wartość dystrybuanty U i zaczniemy nasze poszukiwania od $k = 0$. Następnie będziemy się przesuwać wzduż osi OX , aż znajdziemy k odpowiadające U .

Algorytm:

1. Wstaw $k = 0$, $p = e^{-\lambda}$, $F = p$.
2. Generuj $U \sim U(0, 1)$.
3. Dopóki $U <= F$, wstaw:

$$\begin{aligned} p &= p \cdot \frac{\lambda}{k+1} \\ F &= F + p \\ k &= k + 1. \end{aligned}$$

4. Zwróć $X = i$.

1.1.2 Rozkład wykładniczy

Rozkład wykładniczy można zdefiniować jako czas oczekiwania pomiędzy dwoma zdarzeniami. Jest to rozkład ciągły, którego dystrybuanta wyraża się wzorem (1.1.3):

$$F(x) = \begin{cases} 0 & x < 0 \\ 1 - e^{-\lambda x} & x \geq 0 \end{cases} \quad (1.1.3)$$

gdzie $\lambda > 0$ - pewien parametr częstości.

Odwóćmy teraz dystrybuantę. Jest to funkcja ciągła wyrażona za pomocą funkcji elementarnych, więc wystarczą nam podstawowe przekształcenia.

Niech $y \sim U(0, 1)$. Znajdziemy $x(y)$ za pomocą (1.1.3):

$$\begin{aligned} y &= 1 - e^{-\lambda x}, \\ 1 - y &= e^{-\lambda x}, \\ \log(1 - y) &= -\lambda x, \\ x &= -\frac{\log(1 - y)}{\lambda}. \end{aligned} \quad (1.1.4)$$

A ponieważ $y \stackrel{d}{=} 1 - y$, to (1.1.4) możemy zapisać w postaci (1.1.5).

$$x = -\frac{\log(U)}{\lambda}, \quad (1.1.5)$$

gdzie $U \sim U(0, 1)$.

Algorytm:

1. Generuj $U \sim U(0, 1)$.
2. Wstaw $X = -\frac{\log U}{\lambda}$.

1.1.3 Rozkład geometryczny

Znamy dwie parametryzacje tego rozkładu. Jedną z nich jest interpretacja ilości prób Bernouliego aż do osiągnięcia sukcesu. Drugą - liczba porażek do pierwszego momentu sukcesu. W pierwszym przypadku prawdopodobieństwo, że będziemy potrzebować k prób wynosi (1.1.6) Natomiast prawdopodobieństwo, że do momentu pierwszego sukcesu nastąpi k porażek wynosi (1.1.7). Oznaczmy te dwa przypadki jako zmienne losowe X, Y .

$$P_k = P(X = k) = (1 - p)^{k-1} p, \quad (1.1.6)$$

$$P_k = P(Y = k) = (1 - p)^k p, \quad (1.1.7)$$

gdzie $p \in (0,1)$ - prawdopodobieństwo sukcesu w pojedynczej próbie Bernouliego.

Zauważmy, że

$$Y = X - 1 \quad (1.1.8)$$

Dowód:

$$P(Y = k) = (1 - p)^k p = P(X = k + 1) = P(X - 1 = k) \Leftrightarrow Y = X - 1.$$

Tak naprawdę mamy co najmniej trzy sposoby na generowanie Y. Korzystając z (1.1.8), wystarczy wygenerować X , a potem od jego wartości odjąć jeden. Można wykorzystać też zależności pomiędzy kolejnymi prawdopodobieństwami - wtedy otrzymalibyśmy nieco zmodyfikowany algorytm generowanie zmiennej losowej poissonowskiej. My jednak pójdziemy jeszcze inną drogą. Odwrócić dystrybuantę Y . W tym wypadku nie mamy problemu z silnią. Dlatego wyznaczymy odwrotną dystrybuantę za pomocą działań na szeregach:

$$\begin{aligned} \sum_{i=0}^{j-1} p_i &< U \leq \sum_{i=0}^j p_i, \\ \sum_{i=0}^{j-1} (1-p)^i p &< U \leq \sum_{i=0}^j (1-p)^i p, \\ p \sum_{i=0}^{j-1} (1-p)^i &< U \leq p \sum_{i=0}^j (1-p)^i. \end{aligned}$$

Pod znakami sumy mamy do czynienia z ciągami geometrycznymi. Skorzystamy ze wzoru na sumę tych elementów:

$$\begin{aligned} p \frac{1 - (1-p)^j}{1 - (1-p)} &< U \leq p \frac{1 - (1-p)^{j+1}}{1 - (1-p)}, \\ 1 - (1-p)^j &< U \leq 1 - (1-p)^{j+1}, \\ (1-p)^{j+1} &\leq 1 - U < (1-p)^j. \end{aligned}$$

Teraz, nakładamy na nierówności logarytm naturalny, i po skorzystaniu z własności logarytmów oraz podstawowych przekształceniach uzyskujemy (1.1.9).

$$\begin{aligned} \log((1-p)^{j+1}) &\leq \log(1 - U) < \log((1-p)^j), \\ (j+1) \log(1-p) &\leq \log(1 - U) < (j) \log(1-p), \\ j+1 &\geq \frac{\log(1-U)}{\log(1-p)} > j, \\ j &= \left\lfloor \frac{\log(1-U)}{\log(1-p)} \right\rfloor. \end{aligned} \quad (1.1.9)$$

Znów, korzystamy z faktu, że $U \stackrel{d}{=} 1 - U$ i ostatecznie zapisujemy (1.1.10):

$$j = \left\lfloor \frac{\log(U)}{\log(1-p)} \right\rfloor. \quad (1.1.10)$$

Algorytm:

1. Generuj $U \sim U(0, 1)$.
2. Zwróć $\lfloor \frac{\log U}{1-p} \rfloor$.

1.1.4 Rozkład normalny

Symulując proces Wienera, będziemy potrzebowali generatora zmiennych losowych z rozkładem normalnego. Przypomnijmy postać gęstości tego rozkładu (1.1.11).

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (1.1.11)$$

gdzie μ i σ^2 oznaczają odpowiednio średnią oraz wariancję rozkładu.

Zastosujemy metodę biegunową. Jej zaletą jest to, że jest bardzo szybka i otrzymujemy dwie niezależne zmienne losowe z rozkładu standardowego. Przypomnijmy postać algorytmu. Założymy, że wektor losowy (V_1, V_2) ma rozkład jednostajny w kole jednostkowym $V_1 = R \cos \alpha$, $V_2 = R \sin \alpha$, $R^2 = V_1^2 + V_2^2$.

Algorytm:

1. Generuj $V_1, V_2, \dots, V_i \sim U(-1, 1)$, $V_1 \perp\!\!\!\perp V_2$.
2. Wygeneruj $R^2 = V_1^2 + V_2^2$.
3. Jeśli $R^2 > 1$ wróć do kroku pierwszego.
4. Wstaw $X = \sqrt{\frac{-2 \log R^2}{R^2}} V_1$ $Y = \sqrt{\frac{-2 \log R^2}{R^2}}$.

1.2 Wybrane testy statystyczne zgodności

1.2.1 Test Kołmogorowa-Smirnova

Test KS jest naturalny w użyciu. Jeżeli próbka pochodzi z jakiegoś rozkładu, to jej dystrybuanta empiryczna nie powinna się za bardzo różnić od teoretycznej. W związku z tym będziemy badać maksymalną odległość pomiędzy tymi dwoma funkcjami. Jeżeli będzie za duża - odrzucimy pomysł i zastanowimy się nad następnym. W związku z powyższym użyjemy statystyki

$$D_n = \max_x |F_n(x) - F(x)|, \quad (1.2.1)$$

gdzie:

- $F_n(x)$ - dystrybuanta empiryczna pewnej próby losowej X ,
- $F(x)$ - teoretyczna dystrybuanta sprawdzanego rozkładu.

