

Analiza Przeżycia

Raport 2

Jakub Zdancewicz, Ksawery Józefowski

Wygenerowano 6 października 2025

Table of contents

1	Wstęp	1
2	Estymatory funkcji przeżycia	2
2.1	Estymator Kaplana-Meiera	3
2.2	Estymator Fleminga-Harringtona	5
2.3	Oszacowanie ogona estymatora Kaplana-Meiera	7
2.4	Symulacja estymatora Kaplana-Meiera dla wygenerowanych danych cenzurowanych	8
3	Estymacja średniego czasu życia	11
4	Estymacja przedziałowa wartości średniego czasu życia	13
5	Weryfikacja hipotezy o jednakowym rozkładzie czasów życia	16

1 Wstęp

Sprawozdanie jest podzielone na cztery części.

Część pierwsza obejmuje implementację i wizualizację estymatorów Kaplana-Meiera oraz Fleminga-Harringtona funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekami A i B. Dodatkowo oszacujemy ogon funkcji przeżycia według propozycji Browna, Hollandera i Kowara, dopasowując odpowiedni model wykładniczy. Na zakończenie, na 1000 wygenerowanych zbiorach danych, wyznaczmy estymacje wartości funkcji przeżycia w wybranych punktach korzystając z estymatora Kaplana-Meiera z oszacowanym ogonem oraz przedstawimy histogramy tych estymacji.

W drugiej części zajmiemy się implementacją estymatorów średniego czasu życia, opartych na estymatorach Kaplana-Meiera oraz Fleminga-Harringtona (z opcją szacowania ogona). Następnie wykorzystamy te estymatory do oszacowania średnich czasów do remisji choroby u pacjentów leczonych lekiem A oraz lekiem B.

W części trzeciej skupimy się na metodzie wyznaczania przedziałów ufności dla średniego czasu życia na zadanym poziomie ufności. Następnie zastosujemy ją do określenia przedziałów ufności dla średniego czasu do progresji raka jajnika w dwóch badanych grupach pacjentek Mayo Clinic.

Na końcu pochylimy się nad problemem weryfikacji hipotezy o jednakowym rozkładzie czasów życia w kilku podpopulacjach. W szczególności zastosujemy testy log-rank, Gehana-Breslowa, Tarone'a-Warego oraz Peto-Peto do zweryfikowania hipotezy o jednakowym rozkładzie czasu do progresji choroby w dwóch badanych grupach pacjentek z poprzedniego punktu. Przeanalizujemy wyniki testów, ze szczególnym uwzględnieniem otrzymanych p -wartości, oraz zastanowimy się nad ich związkiem z funkcjami wagowymi oraz estymatorami funkcji przeżycia.

2 Estymatory funkcji przeżycia

Niech X_1, \dots, X_n będą niezależnymi zmiennymi losowymi o jednakowym, lecz nieznanym rozkładzie o dystrybuancie F . Zakładamy, że mamy do czynienia z danymi cenzurowanymi losowo. Z każdą zmienną X_i związana jest zatem zmienna losowa cenzurująca C_i , niezależna od X_i .

Obserwujemy więc zmienne

$$T_i = \min(X_i, C_i)$$

oraz

$$\delta_i = \begin{cases} 1, & \text{gdy } X_i \leq C_i, \\ 0, & \text{gdy } X_i > C_i \end{cases}$$

Naszym celem jest estymacja funkcji przeżycia, zdefiniowanej jako

$$S(t) = \mathbb{P}(X > t)$$

Estymację przeprowadzimy na podstawie danych dotyczących czasu remisji choroby pacjentów leczonych lekami A i B.

2.1 Estymator Kaplana-Meiera

Niech $t_{(1)} \leq \dots \leq t_{(n)}$ będą uporządkowanymi rosnąco wartościami wektora losowego (T_1, \dots, T_n) . Ponadto niech $\delta_{(i)}$, dla $i = 1, \dots, n$, oznacza wartość δ odpowiadającą obserwacji $t_{(i)}$.

Wówczas estymator Kaplana-Meiera definiujemy jako

$$\hat{S}(t) = \prod_{i: t_{(i)} \leq t} \left(\frac{r_i - d_i}{r_i} \right), \quad 0 \leq t \leq t_{(n)},$$

gdzie

$$r_i = \sum_{j=1}^n \mathbf{1}_{[t_{(i)}, \infty)}(t_j) \quad \text{oraz} \quad d_i = \sum_{j=1}^n \mathbf{1}_{\{t_{(i)}\}}(t_j) \mathbf{1}_{\{1\}}(\delta_j)$$

Wielkość r_i oznacza liczbę jednostek pozostających w stanie ryzyka w chwili $t_{(i)}$, natomiast d_i jest liczbą zdarzeń (nieocenzurowanych), które wystąpiły dokładnie w punkcie $t_{(i)}$.

