Analiza Przeżycia Raport 1

Paweł Matławski
album 249732

10 listopada 2020

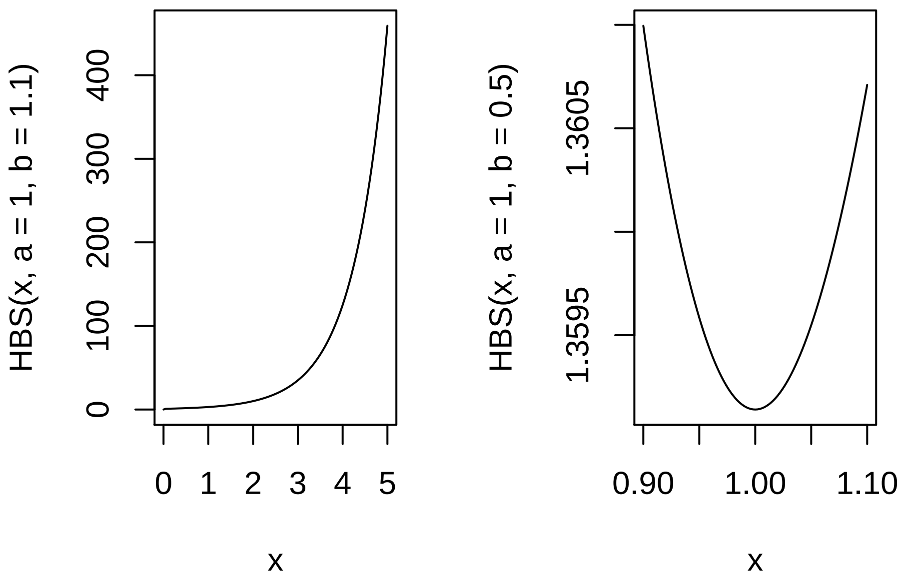
Spis treści

1 Lista nr 1

1.1 Zadanie 1

Narysujemy dwie funkcje hazardu rozkładu $BS(\alpha, \beta)$, odpowiadające wybranym parametrom, tak aby w pierwszym przypadku funkcja była rosnąca, a w drugim miała kształt wannowy.

```
dbs <- function(t, a, b)
{
  return(a*b*exp(a)*(t^(b-1))*exp(t^b)*exp(-a*exp(t^b)))
}

SBS <- function(t, a, b)
{
  exp(a*(1-exp(t^b)))
}

HBS <- function(t, a, b){dbs(t, a, b)/SBS(t, a, b)}
par(mfrow=c(1,2))
curve(HBS(x, a=1, b=1.1), xlim = c(0, 5))
curve(HBS(x, a=1, b=0.5), xlim = c(0.90,1.1))
```



Rysunek 1: Wykresy funkcji hazardu:rosnącej i o kształcie wannowym

1.2 Zadanie 2

Program do generowania zmiennych losowych o rozkładzie $BS(\alpha, \beta)$

Wyprowadzimy wzór na wygenerowanie zmiennej losowej za pomocą dystrybuanty rozkładu i zmiennych losowych o rozkładzie jednostajnym na odcinku $(0, 1)$

$$S(t, \alpha, \beta) = \exp(\alpha[1 - \exp(t^\beta)])$$

$$S(t, \alpha, \beta) = 1 - F(t, \alpha, \beta)$$

$$\exp(\alpha[1 - \exp(t^\beta)]) = 1 - F(t, \alpha, \beta)$$

$$\alpha[1 - \exp(t^\beta)] = \ln(1 - F(t, \alpha, \beta))$$

$$\exp(t^\beta) = 1 - \frac{\ln(1 - F(t, \alpha, \beta))}{\alpha}$$

$$t = (\ln(1 - \frac{\ln(1 - F(t, \alpha, \beta))}{\alpha}))^{\frac{1}{\beta}}$$

Mając dystrybuantę odwrotną, możemy wygenerować zmienne losowe o rozkładzie $BS(t, \alpha, \beta)$ za pomocą rozkładu $U(0, 1)$.