Można pokazać, że dla $n \rightarrow \infty$

$$\sqrt{n}D_n \rightarrow \sup_t |B(F(t))|,$$

gdzie B to most Browna. Ponadto, jeśli F jest ciągła, to

$$\sup_t |B(F(t))| = \sup_{t \in [0,1]} |B(F(t))|.$$

Wtedy mamy do czynienia z rozkładem Kołmogorowa, niezależnym od F [1].
(1.2.1) można zapisać w postaci $\max_j\{D^+, D^-\}$, gdzie:

- $D^+ = \max_j\{\frac{j}{n} - Z_j\}$,
- $D^- = \max_j\{Z_j - \frac{j-1}{n}\}$.

Pomysł dowodu znajdziemy w [2].

Algorytm:

1. Ustal próbę losową X o długości n .
2. Posortuj rosnąco X .
3. Wyznacz zbiór $Z = \{F(X_i), i = 1, 2, \dots, n\}$ dla $X_i \in X$.
4. Oblicz D^+ oraz D^-
5. Zwróć $\max_j\{D^+, D^-\}$

1.2.2 Test Andersona-Darlinga

Statystyka Andersona-Darlinga jest szczególną postacią statystyki testowej Craméra-von Misesa [3]. Jej postać dana jest wzorem:

$$n \int_{-\infty}^{+\infty} \frac{(F_n(x) - F(x))^2}{F(x)(1 - F(x))} dF(x),$$

gdzie:

- n - długość próby,
- $F_n(x)$ - wartość dystrybuanty empirycznej w punkcie x ,

- $F(x)$ - wartość dystrybuanty testowanego rozkładu w punkcie x .

Do obliczenia tej statystyki będziemy używać (1.2.2).

$$A^2 = n - \frac{1}{n} \sum_{i=1}^n [(2i-1) \cdot \log(Z_i) + (2n+1-2i) \cdot \log(1-Z_i)], \quad (1.2.2)$$

gdzie Z jest posortowanym zbiorem $\{F(X_i), i = 1, 2, \dots, n\}$. X_i to kolejne elementy naszej próby losowej. Ponieważ dysponujemy wyjątkowo dużą próbą, nie musimy nanosić żadnych poprawek na wartość tej statystyki.

Algorytm:

1. Wyznacz próbę losową X oraz zakładany rozkład.
2. Jeśli znasz jego parametry, przejdź do kroku 3. W przeciwnym razie wyestymuj je.
3. Wyznacz posegregowany zbiór wartości dystrybuanty zakładanego rozkładu Z za pomocą próby losowej X .
4. Oblicz A^2 .

1.2.3 Symulacyjne oszacowanie p-wartości

Niestety w przypadku testów zgodności nie jesteśmy w stanie szybko znaleźć wartości krytycznych (bez zaglądania do tablic). Dodatkowo pojawia się pytanie - "Jaki poziom istotności wybrać?". Na ratunek przychodzi *p-wartość*. W naszym przypadku interesuje nas to, czy statystyka testowa nie przekoczyła pewnej wartości. W związku z tym *p-wartość* definiujemy jako

$$P_{H_0} = P(S \geq z),$$

gdzie:

- S - wartość wybranej statystyki testowej
- z - wartość statystyki testowej z danej próby

Ponieważ w dalszej części raportu będziemy operować na rozkłach z nieznanymi parametrami, ograniczymy się do wyznaczenia *p-wartości* tylko dla przypadku nieparametrycznego.

Algorytm:

1. Wyestymuj wektor nieznanych parametrów θ_1 wybranego rozkładu.
2. Wybierz test zgodności.
3. Ustal długość badanej próby n , ilość symulacji L oraz $s = 0$.
4. Wyznacz statystykę z z pierwotnej próby.
5. Wygeneruj próbę losową X o liczności n pochodzącą z wybranego rozkładu.
6. Wyestymuj wektor nieznanych parametrów θ_2 próby losowej.
7. Wyznacz odpowiednią statystykę testową S za pomocą X .
8. Sprawdź, czy zachodzi $S \geq z$. Jeśli tak, zwiększ s o jeden.
9. Powtarzaj kroki 5 – 8 L razy.
10. Zwróć $P = \frac{s}{L}$.

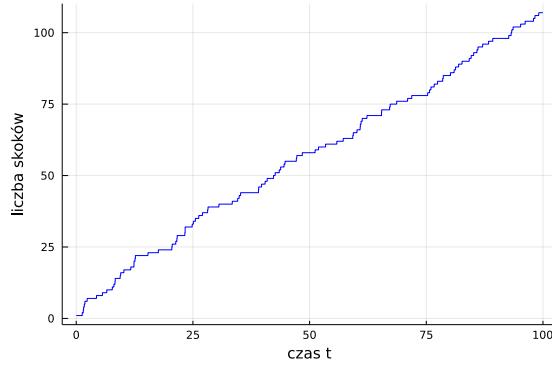
1.3 Jednorodny Proces Poissona

Jednorodnym procesem Poissona z parametrem intensywności $\lambda > 0$ nazywamy proces liczący $\{N(t), t \geq 0\}$, który spełnia następujące założenia:

- $N(0) = 0$
- $N(t)$ ma niezależne przyrosty.
- $N(t)$ ma stacjonarne przyrosty.
- $N(t) \sim Poiss(\lambda t)$.

Niezależność przyrostów oznacza, że ilość skoków na zadanym przedziale czasowym nie zależy od tego co się działo ani wcześniej ani później. Zapiszmy to w sposób matematyczny:

$\forall (0 \leq t_1 < t_2 < \dots < t_n)$ zmienne losowe $N(t_2) - N(t_1)$, $N(t_3) - N(t_2)$, ..., $N(t_n) - N(t_{n-1})$ są niezależne. Stacjonarność przyrostów mówi natomiast o tym, że $N(t+s) - N(t) \stackrel{d}{=} N(s)$. Oznacza to, że danym przedziale czasowym powinno wystąpić średnio tyle samo skoków, co na innym dowolnie wybranym przedziale o tej samej długości.



Rysunek 1.1: Przykładowa trajektoria procesu Poissona

Aby wysymulować pojedynczą trajektorię procesu Poissona, wykorzystamy fakt, że czasy oczekiwania na kolejny skok pochodzą z rozkładu wykładniczego o parametrze λ . (1.3.1) Jeśli znamy wektor kolejnych skoków $N(t)$, to wiemy też, jak wygląda cała trajektoria $N(t)$. Przykład możemy zobaczyć na rys. 1.1.

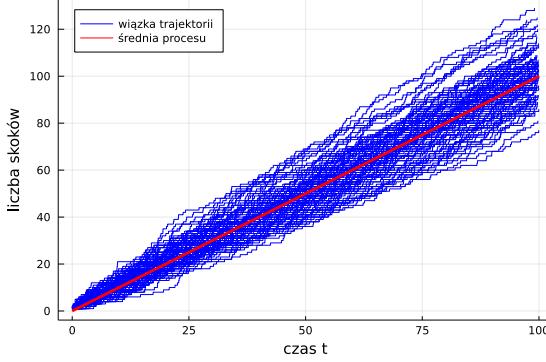
Niech T będzie oznaczać czas oczekiwania na kolejny skok w Jednorodnym Procesie Poissona z parametrem intensywności λ . Pokażemy, że $T \sim Exp(\lambda)$.

$$P(T > \tau) = P(N(\tau) = 0) = e^{-\lambda\tau} \Rightarrow T \sim Exp(\lambda). \quad (1.3.1)$$

Algorytm:

1. Wstaw $I = 0$, $t = 0$.
2. Generuj $U \sim U(0,1)$.
3. Wstaw $t = t - \frac{\log U}{\lambda}$. Jeśli $t > T$ STOP. W przeciwnym razie wstaw $I = I + 1$, $S_I = t$.
4. Wróć do 2.