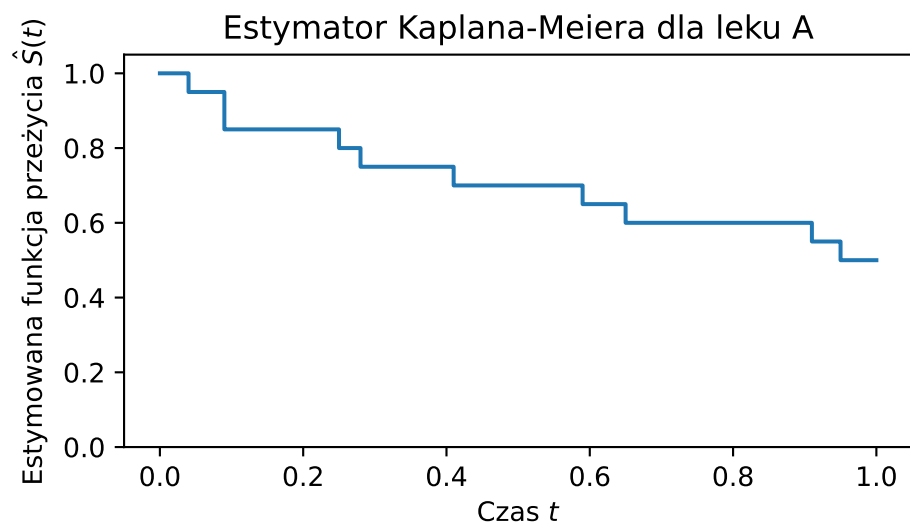
Poniżej przedstawiamy implementację tego estymatora w języku Python.

```
def S_hat(t, df):
    s_hat = 1
    data = df['Data'].to_numpy()
    delta = df['Delta'].to_numpy()
    order = data.argsort()
    data = data[order]
    delta = delta[order]

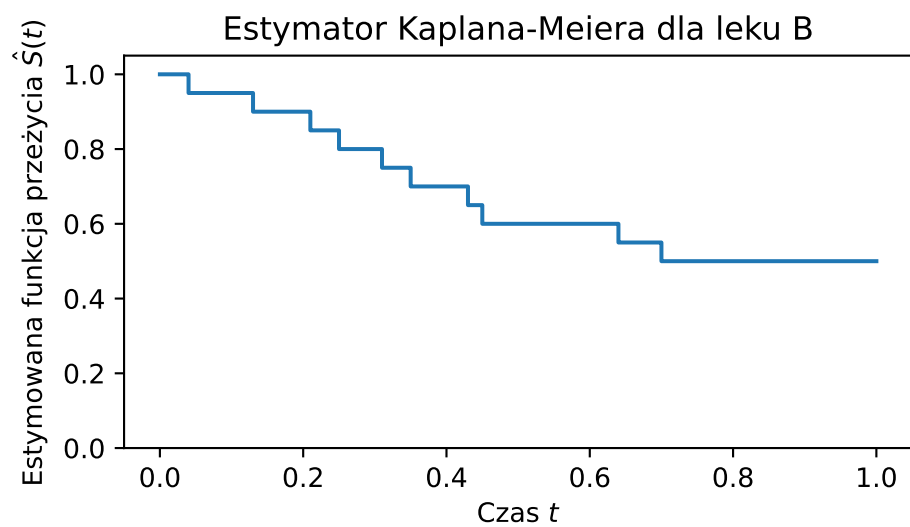
    unique_times = np.unique(data[data <= t])

    for s_i in unique_times:
        if s_i > t:
            break
        r_i = np.sum(data >= s_i)
        d_i = np.sum((data == s_i) & (delta == 1))
        if d_i == 0:
            continue
        s_hat *= (r_i - d_i) / r_i
    return s_hat
```

Na wykresach 1 i 2 przedstawiamy estymatory funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekami A i B, odpowiednio. Widzimy wyraźne różnice między wykresami estymatora, co pozwala przypuszczać, że leki A i B nie wykazują podobnego działania.



Rysunek 1: Wykres estymatorów Kaplana-Meiera funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekiem A



Rysunek 2: Wykres estymatorów Kaplana-Meiera funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekiem B

2.2 Estymator Fleminga-Harringtona

Innym popularnym estymatorem funkcji przeżycia jest estymator Fleminga-Harringtona. Wykorzystuje on zależność między funkcją przeżycia a funkcją hazardu h :

$$S(t) = \exp[-H(t)],$$

gdzie

$$H(t) = \int_0^t h(u) du,$$

oraz estymator skumulowanej funkcji hazardu zaproponowany przez Nelsona-Aalena:

$$\hat{H}(t) = \sum_{j: t_j \leq t} \frac{d_j}{r_j}, \quad \text{dla } 0 \leq t \leq t_{(n)},$$

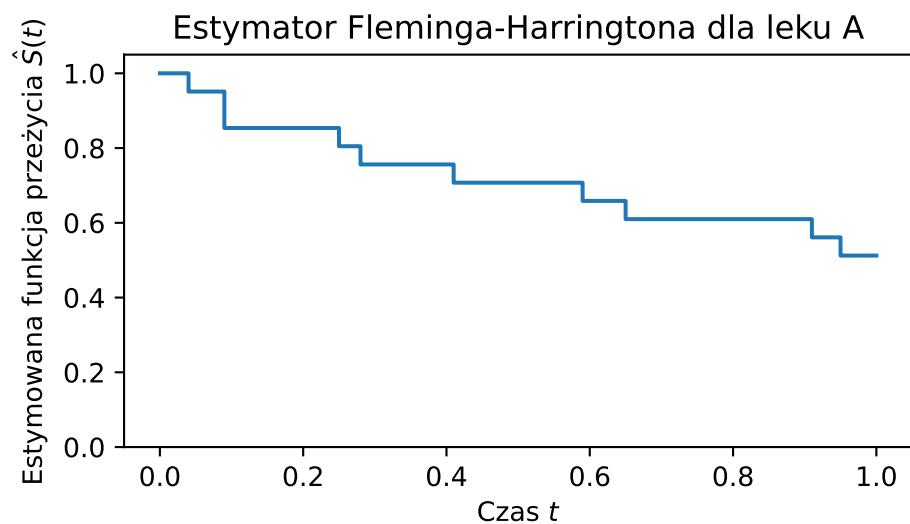
gdzie r_j i d_j są zdefiniowane tak samo jak w estymatorze Kaplana-Meiera.