Sprawdźmy poprawność tego algorytmu. Narysujmy na wykresie (rys. 1.2) 100 trajektorii i nanieśmy średnią wartość procesu. Korzystając z definicji Jednorodnego Procesu Poissona wiemy, że w danym punkcie t , średnio powinniśmy znaleźć się w punkcie λt . Rzeczywiście, trajektorie oplatają wartość średnią. Na tej podstawie stwierdzamy, że generator trajektorii jest poprawny.



Rysunek 1.2: Sprawdzenie poprawności algorytmu symulacji JPP

1.4 Proces Ryzyka

Ogólny proces Ryzyka

Procesem Ryzyka nazywamy proces stochastyczny opisujący kapitał firmy ubezpieczeniowej. Proces ten zadany jest wzorem

$$R(t) = u + c(t) - \sum_{i=1}^{N(t)} X_i,$$

gdzie:

- $u > 0$ - kapitał początkowy,
- $c(t)$ - premia (przychody ze sprzedaży polis),
- $N(t)$ - proces liczący straty,
- X_i - zmienna losowa *i.i.d.*, $X_i > 0$, reprezentatywna wartość wypłaconych odszkodowań.

Oczywiście $c(t)$ powinno być dobrane w taki sposób, aby firma nie tylko nie zbankrutowała, ale także przynosiła zyski. W naszych rozważaniach przyjmujemy, że firma nie ma możliwości wzięcia kredytu. To znaczy moment, w którym kapitał firmy spada poniżej zera nazywamy **momentem ruiny**. Matematycznie:

$$\tau(u) \stackrel{\text{def}}{=} \inf\{t \geq 0 : R(t) < 0\}.$$

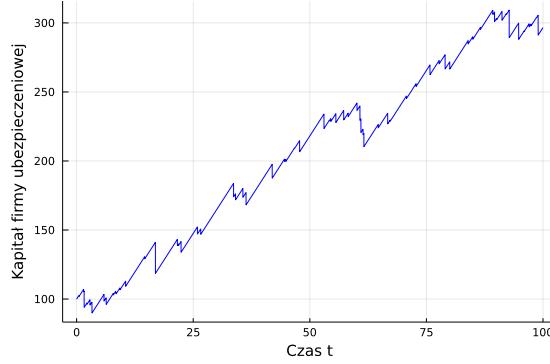
Zakładamy, że trajektorie procesu są lewostronne ciągłe. Zapiszmy algorytm pozwalając wykonać symulację $R(t)$ na zadanym przedziale czasowym $[0, T]$, $T > 0$.

Algorytm:

1. Generuj $N(t)$ na $[0, T]$.
2. Generuj X_1, X_2, \dots, X_{NT} .
3. Wstaw $R(t) = u + c(t) - \sum_{i=1}^{N(t)} X_i$.

Klasyczny Proces Ryzyka

W **Klasycznym Procesie Ryzyka** premia $c(t)$ jest funkcją liniową postaci ct , gdzie c jest pewną stałą, natomiast $N(t)$ to Jednorodny Proces Poissona z intensywnością $\lambda > 0$. Aby wyznaczyć c , najpierw znajdziemy średnią stratę kapitału firmy ubezpieczeniowej:



Rysunek 1.3: Przykładowa trajektoria procesu Ryzyka

$$\mathbb{E}\left(\sum_{i=1}^{N(t)} X_i\right) = \mathbb{E}\left(\mathbb{E}\left(\sum_{i=1}^{N(t)} X_i | N(t)\right)\right) = \mathbb{E}(\mu N(t)) = \mu \lambda t,$$

gdzie μ - średnia wypłata odszkodowania ($\mathbb{E} X_1$).

Więc już wiemy, że $c(t) > \mu \lambda t$, ponieważ zakład chce zarobić. Aby dokładniej opisać naszą funkcję premii, wprowadźmy pojęcie *narzutu* $\theta > 0$, które będziemy interpretować jako tempo sprzedaży polis ubezpieczeniowych. Zapiszmy ostatecznie:

$$c(t) = (1 + \theta)\mu \lambda t. \quad (1.4.1)$$

Możemy już przedstawić pełną postać klasycznego procesu $R(t)$:

$$R(t) = u + (1 + \theta)\mu \lambda t - \sum_{i=1}^{N(t)} X_i. \quad (1.4.2)$$

Przykładową trajektorię takiego procesu można zobaczyć na rys. 1.3.

1.5 Ruch Browna

Ruchem Browna (procesem Wienera) nazywamy proces stochastyczny $B = (B_t)_{t \geq 0}$, taki że:

- $B_0 = 0$.
- B ma niezależne przyrosty.
- $B_t - B_s \sim N(0, t - s)$, dla $0 \leq s < t$.
- B ma trajektorie ciągłe z prawdopodobieństwem 1.

W tej pracy ograniczymy się do jego zmodyfikowanej wersji:

$$B_t^x = B_t^0 + x,$$

Gdzie B_t^0 to klasyczny ruch Browna. Oznacza to tyle, że proces B_t^x rozpoczyna się w punkcie x . Chcąc uzyskać jego numeryczne przybliżenie przy pomocy symulacji komputerowej, wykorzystamy następujący algorytm:

Algorytm(0)

1. $B_0^x = x$

$$2. \quad B_{i+1}^x = B_i^x + dt^{\frac{1}{2}}\psi_i$$

gdzie $\psi_i \sim N(0,1)$, i.i.d, natomiast dt to najmniejszy krok czasowy. Algorytm dla wersji procesu startującego w zerze jest nieomal identyczny.

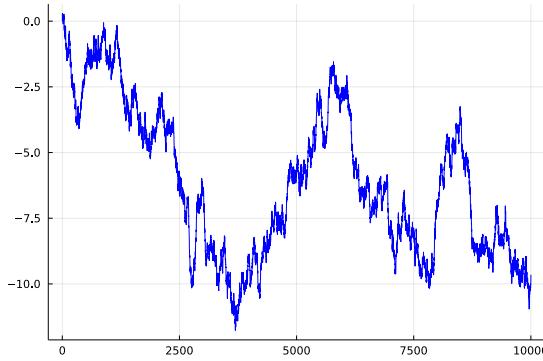
Ponieważ ruch Browna jest procesem gausowskim, jest on wyznaczony jednoznacznie przez wartość średnią, oraz autokowariancje:

Funkcja autokowariancji

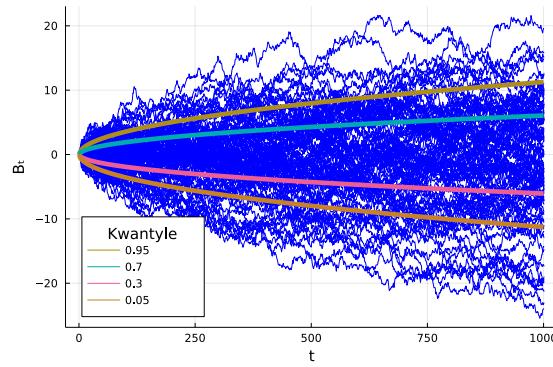
$$\begin{aligned} Cov(B_t, B_s) &= \mathbb{E}(B_s B_t) - \mathbb{E}B_s \mathbb{E}B_t = \\ &= \mathbb{E}(B_s(B_s B_t)) + \mathbb{E}(B_s^2) \\ &= \mathbb{E}B_s \mathbb{E}(B_s B_t) + \mathbb{E}B_s^2 = s \end{aligned}$$

Ogólnie

$$Cov(B_t, B_s) = \min\{s, t\}$$



Rysunek 1.4: Przykładowa trajektoria Ruchu Browna



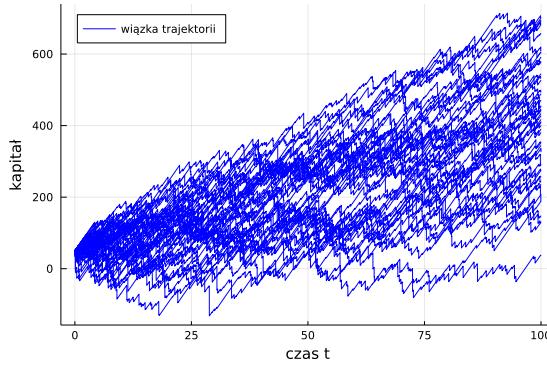
Rysunek 1.5: Teoretyczne kwantyle na tle stu trajektorii

2 Analiza wybranego klasycznego modelu procesu Ryzyka

2.1 Wizualizacja danych

Dysponujemy 50-cioma trajektoriami z pewnego klasycznego procesu Ryzyka $R(t)$ wygenerowanych na odcinku $[0, T]$, $T = 100$ z krokiem $h = 0.01$ i kapitałem początkowym $u = 50$. Naszym pierwszym celem będzie znalezienie jawnnej postaci $R(t)$.

Zacznijmy od zwizualizowania danych trajektorii na wykresie 2.1. Niestety, z samego wykresu nie jesteśmy zbyt wiele wynioskować. Z trajektorii odstających upewniamy się, że rzeczywiście przyrost kapitału jest liniowy. Na ten moment nie wiemy nic o rozkładzie wypłat. Widzimy za to kilka odstających trajektorii, które mogą zaniżać średnią wartość procesu.



Rysunek 2.1: Przykładowa trajektoria procesu Ryzyka

2.2 Oszacowanie parametru intensywności procesu Poissona

Pomysł: Jeśli $N(t)$ to jednorodny proces Poissona z parametrem intensywności λ , to czasy oczekiwania na kolejny skok są z rozkładu wykładniczego z tym samym parametrem, co już pokazaliśmy wcześniej (1.3.1). W związku z tym należy znaleźć momenty tych skoków, a następnie z nich wyliczyć czasy oczekiwania pomiędzy kolejnymi skokami. Otrzymamy prostą próbę losową z rozkładu wykładniczego z parametrem λ . Następnie, za pomocą metody momentów oraz największej wiarogodności wyestymujemy porządkany parametr.

Wektor momentów skoków

Trajektoria to pewien wektor przedstawiający kapitał firmy ubezpieczeniowej w pewnym czasie τ . Oczywiście do momentu wypłaty odszkodowania musi być ona rosnąca. Tym faktem posłużymy się znajdując momenty skoków.

Zapiszmy algorytm, który zapisze czasy τ^* , w których nastąpił jakikolwiek spadek w kapitale. Zakładamy przy tym, że trajektoria jest lewostronnie ciągła.

1. Ustal $\tau = 0$, $h = 0.01$, $I = 1$, trajektorię ryzyka R oraz pusty wektor V .
2. Zwięksź czas τ o krok h , zwięksź I o jeden.
3. Sprawdź, czy firma odnotowała spadek ($R[I] < R[I - 1]$). Jeśli tak, wstaw τ do wektora V . W przeciwnym przypadku kontynuuj.
4. Powtarzaj 2-3, dopóki $I < \#R$.
5. Zwróć V .

Wektor czasów oczekiwania

Mając do dyspozycji momenty skoków, w bardzo łatwy sposób jesteśmy w stanie znaleźć czasy oczekiwania. Weźmiemy po prostu różnice kolejnych momentów. Powtórzymy cały proces dla każdej z trajektorii, a następnie złączymy wszystkie wektory w jeden. Możemy tak zrobić, gdyż każda symulacja jest od siebie niezależna. Zwiększymy przy tym dokładność estymatorów.

Algorytm:

1. Znajdź wektor momentów skoków V .
2. Ustal licznosć k wektora V .
3. Stwórz wektor czasów oczekiwania T_j o długości $k - 1$.
4. Ustal $T[1] = V[1]$, $I = 2$.
5. Wstaw $T[I] = V[I] - V[I - 1]$.
6. Zwiększ I o jeden.
7. Powtórz kroki 5-6 $k - 2$ razy.
8. Powtórz kroki 1-7 dla każdej z trajektorii.
9. Wstaw $T = \bigcup T_j$.

W naszym przypadku $j = 1, 2, \dots, 50$. Dla formalności przedstawmy otrzymany wektor na histogramie częstotliwości. Upewniamy się w ten sposób, że nie popełniliśmy błędu. Przypomnijmy, że nasz wektor czasów oczekiwania jest dosyć dużą próbą losową z rozkładu wykładniczego z parametrem λ . Do estymacji tego parametru posłużymy się metodami największej wiarogodności oraz momentów.

Metoda Największej wiarogodności polega na znalezieniu takiego parametru, który maksymalizuje funkcję wiarogodności.

$$\begin{aligned} f(x) &= \theta e^{-\theta x}, \\ \mathcal{L}(x_1, \dots, x_n, \theta) &= \prod_{i=1}^n \theta e^{-\theta x_i}, \\ \ln(\mathcal{L}(x_1, \dots, x_n, \theta)) &= \sum_{i=1}^n \ln(\theta e^{-\theta x_i}) \\ &= \sum_{i=1}^n (\ln(\theta) - \theta x_i) = n \ln(\theta) - \theta \sum_{i=1}^n x_i, \end{aligned}$$

Szukamy teraz maksimum tej funkcji:

$$\begin{aligned} \frac{d(\ln(\mathcal{L}(x_1, \dots, x_n, \theta)))}{d\theta} &= \frac{n}{\theta} - \sum_{i=1}^n x_i = 0, \\ \Leftrightarrow \theta &= \frac{n}{\sum_{i=1}^{\infty} X_i}. \end{aligned}$$

Metoda momentów polega na analitycznym obliczeniu momentów zwykłych, a potem używaniu ich odpowiedników dla populacji. Wyznaczamy tyle momentów, ile nieznanych jest parametrów. W naszym przypadku

jest to jeden parametr.

$$\begin{aligned}
 X &\sim \text{Exp}(\theta) \\
 EX &= \int_0^\infty x\theta e^{-\theta x} = \frac{1}{\theta} \int_0^\infty ye^{-y} dy \\
 &= \frac{1}{\theta} [-e^{-y} - ye^{-1}]_0^\infty = \frac{1}{\theta} = \frac{1}{n} \sum_{i=1}^{\infty} X_i \\
 \Rightarrow \theta &= \frac{n}{\sum_{i=1}^{\infty} X_i}
 \end{aligned}$$

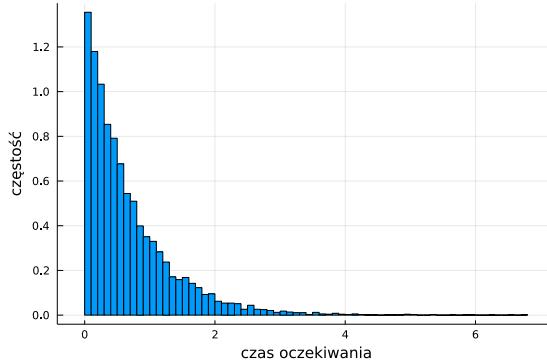
Okazuje się, że te dwa estymatory są sobie równe. Z teorii statystyki wiemy również, że są asymptotycznie nieobciążone. Zapiszmy ostatecznie:

$$\bar{\lambda} = \frac{n}{\sum_{i=1}^{\infty} X_i}. \quad (2.2.1)$$

Po podstawieniu do wzoru (2.2.1) otrzymujemy, że nasze

$$\lambda = 1.4681753373312858.$$

Celowo nie zaokrąglamy wartości, by jak najdokładniej odzwierciedlić analizowany proces.



Rysunek 2.2: Histogram częstości czasów oczekiwania na kolejny skok

2.3 Znalezienie rozkładu wypłat odszkodowań

Pomysł: Postąpimy podobnie jak w przypadku szacowania parametru intensywności procesu Poissona. W tym celu przeiterujemy się po całej trajektorii i ponownie znajdziemy miejsca, w których nastąpił skok. Tym razem zapiszemy jednak różnicę w wartości kapitału firmy, a nie momenty skoków.