Estymator Fleminga-Harringtona jest więc dany wzorem:

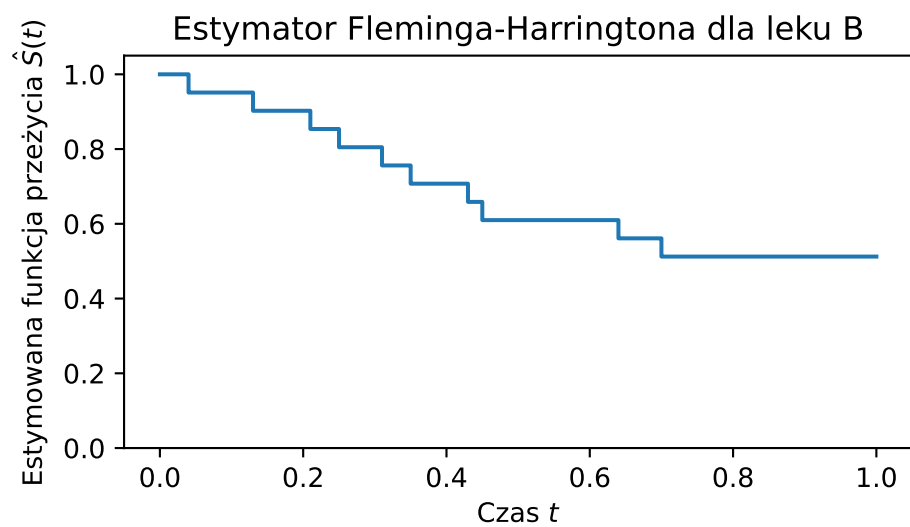
$$\hat{S}(t) = \exp\left(-\sum_{j: t_j \leq t} \frac{d_j}{r_j}\right), \quad \text{dla } 0 \leq t \leq t_{(n)}.$$

Poniżej przedstawiamy implementację tego estymatora w języku Python.

```
def S_tld(t, df):
    data = df['Data'].to_numpy()
    delta = df['Delta'].to_numpy()
    order = data.argsort()
    data = data[order]
    delta = delta[order]
    s_tld = 0.0
    for j in range(len(data)):
        if data[j] > t:
            break
        r_j = np.sum(data >= data[j])
        d_j = np.sum((data == data[j]) & (delta == 1))
        s_tld += d_j / r_j
    return np.exp(-s_tld)
```



Rysunek 3: Wykres estymatorów Fleminga-Harringtona funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekiem A



Rysunek 4: Wykres estymatorów Fleminga-Harringtona funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekiem B

Ponownie jak w przypadku estymatora Kaplana-Meiera, na wykresach 3 i 4 widać wyraźne różnice między wykresami estymatorów, co pozwala przypuszczać, że leki A i B nie wykazują podobnego działania.

2.3 Oszacowanie ogona estymatora Kaplana-Meiera

Estymator Kaplana-Meiera jest dobrze określony dla każdego $t \leq t_{(n)}$.

Dodatkowo, jeśli $\delta_{(n)} = 1$, oszacowanie funkcji przeżycia w przedziale $(t_{(n)}, \infty)$ również jest dobrze określone i wynosi 0.

Natomiast gdy $\delta_{(n)} = 0$, estymator Kaplana-Meiera nie jest zdefiniowany dla $t > t_{(n)} =: t^+$.

W takim przypadku Brown, Hollander i Kowar sugerują oszacowanie “ogona” funkcji przeżycia za pomocą odpowiednio dobranej funkcji przeżycia rozkładu wykładniczego, takiej, która w punkcie t^+ jest równa $\hat{S}(t^+)$.

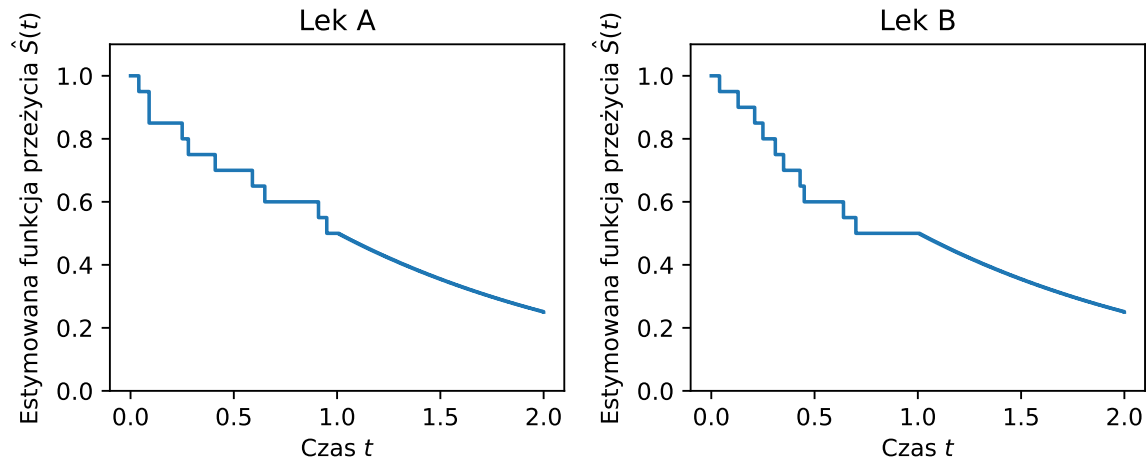
Stąd otrzymujemy:

$$\hat{S}(t) = \exp \left[\frac{\ln \hat{S}(t^+)}{t^+} t \right], \quad \text{dla } t > t^+$$

Poniżej przedstawiamy kod w Pythonie generujący oszacowane wartości funkcji przeżycia na ogonie, zgodnie z propozycją Browna, Hollandera i Kowara.

```
def get_tail(t, df):
    t_plus = np.max(df['Data'])
    theta = np.log(S_hat(t_plus, df)) / t_plus
    return np.exp(theta * t)
```

Na rysunku 5 przedstawiamy wykresy estymatora Kaplana-Meiera funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekami A i B, wraz z “ogonem” oszacowanym zgodnie z propozycją Browna, Hollandera i Kowara.