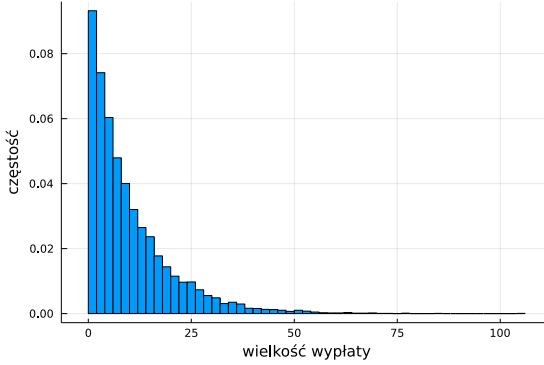
Algorytm:

1. Ustal $I = 1$, trajektorię ryzyka R oraz pusty wektor X_j .
2. Zwięksź I o jeden.
3. Wyznacz $\Delta R = R[I - 1] - R[I]$.
4. Jeśli $\Delta R >= 0$, wstaw wynik do wektora V .
5. Powtarzaj 2-3, dopóki $I < \#R$.

6. Zwróć X .
7. Powtórz kroki 1-6 dla każdej z trajektorii.
8. Wstaw $X = \bigcup X_j$.

Ponownie otrzymujemy próbę losową. Tym razem nie znamy jednak jej rozkładu. Rozkład częstości pokazany na histogramie bardzo przypomina rozkład wykładniczy. W związku z tym użyjemy testów statystycznych omówionych w poprzednim dziale, by sprawdzić, czy nasza próba może pochodzić z rozkładu właśnie wykładniczego. Wyznaczmy p-wartość testów. Jeśli przekroczą one wartości 0.05 w obu przypadkach, przyjmujemy hipotezę zerową na poziomie istotności 5%. Wyniki gromadzimy w tabeli (1) i odczytujemy, że rzeczywiście dane mogą pochodzić z rozkładu wykładniczego. Po porównaniu dystrybuant teoretycznej i empirycznej nie mamy już wątpliwości. Dysponujemy próbą losową z rozkładu wykładniczego! Pozostaje wyestymować jej parametr ρ . Podstawiając do wzoru (2.2.1) otrzymujemy, że

$$\rho = 0.09901122812369685.$$



Rysunek 2.3: Histogram częstości wartości wyplat odszkodowań

Statystyka	Wartość	P-wartość	Wartość krytyczna
D_n	0.915	0.1614	1.094
A^2	1.024	0.109	1.321

Tabela 1: Wyniki testów zgodności

2.4 Obliczenie wartości parametru narzutu

Premię wyznaczmy sposobem bardziej uniwersalnym, tzn. działającym nie tylko w przypadku klasycznego procesu ryzyka.

Najpierw spójrzmy na wzór (1.4.2). Po krótkim zastanowieniu dochodzimy do wniosku, że średnio nasz proces powinien wzrastać o $\theta\lambda^{\frac{1}{\rho}}$. Dlaczego?

Otoż spójrzmy jeszcze na (1.4.1) i na jego wyprowadzenie. Skoro człon $\mu\lambda$ odpowiada za minimalizowanie strat, to pozostała część musi odpowiadać za wzrost kapitału!

Wyznaczmy więc empiryczną średnią wartość procesu z naszego zestawu trajektorii, a następnie wyznaczmy metodą regresji liniowej, korzystając z `numpy.polyfit` [4], prostą, której współczynnik kierunkowy powinien być równy naszej teoretycznej średniej procesu. Znając ρ oraz λ , podstawimy do wzoru i otrzymamy nasz pożądany parametr.

Algorytm:

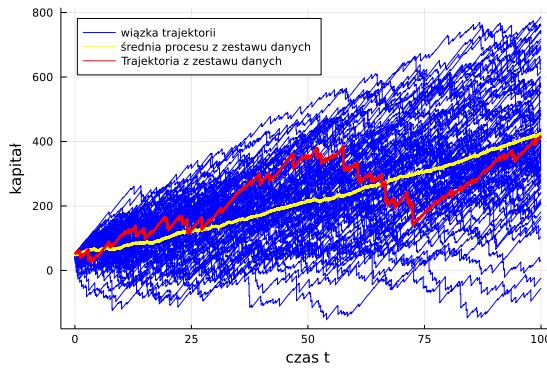
1. Wyznacz empiryczną średnią procesu z zestawu danych M .
2. Dopasuj prostą l do średniej M .
3. Zapisz jej współczynnik kierunkowy a .
4. Wstaw $\theta = \frac{a\rho}{\lambda}$.

Ostatecznie otrzymujemy, że

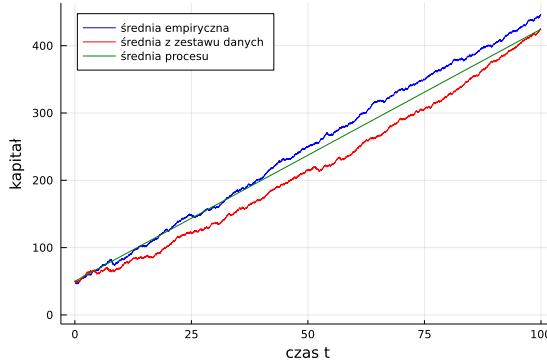
$$\theta = 0.2526239274374162.$$

2.5 Sprawdzenie poprawności wyników

Mając do dyspozycji jawną postać naszego procesu ryzyka możemy generować nowe trajektorie. Na rysunku 2.4 zamieszczamy 100 nowych przebiegów ustalonego procesu Ryzyka. Nałożymy na nim jedną losowo wybraną trajektorię z zestawu danych. Zaznaczmy również średnią empiryczną z zestawu danych. Na rysunku 2.5 porównujemy średnią empiryczną z 50-ciu nowych trajektorii z teoretyczną, na tle empirycznej z zestawu danych. Widzimy, że dla malej próby, tak jak w zestawie danych, mogą nastąpić mniej lub bardziej poważne odchylenia. Możemy otrzymać nawet bardziej ekstremalne przypadki. Wystarczy dostatecznie wiele razy generować małą wiązkę trajektorii. W końcu natrafimy na osobliwy przypadek.



Rysunek 2.4: Przykładowa wiązka stu trajektorii procesu ryzyka dla naszych parametrów



Rysunek 2.5: Porównanie średnich empirycznych

2.6 Prawdopodobieństwa bankructwa

2.6.1 W czasie skończonym

Prawdopodobieństwo ruiny w skończonym czasie definiujemy jako

$$\psi(u,T) = P(\tau(u) < T),$$

gdzie:

- $u > 0$ - kapitał początkowy,
- $T > 0$ - badany okres.

Najczęściej $\psi(u,T)$ musimy wyznaczać symulacyjnie. Jednak w przypadku, gdy zmienne losowe odpowiadające za wypłaty są z rozkładu wykładniczego, znamy jego teoretyczną wartość. Zapiszmy jego postać dla przypadku, gdy $c = 1$, $\rho = 1$.

Tutaj c - premia, ρ - parametr rozkładu wykładniczego.