Rysunek 5: Wykresy estymatorów Kaplana-Meiera funkcji przeżycia czasu do remisji choroby u pacjentów leczonych lekami A oraz B wraz z ogonem wykładniczym

2.4 Symulacja estymatora Kaplana-Meiera dla wygenerowanych danych cenzurowanych

W tej sekcji przeprowadzimy symulację wartości estymatora Kaplana-Meiera z “ogonem” oszacowanym zgodnie z propozycją Browna, Hollandera i Kowara. Symulacja dotyczy wartości w punktach t_0 oraz $2t_0$, przy czym za parametr t_0 przyjęliśmy w przybliżeniu wartość oczekiwaną badanego rozkładu, tj. $t_0 = 1.5$.

Symulacja oparta będzie na danych cenzurowanych I-go typu, generowanych z rozkładu $\mathcal{GE}(\lambda = 1, \alpha = 2)$. Wygenerujemy $M = 1000$ niezależnych zbiorów danych dla trzech rozmiarów próby: $n \in \{30, 50, 100\}$.

Poniżej przedstawiamy kod symulacji w języku Python.

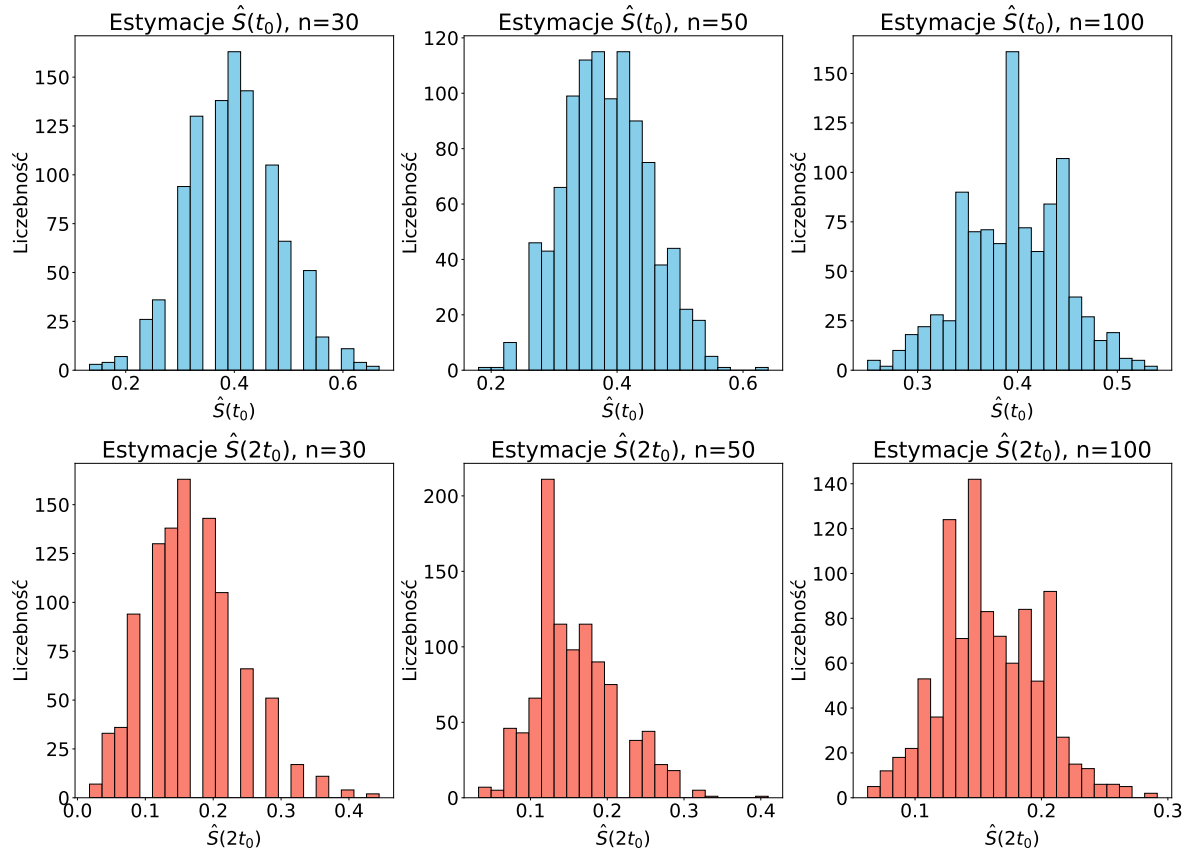
```
def sim(M=1000, alpha=2, L=1, n_values=[30, 50, 100]):
    t0=1.5
    results = {}
    np.random.seed(38)
    for n in n_values:
        est_t0 = []
        est_2t0 = []
        for _ in range(M):
            df = type_I(t0=t0, L=L, alpha=alpha, n=n)
            t_plus = np.max(df['Data'])
            if t0 <= t_plus:
```



```

        S_t0 = S_hat(t0, df)
    else:
        S_t0 = get_tail(t0, df)
    if 2*t0 <= t_plus:
        S_2t0 = S_hat(2*t0, df)
    else:
        S_2t0 = get_tail(2*t0, df)
    est_t0.append(S_t0)
    est_2t0.append(S_2t0)
    results[n] = {
        't0': np.array(est_t0),
        '2t0': np.array(est_2t0)
    }
return results

```



Rysunek 6: Histogramy wartości estymatora Kaplana-Meiera z “ogonem” w punktach t_0 i $2t_0$, uzyskane w wyniku symulacji dla $M = 1000$ niezależnych zbiorów danych cenzurowanych I-go typu z rozkładu $GE(\lambda = 1, \alpha = 2)$ dla różnych rozmiarów próby $n \in \{30, 50, 100\}$

Na wykresie 6 przedstawiamy histogramy oszacowań estymatora Kaplana-Meiera z ogonem w punktach t_0 oraz $2t_0$ dla różnych rozmiarów próby n .