Mamy wtedy:

$$\psi(u,T) = \lambda \exp(-(1-\lambda)u) - \frac{1}{\pi} \int_0^\pi \frac{f_1(x)f_2(x)}{f_3(x)} dx, \quad (2.6.1)$$

gdzie:

$$\begin{aligned} f_1(x) &= \lambda \exp(2\sqrt{\lambda}T \cos x - (1+\lambda)T + u(\sqrt{\lambda} \cos x - 1)), \\ f_2(x) &= \cos(u\sqrt{\lambda} \sin x) - \cos(u\sqrt{\lambda} \sin x + 2x), \\ f_3(x) &= 1 + \lambda - 2\sqrt{\lambda} \cos x. \end{aligned}$$

Dla $\rho \neq 1$ korzystamy z faktu, że prawdopodobieństwo ruiny

$$\psi_{\lambda,\rho}(u,T) = \psi_{\frac{\lambda}{\rho},1}(\rho u, \rho T).$$

Dla $c \neq 1$ korzystamy z faktu, że

$$\psi_{\lambda,c}(u,T) = \psi_{\frac{\lambda}{c},1}(u, cT).$$

W naszym przypadku zarówno ρ jak i c są różne od jedynki. W związku z tym musimy połączyć te dwa przypadki (2.6.2).

$$\psi(u,T) = \psi_{\frac{\lambda}{\rho c},1}(\rho u, \rho c T). \quad (2.6.2)$$

Po podstawieniu do (1.4.2) otrzymujemy, że:

$$\psi(u,T) = \psi_{\frac{1}{1+\theta},1}(\rho u, (1+\theta)\lambda T). \quad (2.6.3)$$

Podstawiamy nasze parametry do (2.6.3) i wyniki wstawiamy do tabeli 3.

Trzeba jednak przyznać, że korzystanie z (2.6.1) może być mało praktyczne, a zwykle - po prostu niemożliwe. Zamiast tego możemy wygenerować N trajektorii i zliczać te, które spadły poniżej zera. Wydaje się to o wiele prostsze.

Algorytm

1. Generuj N trajektorii $R^{(1)}(t), \dots, R^{(N)}(t)$ procesu ryzyka na $[0, T]$.
2. Wyznacz $n = \#\{i \in \{1, \dots, N\} : \min_{T \in [0, T]} R^{(i)}(t) < 0\}$.
3. Wstaw $\psi(u,T) = \frac{n}{N}$.

Nasze wyniki umieszczamy w tabeli 2

2.6.2 W czasie nieskończonym

Prawdopodobieństwo ruiny w nieskończonym czasie definiujemy jako

$$\psi(u) = P(\tau(u) < \infty),$$

gdzie $u > 0$ - kapitał początkowy.

Wyznaczenie tego prawdopodobieństwa może wydawać się trudne. Mamy bowiem do czynienia z czasem nieskończonym, a to.. dosyć długo. Na szczęście matematykom udało się rozwiązać ten problem. Sukces ten przypisuje się dwóm wybitnym jednostkom - F. Pollaczkowi oraz A. Chinczynowi. Ich wynik zapisujemy w (2.6.4).

$$\psi(u) = \frac{\theta}{1+\theta} \sum_{n=0}^{\infty} \left(\frac{1}{1+\theta} \right)^n \cdot B_n(u), \quad (2.6.4)$$

gdzie $B_n(u) = P(Y_1 + \dots + Y_n > u)$ oraz Y_i to zmienne losowe *i.i.d.* o gęstości $f(x) = \frac{1-F_{X_i}(x)}{\mathbb{E} X_i}$.

W naszym przypadku wielkości wypłat są z rozkładu wykładniczego. W związku z tym postać $\psi(u)$ powinna być znana. Policzmy $\psi(u)$. Mamy

$$X_i \sim Exp(\rho) \Rightarrow f(x) = \frac{1 - (1 - e^{-\rho x})}{\frac{1}{\rho}} = \rho e^{-\rho x} \sim Exp(\rho).$$

Skoro Y_i jest z rozkładu wykładniczego, to suma wszystkich Y_i będzie z rozkładu Erlanga (gamma z parametrami n, ρ). Ten fakt udowodniliśmy w poprzednim raporcie. Podstawiając do (2.6.4) i wstawiając wzór na gęstość rozkładu Erlanga:

$$\begin{aligned} \psi(u) &= \frac{\theta}{1+\theta} \sum_{n=1}^{\infty} \left(\frac{1}{1+\theta} \right)^n \int_u^{\infty} \frac{(\rho x)^{n-1}}{(n-1)!} \rho e^{-\rho x} dx \\ &= \frac{\theta \rho}{(1+\theta)^2} \int_u^{\infty} \rho e^{-\rho x} \sum_{n=1}^{\infty} \left(\frac{1}{1+\theta} \right)^{n-1} \frac{(\rho x)^{n-1}}{(n-1)!} dx \end{aligned}$$

Jeśli dobrze się przyjrzymy, to zauważymy, że suma pod całką jest tak na prawdę rozwinięciem Maclaurina funkcji $e^{\frac{\rho x}{1+\theta}}$. Podstawiając ją, otrzymujemy łatwą do przeliczenia całkę:

$$\begin{aligned} &= \frac{\theta \rho}{(1+\theta)^2} \int_u^{\infty} e^{-\rho x} e^{\frac{\rho x}{1+\theta}} dx = \frac{\theta \rho}{(1+\theta)^2} \int_u^{\infty} e^{-\rho(x - \frac{x}{1+\theta})} dx \\ &= \frac{\theta \rho}{(1+\theta)^2} \int_u^{\infty} e^{-\rho(x - \frac{\theta}{1+\theta})} dx = \frac{1}{1+\theta} e^{-\rho u \frac{\theta}{1+\theta}}. \end{aligned}$$

Więc ostatecznie zapisujemy (2.6.5):

$$\psi(u) = \frac{1}{1+\theta} e^{-\rho u \frac{\theta}{1+\theta}}. \quad (2.6.5)$$

Wstawiając $u = 50, T = 100, 200$, otrzymujemy teoretyczne prawdopodobieństwo ruiny w czasie nieskończonym. Wyniki zapisujemy do tabeli 3.

Możemy oszacować prawdopodobieństwo $\psi(u)$ metodą Monte Carlo.
 Najpierw pokażemy, że (2.6.6)

$$\psi(u) = P(Y_1 + \dots + Y_K > u), \quad \text{gdzie } K \sim Geom\left(\frac{\theta}{1+\theta}\right), \quad K \perp\!\!\!\perp Y_i. \quad (2.6.6)$$

Dowód:

$$\begin{aligned} P(Y_1 + \dots + Y_K > u) &= \sum_{n=0}^{\infty} P(Y_1 + \dots + Y_n > u) P(K = n) \\ &= \sum_{n=0}^{\infty} P(Y_1 + \dots + Y_n > u) \left(\frac{\theta}{1+\theta}\right) \left(\frac{1}{1+\theta}\right)^n = \psi(u) \end{aligned}$$

Teraz mamy już wszystkie informacje potrzebne do zbudowania algorytmu. Będziemy generować zmienne losowe z rozkładu podanym w (2.6.4), a następnie sprawdzać, czy ich suma jest większa od kapitału początkowego. Wyniki zapiszemy do tabeli 2.

Algorytm

1. Ustal N symulacji.
2. Generuj $K \sim Geom\left(\frac{\theta}{1+\theta}\right)$.
3. Generuj Y_1, \dots, Y_K - i.i.d o gęstości $f(x) = \frac{1-F_{X_i}(x)}{\mu}$.
4. Jeśli $Y_1 + \dots + Y_K > u$, wstaw $Z(i) = 1$, w przeciwnym wypadku $Z(i) = 0$
5. Powtórz kroki 2-4 N -krotnie.
6. Wstaw $\psi(u) = \frac{Z(1)+\dots+Z(N)}{N}$.

Czas	Prawdopodobieństwo
1	0.29
2	0.2926
∞	0.2943

Tabela 2: Wyniki otrzymane metodą Monte Carlo

Czas	Prawdopodobieństwo
1	0.2919
2	0.294
∞	0.2941

Tabela 3: Wyniki teoretyczne

Wnioski

A więc osiągnęliśmy nasz cel. Z pozornie zbyt małej ilości trajektorii, które jak się okazało były znacznie odstające od średniej, uzyskaliśmy pełny model opisujący potencjalną przyszłość pewnej firmy ubezpieczeniowej. Trzeba przyznać, że prawdopodobieństwo bankructwa jest dosyć spore. Zwróćmy uwagę na fakt, że jeżeli zakład nie upadnie w najbliższym czasie, to już prawdopodobnie tego nie zrobi (tab. 3). Wynika to z prostej przyczyny - Firma startuje z dosyć niskim budżetem. Jeżeli prezesi będą mieli pecha, to klienci będą domagać się zbyt wielu odszkodowań za pewne uszczerbki. Wtedy niestety zakład czeka ruina. Jeśli uda się jednak przetrwać początkowy okres, to po pewnym czasie firma rozwinie się za bardzo, żeby zbankrutować.