Na podstawie przedstawionych histogramów widać, że wraz ze wzrostem rozmiaru próby n kształty rozkładu oszacowań estymatora Kaplana-Meiera stają się coraz mniej podobne do kształtu rozkładu normalnego. Obserwacja ta nie pozwala na wyciąganie wniosków o jego asymptotycznej normalności w punktach t_0 oraz $2t_0$.

3 Estymacja średniego czasu życia

Niech $\mu = \mathbb{E}[X]$ oznacza średni czas do wystąpienia zdarzenia. Jeżeli zmienna losowa X ma absolutnie ciągły rozkład o nośniku $(0, \infty)$ oraz funkcję przeżycia $S(t)$, to wartość oczekiwana może zostać zapisana jako

$$\mu = \int_0^\infty S(t) dt$$

Aby oszacować średni czas życia, zastępujemy nieznaną funkcję przeżycia $S(t)$ jej estymatorem. W dalszej części wykorzystujemy w tym celu estymator Kaplana-Meiera oraz estymator Fleminga-Harringtona.

W praktyce często zdarza się, że obserwacja $t_{(n)}$ jest cenzurowana. W takim przypadku rozważane estymatory $S(t)$ nie są określone dla $t > t_{(n)}$, co uniemożliwia bezpośrednie zastosowanie wzoru na wartość oczekiwaną. Aby oszacować ogon całki, niezbędne jest przedłużenie estymowanej funkcji przeżycia.

W tym celu stosujemy podejście Browna, Hollandera i Kowara, polegające na dopasowaniu wykładniczego ogona tak, aby pokrywał się on z estymowaną funkcją przeżycia w ostatnim niecenzurowanym punkcie t_{\max} .

Całkując tą aproksymację otrzymujemy

$$\int_{t_{\max}}^\infty S(t) dt = \int_{t_{\max}}^\infty \exp\left(\frac{\ln S(t_{\max})}{t_{\max}} t\right) dt = -\frac{t_{\max}}{\ln S(t_{\max})} S(t_{\max})$$

Wyrażenie to stanowi estymację brakującej części całki, którą należy dodać do części policzonej na przedziale $[0, t_{\max}]$.

Poniżej przedstawiamy implementację opisanego estymatora dla rozważanych funkcji przeżycia, z możliwością uwzględnienia ogona, w języku Python.

```

def est_mean(df, est="KM", wtail=True):
    if est == "KM":
        S_func = S_hat
    elif est == "FH":
        S_func = S_tld
    else:
        raise ValueError("Nieznany typ estymatora")
    data = df['Data'].values
    delta = df['Delta'].values
    sort_idx = np.argsort(data)
    data = data[sort_idx]
    delta = delta[sort_idx]
    t_max = np.max(data)
    times = np.concatenate([[0], data])
    times = times[times <= t_max]
    surv_prob = np.array([S_func(t, df) for t in times])
    mean_t = 0.0
    for i in range(len(times) - 1):
        dt = times[i + 1] - times[i]
        area = surv_prob[i] * dt
        mean_t += area
    if wtail:
        S_tmax = S_func(t_max, df)
        if S_tmax > 0:
            tail_integral = -t_max / np.log(S_tmax) * S_tmax
            mean_t += tail_integral
    return mean_t

```

Tabela 1: Średnie czasy przeżycia dla Leku A i Leku B oszacowane przy użyciu estymatorów Kaplana-Meiera i Fleminga-Harringtona

Lek	Estymator	Ogon	Średnia
A	KM	nie	0.7108
A	KM	tak	1.4321
A	FH	nie	0.7179
A	FH	tak	1.4840
B	KM	nie	0.6729
B	KM	tak	1.3943
B	FH	nie	0.6810
B	FH	tak	1.4471

W tabeli 1 przedstawiamy średnie czasy do remisji choroby pacjentów leczonych lekiem A i lekiem B, oszacowane za pomocą powyższej metody. Uwzględniamy estymatory Kaplana-Meiera oraz Fleminga-Harringtona, zarówno w wersjach z oszacowaniem ogona, jak i bez niego.

Można zauważyć, że w przypadku wersji bez szacowania ogona oszacowane średnie czasy za pomocą estymatorów Kaplana-Meiera oraz Fleminga-Harringtona są bardzo zbliżone. Natomiast w wersjach z oszacowaniem ogona różnice między średnimi stają się bardziej widoczne. Zgodnie z intuicją oraz na podstawie wcześniejszych wyników dotyczących leków można dojść do wniosku, że pominięcie szacowania ogona jest błędne. Rzeczywiście, dane cenzurowane nie zakładają, że zdarzenie nie występuje po momencie cenzury, zatem estymacja średniej wyłącznie na podstawie danych niecenzurowanych daje niedokładną wartość.

Zauważmy również, że istotne jest rozróżnienie pomiędzy samymi estymatorami. Estymator Kaplana-Meiera daje wartości nieco mniejsze niż estymator Fleminga-Harringtona.

4 Estymacja przedziałowa wartości średniego czasu życia

Do wyznaczenia asymptotycznego punktowowo przedziału ufności wykorzystamy asymptotyczne własności estymatora

$$\hat{\mu}_\tau = \int_0^\tau \hat{S}(t) dt,$$

gdzie τ jest przyjętym czasem odzwierciedlającym wiedzę a priori o maksymalnym czasie do wystąpienia zdarzenia.

Funkcję przeżycia S szacujemy na przedziale $(t_{(n)}, \tau)$ poprzez $\hat{S}(t_{(n)})$.

Okazuje się, że przy odpowiednich założeniach estymator ten jest asymptotycznie normalny

$$\hat{\mu}_\tau \sim \mathcal{AN}(\mu_\tau, V(\hat{\mu}_\tau)),$$

gdzie $V(\hat{\mu}_\tau)$ oznacza wariancję estymatora $\hat{\mu}_\tau$.

Wariancja ta jest zazwyczaj nieznana, dlatego musi zostać oszacowana. Za estymator wariancji estymatora $\hat{\mu}_\tau$ przyjmujemy

$$\hat{V}(\hat{\mu}_\tau) = \sum_{i=1}^D \left(\int_{s_i}^\tau \hat{S}(t) dt \right)^2 \frac{d_i}{r_i(r_i - d_i)},$$

gdzie r_i oraz d_i są zdefiniowane jak wcześniej, natomiast D oznacza liczbę różnych, uporządkowanych, niecenzurowanych czasów zdarzeń s_i .

Wtedy przedział postaci $[T_L, T_U]$, gdzie

$$T_L = \hat{\mu}_\tau - z(1 - \alpha/2) \sqrt{\hat{V}(\hat{\mu}_\tau)},$$

$$T_U = \hat{\mu}_\tau + z(1 - \alpha/2) \sqrt{\hat{V}(\hat{\mu}_\tau)},$$

a $z(q)$ oznacza kwantyl rzędu q standardowego rozkładu normalnego, stanowi asymptotyczny punktowy przedział ufności na poziomie istotności $1 - \alpha$ dla wartości średniej μ rozkładu czasu życia.

Poniżej przedstawiamy kod funkcji obliczającej wartości całki z funkcji $\hat{S}(t)$ na zadanym przedziale oraz funkcji zwracającej realizację przedziału ufności na wybranym poziomie istotności.

```
def int_s(df, tau, s=0, est="KM"):
    if est == "KM":
        S_func = S_hat
    elif est == "FH":
        S_func = S_tld
    else:
        raise ValueError("Nieznany typ estymatora")

    data = df["Data"].values
    times = data[(data >= s) & (data <= tau)]
    times = np.concatenate([s], times, [tau])
    times = np.sort(times)
    surv_vals = np.array([S_func(t, df) for t in times])

    integral = 0.0
    for i in range(len(times) - 1):
        dt = times[i+1] - times[i]
        integral += surv_vals[i] * dt

    return integral

def mu_estimate(df, tau, alpha=0.05):
    data = df["Data"].values
    delta = df["Delta"].values

    sort_idx = np.argsort(data)
    data = data[sort_idx]
    delta = delta[sort_idx]

    mu_tau = int_s(df, tau)
    event_times = np.unique(data[delta == 1])

    var = 0.0
```

```

for s_i in event_times:
    if s_i > tau:
        break
    r_i = np.sum(data >= s_i)
    d_i = np.sum((data == s_i) & (delta == 1))

    S = int_s(df, tau, s=s_i)

    var += (S ** 2) * d_i / (r_i * (r_i - d_i))

V = np.sqrt(var)
z = norm.ppf(1 - alpha/2)
lower = mu_tau - z*V
upper = mu_tau + z*V

return lower, upper

```

Rozważamy dane pochodzące z badania przeprowadzonego w Mayo Clinic, obejmującego pacjentki z rakiem jajnika w stadium II lub IIIA. Głównym celem jest ustalenie, czy stopień zaawansowania choroby był związany z czasem do progresji choroby. W tym celu wyznaczymy realizację przedziałów ufności dla średniego czasu do progresji choroby w dwóch badanych grupach pacjentek.

Przyjmujemy poziom istotności

$$\alpha = 0.05$$

oraz maksymalny czas do wystąpienia zdarzenia

$$\tau_1 = 760 \quad \text{oraz} \quad \tau_2 = 985$$

Wartości τ obliczyliśmy korzystając w obu grupach ze wzoru:

$$\tau = t_{\max}^{\text{niecenzurowany}} + (1 - p_c) \cdot \left(t_{\max}^{\text{cenzurowany}} - t_{\max}^{\text{niecenzurowany}} \right),$$

gdzie:

- $t_{\max}^{\text{niecenzurowany}}$ to ostatni moment, w którym odnotowano progresję choroby,
- $t_{\max}^{\text{cenzurowany}}$ to największy czas obserwacji w grupie,
- p_c oznacza proporcję obserwacji cenzurowanych w grupie.

Wzór uwzględnia intuicję, że im większa jest liczba danych cenzurowanych, tym mniejsza powinna być szansa na wystąpienie dalszych zdarzeń niecenzurowanych.

Dla tak wybranych wartości α i τ otrzymujemy następującą realizację przedziałów ufności:

```

full_A = np.array([28, 89, 175, 195, 309, 462])
censored_A = np.array([377, 393, 421, 447, 709, 744, 770, 1106, 1206])
data_A = np.concatenate([full_A, censored_A])
delta_A = np.array([True]*len(full_A) + [False]*len(censored_A))

full_B = np.array([34, 88, 137, 199, 280, 291, 309, 351, 358, 369,
                  369, 370, 375, 382, 392, 451])
censored_B = np.array([299, 300, 429, 1119])
data_B = np.concatenate([full_B, censored_B])
delta_B = np.array([True]*len(full_B) + [False]*len(censored_B))

df_A = pd.DataFrame({"Data": data_A, "Delta": delta_A})
df_B = pd.DataFrame({"Data": data_B, "Delta": delta_B})
tau_1 = 760
tau_2 = 985
L_A_1, R_A_1 = mu_estimate(df_A, tau_1)
L_B_2, R_B_2 = mu_estimate(df_B, tau_2)
L_A_3, R_A_3 = mu_estimate(df_A, tau_2)
L_B_4, R_B_4 = mu_estimate(df_B, tau_1)

```

Tabela 2: Realizacje przedziałów ufności funkcji przeżycia dla obu grup pacjentek przy różnych wartościach τ

Tau	Grupa	Przedział ufności
760	niski stopień	[379.71, 673.54]
760	wysoki stopień	[269.24, 427.67]
985	niski stopień	[445.11, 858.13]
985	wysoki stopień	[260.25, 476.04]

Wyniki przedstawione w tabeli 2 pokazują, że dla obu wartości τ zarówno górne, jak i dolne granice przedziałów ufności są niższe w przypadku pacjentek z wysokim stopniem zaawansowania choroby niż u pacjentek z niższym stopniem zaawansowania. Ponadto, przedziały te są węższe dla grupy o wysokim stopniu. Porównując wartości τ , można zauważyć, że ma on istotny wpływ w grupie o niskim stopniu zaawansowania, gdzie przedziały zostały przesunięte, natomiast w grupie o wysokim stopniu jego wpływ jest znikomy.