3 Wybrane zależności dla trajektorii procesu Wienera

3.1 Średni czas wyjścia z danego przedziału

Będziemy badać średni czas wyjścia z przedziału $[a,b]$, dla $x \in [a,b]$. Oznaczmy go jako

$$\mathbb{E}\tau_x, \text{ gdzie } \tau_x := \inf\{t \geq 0 : B_t^x \notin (a,b)\}.$$

Ponieważ na komputerze nie jest możliwe wykonanie symulacji procesu Wienera w czasie ciągłym, ustalamy najmniejszy krok czasowy dt , potrzebny do zastosowania metody numerycznej. Aby móc jak najlepiej oszacować średni czas wyjścia, powtarzamy n razy symulację dla danego x .

Algorytm:

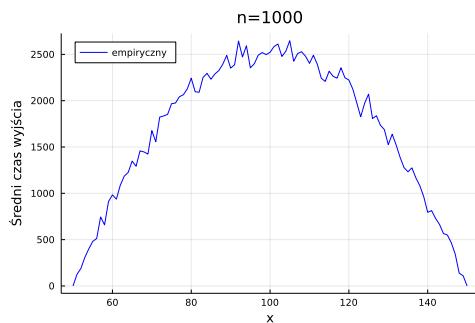
1. Ustal punkt startowy x , najmniejszy krok czasowy dt , liczbę powtórzeń symulacji n oraz granice przedziału a i b .
2. Jeżeli $x = a$ lub $x = b$ zwróć 0.
3. Ustal $T = 0$.
4. Generuj trajektorię procesu Wienera B_t^x dopóki $B_t^x \leq a$ lub $B_t^x \geq b$.
5. Zwięksź T o ilość kroków czasowych dt , które minęły do tego momentu.
6. Powtórz kroki 4-5 n razy.
7. Zwróć $\frac{T}{n}$.

Chcąc zbadać zależność średniego czasu wyjścia od odległości punktu startowego x do granic przedziału $[a,b]$ musimy ustalić dx , czyli różnicę między kolejnymi początkowymi $x \in [a,b]$.

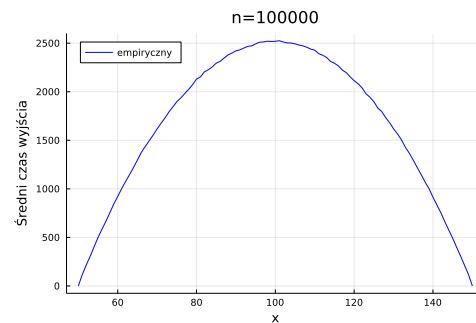
Algorytm:

1. Ustal różnicę dx , najmniejszy krok czasowy dt , liczbę powtórzeń symulacji n , oraz granice przedziału a i b .
2. Stwórz pusty wektor y .
3. Stwórz wektor v , początkowych x -ów należących do przedziału $[a,b]$, tak aby $v[i+1] - v[i] = dx$.
4. Dla każdego elementu z wektora v oblicz średni czas wyjścia i wstaw otrzymany wynik do wektora y .
5. Zwróć y

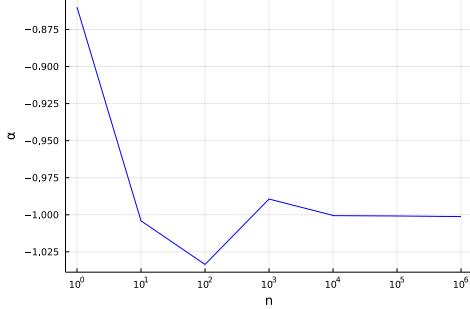
Mając powyższe informacje, możemy przeprowadzić symulację. Rozważmy przypadki dla $n = 10^3$ i $n = 10^5$, $dt = 0.1$, $dx = 1$, $a = 50$, $b = 150$. Otrzymujemy wyniki widoczne na rys. 3.1 i rys. 3.2



Rysunek 3.1



Rysunek 3.2



Rysunek 3.3

Spróbujemy aproksymować funkcję, która opisuje otrzymywane wyniki. Po kształcie wykresu przypuszczamy, że może to być wielomian co najmniej stopnia II na przedziale $[a,b]$. Wiemy również, że jego zera znajdują się w punktach a oraz b , ponieważ czas wyjścia w przypadku gdy $x = a$ lub $x = b$, będzie wynosił 0. Założymy więc, że jest to funkcja kwadratowa, po zapisaniu w postaci kwadratowej wygląda następująco:

$$f(x) = \alpha(x - a)(x - b),$$

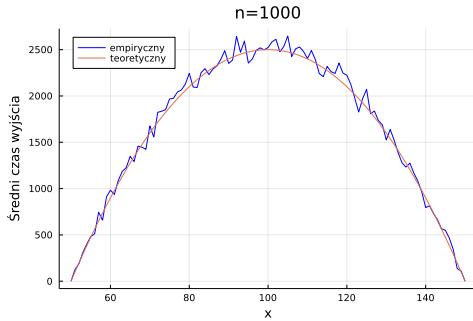
gdzie α jest szukanym współczynnikiem. W rozważanym przez nas przypadku:

$$f(x) = \alpha(x - 50)(x - 150).$$

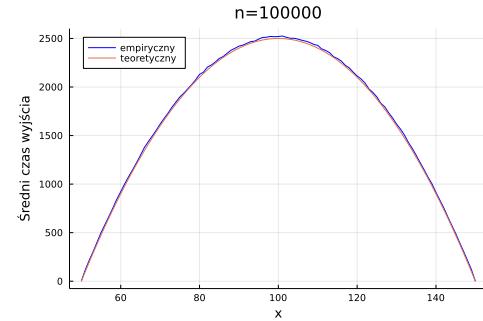
Aby znaleźć współczynnik α użyjemy modułu `numpy.polyfit` [4]. Dla przypadku z $n = 10^3$ otrzymujemy $\alpha = 1.012$, natomiast dla $n = 10^5$, $\alpha = -1.001$. Na wykresie 3.3 widzimy jak współczynnik α zmienia się wraz ze wzrostem n . Możemy przypuścić, że $\lim_{n \rightarrow \infty} c = -1$. Podobnie średnia ze stu powtórzeń algorytmu(2) dla $n = 1000$ wraz z aproksymacją `numpy.polyfit` wskazała, że $\alpha = 1$. Ostatecznie otrzymujemy:

$$f(x) = -(x - 50)(x - 150).$$

Zestawienie funkcji empirycznej z teoretyczną znajduje się na rys. 3.4 oraz rys. 3.5.



Rysunek 3.4



Rysunek 3.5

3.2 Prawdopodobieństwo wyjścia przez wybraną granicę

Przejdziemy teraz do szacowania, że wyjście nastąpiło przez granicę b , czyli:

$$P(B_{\tau_x}^x = b).$$

Jako, że symulacja działa w czasie dyskretnym, do określenia prawdopodobieństwa użyjemy wzoru:

$$P(B_{\tau_x}^x \geq b).$$

Postępujemy podobnie jak w przypadku szukania średniego czasu wyjścia.