5 Weryfikacja hipotezy o jednakowym rozkładzie czasów życia

Celem tej części analizy jest porównanie czasu do wystąpienia progresji choroby w dwóch badanych grupach pacjentek. Przyjmujemy poziom istotności $\alpha = 0.05$. Weryfikowana jest

hipoteza zerowa o równości funkcji przeżycia w obu grupach:

$$H_0 : S_1(t) = S_2(t) \quad (\text{rozkłady są jednakowe}),$$

$$H_1 : S_1(t) \neq S_2(t) \quad (\text{rozkłady różnią się})$$

Statystyka testowa stosowana do weryfikacji hipotezy oparta jest na wektorze $k - 1$ statystyk $Z_j(\tau)$ i ma postać:

$$Z^2 = \mathbf{Z}(\tau) \Sigma^{-1} \mathbf{Z}(\tau)^T$$

gdzie:

- Σ jest macierzą oszacowanych wariancji $\hat{\sigma}_{jj}$ oraz kowariancji $\hat{\sigma}_{ij}$ statystyk $Z_j(\tau)$,
- $\mathbf{Z}(\tau) = (Z_1(\tau), \dots, Z_{k-1}(\tau))$

Statystyka Z^2 dąży asymptotycznie w rozkładzie do rozkładu χ^2 z $k - 1$ stopniami swobody.

W naszym przypadku rozważamy dwie grupy pacjentek, zatem $k = 2$. Statystyka $Z_j(\tau)$ ma postać:

$$Z_j(\tau) = \sum_{i=1}^D W(t_i) \left(d_{ij} - r_{ij} \frac{d_i}{r_i} \right)$$

gdzie stosujemy oznaczenia:

- D - liczba zdarzeń niecenzurowanych w całej próbie,
- d_{ij} - liczba zdarzeń w grupie j w chwili t_i ,
- r_{ij} - liczba pacjentek w grupie j w stanie ryzyka w chwili t_i ,
- $d_i = \sum d_{ij}$
- $r_i = \sum r_{ij}$
- $W(t_i)$ - funkcja wagowa charakterystyczna dla wybranego testu

W analizie porównujemy cztery warianty testu, różniące się doбором wag $W(t_i)$:

- test log-rank,
- test Gehana-Breslowa,
- test Tarone'a-Warego,
- Test Peto-Peto.

Wartości p obliczono przy użyciu pakietu `lifelines`. Wyniki analizy przedstawiamy w Tabeli 3.

```
def pval (test, df):
    if test == "log-rank":
        result = multivariate_logrank_test(df['Data'], df['Groups'],
                                           df['Delta'])
    elif test == "Gehan-Breslow":
        result = multivariate_logrank_test(df['Data'], df['Groups'],
                                           df['Delta'], weightings="wilcoxon")
    elif test == "Tarone-Ware":
        result = multivariate_logrank_test(df['Data'], df['Groups'],
                                           df['Delta'], weightings="tarone-ware")
    elif test == "Peto-Peto":
        result = multivariate_logrank_test(df['Data'], df['Groups'],
                                           df['Delta'], weightings="peto")
    return result
```

Tabela 3: Wartości p dla czterech analizowanych testów

Test	p-value
log-rank	0.0183
Gehan-Breslow	0.1342
Tarone-Ware	0.0550
Peto-Peto	0.1015

Wyniki zastosowanych testów wykazują wyraźną rozbieżność. Test log-rank prowadzi do odrzucenia hipotezy zerowej, podczas gdy pozostałe testy nie dają do tego podstaw. Dodatkowo otrzymane wartości p znacznie się różnią: testy Gehana-Breslowa oraz Peto-Peto dają wysokie wartości p , znacznie przekraczające poziom istotności α , test Tarone’a-Warego znajduje się blisko granicy istotności, natomiast test log-rank zwraca wartość zdecydowanie niższą niż α . Taka rozbieżność uniemożliwia jednoznaczne rozstrzygnięcie, czy hipotezę zerową należy odrzucić.