Algorytm:

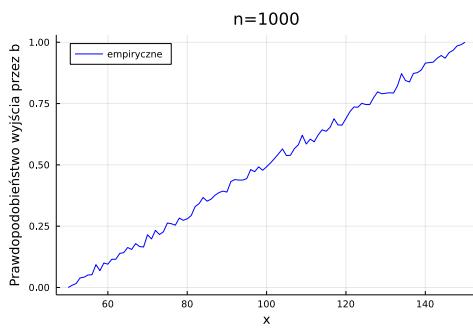
1. Ustal punkt startowy x , najmniejszy krok czasowy dt , liczbę powtórzeń symulacji n oraz granice przedziału a i b .
2. Jeżeli $x = a$ zwróć 0.
3. Jeżeli $x = b$ zwróć 1.
4. Ustal $k = 0$
5. Generuj trajektorię procesu Wienera B_t^x dopóki $B_t^x \leq a$ lub $B_t^x \geq b$.
6. Jeśli $w \geq b$ zwięksź k o jeden.
7. Powtórz kroki 5-6 n razy.
8. Zwróć $\frac{k}{n}$.

Analogicznie tworzymy algorytm, mający na celu zbadanie zależności między prawdopodobieństwem wyjścia przez b a początkowym $x \in [a,b]$.

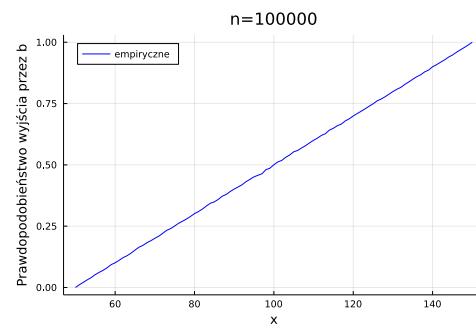
Algorytm:

1. Ustal różnicę dx , najmniejszy krok czasowy dt , liczbę powtórzeń symulacji n , oraz granice przedziału a i b .
2. Stwórz pusty wektor y .
3. Stwórz wektor v , początkowych x -ów należących do przedziału $[a,b]$, tak aby $v[i+1] - v[i] = dx$.
4. Dla każdego elementu z wektora v oblicz prawdopodobieństwo wyjścia przez b i wstaw otrzymany wynik do wektora y .
5. Zwróć y

Przeprowadzimy symulację dla $n = 10^3$ i $n = 10^5$, $dt = 0.1$, $dx = 1$, $a = 50$, $b = 150$. Otrzymujemy wyniki widoczne na rys. 3.6 i rys. 3.7



Rysunek 3.6



Rysunek 3.7

Po kształcie wykresu spodziewamy się, że jest to funkcja liniowa. Kiedy początkowe $x = a$, prawdopodobieństwo wyjścia przez b wynosi 0, natomiast gdy początkowe $x = b$, prawdopodobieństwo to będzie wynosić 1.

Znając dwa punkty przez które przechodzi funkcja liniowa $f(x) = \beta x + \gamma$, gdzie β i γ są szukanymi parametrami, możemy wyprowadzić ogólny wzór rozwiązując następujący układ równań:

$$\begin{cases} \beta a + \gamma = 0 \\ \beta b + \gamma = 1 \end{cases} \Leftrightarrow \begin{cases} \beta = \frac{-1}{a-b} \\ \gamma = \frac{a}{a-b} \end{cases}$$

Zatem:

$$f(x) = -\frac{1}{a-b}x + \frac{a}{a-b}$$

Podstawiając nasze dane ($a = 50$, $b = 150$), otrzymujemy:

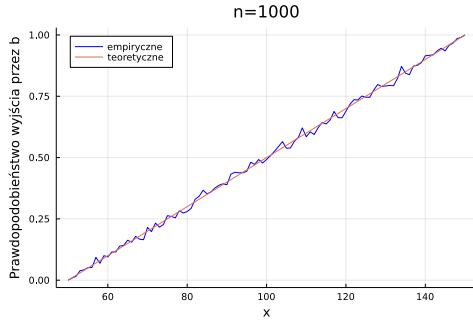
$$f(x) = \frac{1}{100}x - \frac{1}{2}$$

Sprawdzamy, czy moduł `numpy.polyfit` da nam podobny rezultat, przy jego pomocy otrzymujemy:

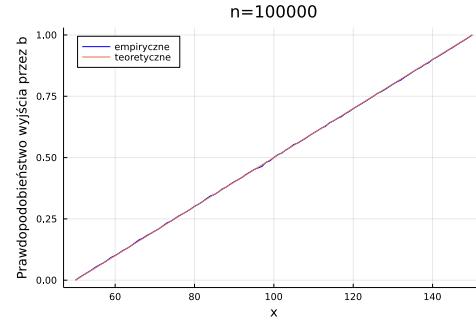
$$f(x) = 0.00997x - 0.497 \quad \text{dla } n = 10^3$$

$$f(x) = 0.01x - 0.5 \quad \text{dla } n = 10^5$$

Co zgadza się z naszymi teoretycznymi obliczeniami (im większe n tym większa dokładność). Na wykresach zestawienie empirycznej i teoretycznej funkcji prezentuje się następująco (rys. 3.8 i rys. 3.9):



Rysunek 3.8



Rysunek 3.9

Wnioski

W zadaniu szukaliśmy zależności między średnim czasem wyjścia z przedziału $[a,b]$ oraz prawdopodobieństwem, że wyjście to nastąpi przez b , a wartością początkową x . Musimy jednak pamiętać, że przeprowadzając symulację metodą numeryczną, dostaniemy jedynie przybliżone zachowanie procesu Wienera. Aby uzyskać rzeczywisty efekt, nasz minimalny krok czasowy dt musiałby dążyć do zera. Żeby wyniki były jak najbardziej rzetelne, musi zostać spełniony następujący warunek:

$$dt \ll |a - b|.$$

Należy jednak pamiętać, że dla ustalonego dt , zwiększając różnicę $|a - b|$, zwiększa się również czas wykonywania symulacji, dlatego mając ograniczone warunki, nie powinna być ona zbyt duża. Wykonując więcej powtórzeń n , również poprawiamy dokładność otrzymanych wyników, jednak tutaj również dobrze jest chować umiar.

Obserwując średni czas wyjścia, w zależności od początkowego x , widzimy że rośnie on wraz z odległością do bliższej granicy a lub b . Będzie on najdłuższy, kiedy $|x - a| = |x - b|$, czyli dla $x = \frac{a+b}{2}$. Wtedy też prawdopodobieństwo wyjścia przez granicę b jest równe prawdopodobieństwu wyjścia przez granicę a :

$$P(B_{\tau^x}^x = b) = P(B_{\tau^x}^x = a) = \frac{1}{2}$$

Ogólnie:

$$P(B_{\tau^x}^x = b) = 1 - P(B_{\tau^x}^x = a).$$

4 Podsumowanie

Przedstawione zadania pozwoliły dogłębiście przyjrzeć się zagadnieniom z zakresu rachunku prawdopodobieństwa, statystyki matematycznej oraz symulacji komputerowych. Poznaliśmy nowe metody, nowe testy, przypomnieliśmy sobie metody generowania zmiennych losowych. Ten projekt zdecydowanie spowodował utrwalenie bieżącej wiedzy. Pomimo wielu trudności, które napotkały nas w trakcie przygotowania tego raportu, jesteśmy ostatecznie zadowoleni z efektów. Jest to największy projekt na dotychczasowych studiach i cieszymy się, że mogliśmy go przedstawić. Dziękujemy za godziny poświęcone na konsultacjach oraz wyrozumiałość i pomoc w czasie ćwiczeń!

Literatura

- [1] R.M.Dudley - The Dvoretzky-Kiefer-Wolfowitz inequality with sharp constant - massarts 1990 proof.
- [2] Goodness of-Fit Techniques, edited by Ralph B. D'Agostino and Michael A.Stephens, p.97-101.
- [3] Cížek, Härdle, Weron - Statistical Tools for Finance and Insurance, p.307
- [4] Dokumentacja funkcji `numpy.polyfit` <https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>.