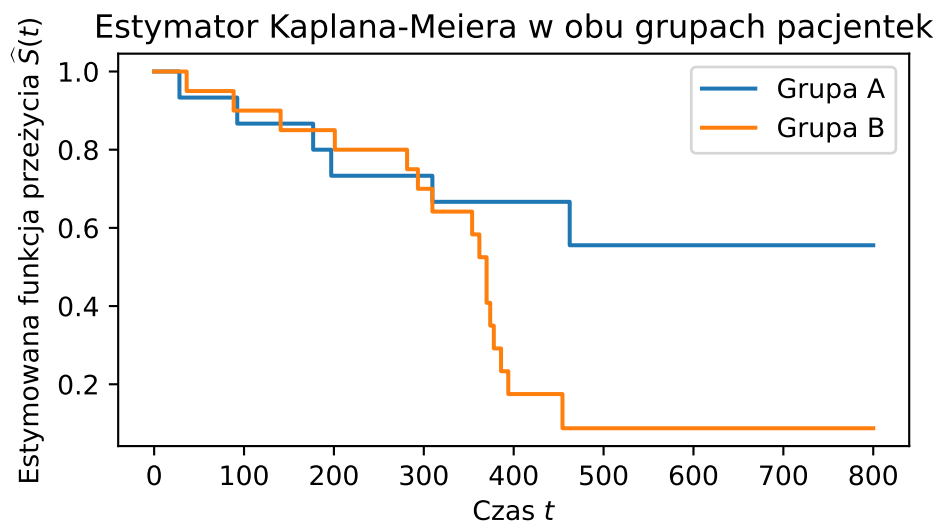
Różnice te mogą wynikać z odmiennych własności danych, na które poszczególne testy kładą nacisk poprzez zastosowane funkcje wagowe. Aby to zweryfikować, przeanalizujemy krzywe przeżycia dla obu grup oraz funkcje wagowe poszczególnych testów, przedstawione odpowiednio na rysunkach 7 i 8. Poniżej przedstawiamy też implementację używanych funkcji wagowych.

```

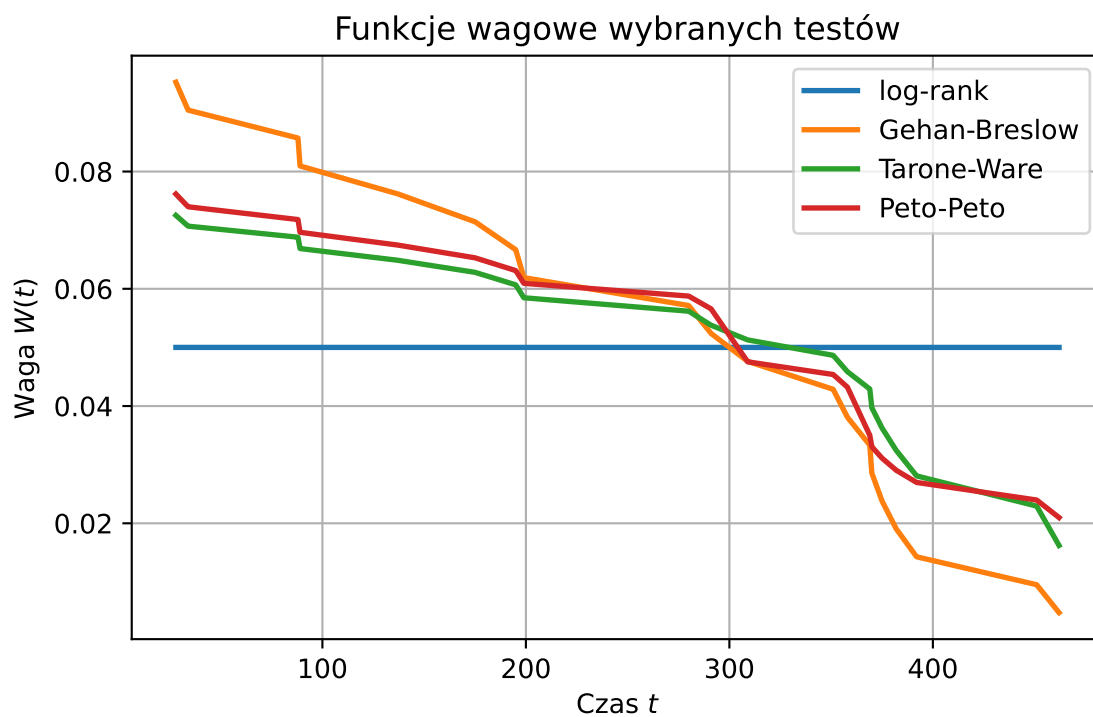
def W_t(test, df, df_uc):
    df_uc = np.asarray(df_uc)
    D = len(df_uc)
    r_i = np.array([np.sum(df_uc >= t) for t in df_uc])
    if test == "log-rank":
        W = np.ones(D)
    elif test == "Gehan-Breslow":
        W = r_i
    elif test == "Tarone-Ware":
        W = np.sqrt(r_i)
    elif test == "Peto-Peto":
        data = df['Data'].to_numpy()
        delta = df['Delta'].to_numpy()
        order = np.argsort(data)
        data = data[order]
        delta = delta[order]
        W = []
        for t in df_uc:
            W_ti = 1.0
            for j in range(len(data)):
                if data[j] > t:
                    break
                r_j = np.sum(data >= data[j])
                d_j = np.sum((data == data[j]) & (delta == 1))
                W_ti *= (1 - d_j / (r_j+1))

            W.append(W_ti)
        W = np.array(W)
    return W / np.sum(W)

```



Rysunek 7: Wykres estymatora Kaplana-Meiera funkcji przeżycia czasu do progresji choroby w obu grupach



Rysunek 8: Wizualizacja funkcji wagowej $W(t)$ dla czterech badanych testów

Na wykresie 7 możemy zauważyć, że funkcje przeżycia obu grup zachowują się bardzo podobnie dla małych wartości t , natomiast dla większych wartości czasu pojawiają się wyraźne różnice. W konsekwencji testy, które kładą większy nacisk na zdarzenia w późniejszym okresie obserwacji, są w stanie wykryć rozbieżności w rozkładach, podczas gdy testy skupiające się na początkowych zdarzeniach tego nie zrobią.

Analiza wykresu funkcji wagowych 8 potwierdza tę obserwację. Test log-rank stosuje równomierne wagi w całym okresie obserwacji, co pozwala mu wychwycić różnicę w rozkładach obu grup i skutkuje niską wartością p . Natomiast test Gehana-Breslowa mocno koncentruje się na początkowych zdarzeniach, a wagi dla późniejszych czasów są niewielkie, co tłumaczy wysoką wartość p tego testu.

Jeśli chodzi o testy Peto-Peto i Tarone'a-Warego, ich funkcje wagowe są podobne dla końcowych zdarzeń, ale różnią się na początku obserwacji. W efekcie test Tarone'a-Warego jest bardziej czuły na różnice w rozkładach niż test Peto-Peto, co sprawia, że jego wartość p znajduje się bliżej poziomu istotności.