$F^{-1}(U_1), ..., F^{-1}(U_n)$ to zmienne losowe z rozkładu o dystrybuancie F

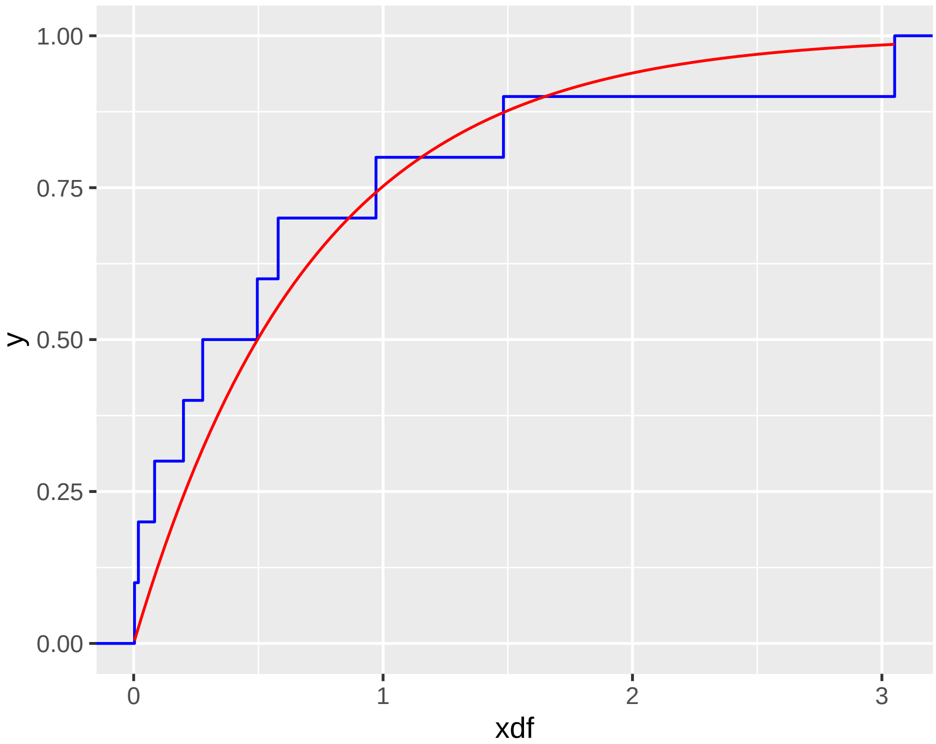
```
RVBS <- function(n, a, b)
{
  return(log(1-log(1-runif(n))/a)^(1/b))
}
```

1.3 Zadanie 3

Po wygenerowaniu 10 liczb z rozkładu $BS(\alpha, \beta)$, narysowaliśmy dystrybuantę empiryczną oraz teoretyczną dla tych danych.

```
RVBS <- function(n, a, b)
{
  return(log(1-log(1-runif(n))/a)^(1/b))
}

xdf <- RVBS(10, a = 1, b = 0.5)
library('ggplot2')
library(MASS)

parameters <- fitdistr(xdf, densfun = 'exponential')
parameters$estimate

##     rate
## 1.395962
```

```r
test <- function(x) {pexp(x, rate = parameters$estimate)}
ggplot(data = as.data.frame(xdf), aes(x=xdf))+
  geom_step(stat="ecdf", color='blue')+
  stat_function(fun = test, col='red')
```



Rysunek 2: Dystrybuanta empiryczna i teoretyczna

1.4 zadanie 4

Po wygenerowaniu 100 liczb z rozkładu $BS(\alpha, \beta)$ wyznaczam podstawowe wartości statystyk opisowych i ilistruuję je na odpowiednich rysunkach.

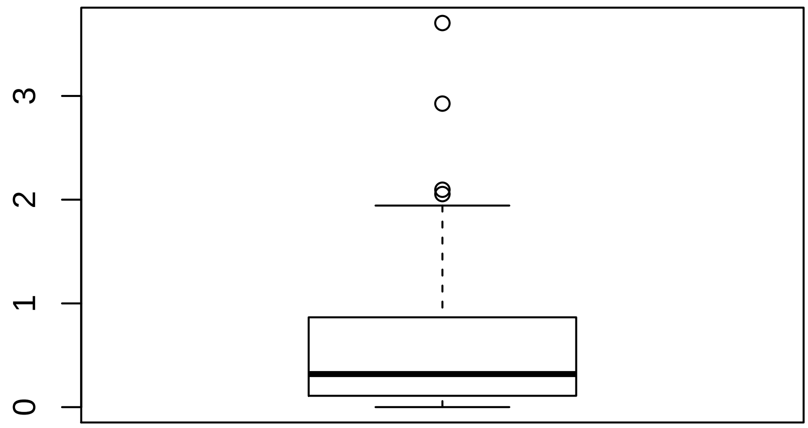
```r
library(xtable)
library(MASS)
library(EnvStats)

##
## Attaching package: 'EnvStats'
## The following object is masked from 'package:MASS':
##
##     boxcox
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```
xdf2 <- RVBS(100, a=1, b=0.5)
statopis <- c(summary(xdf2), sd(xdf2), IQR(xdf2))
statopis.matrix <- matrix(statopis, nrow = 1, ncol = length(statopis), dimnames = list(c
xtable(statopis.matrix)
```

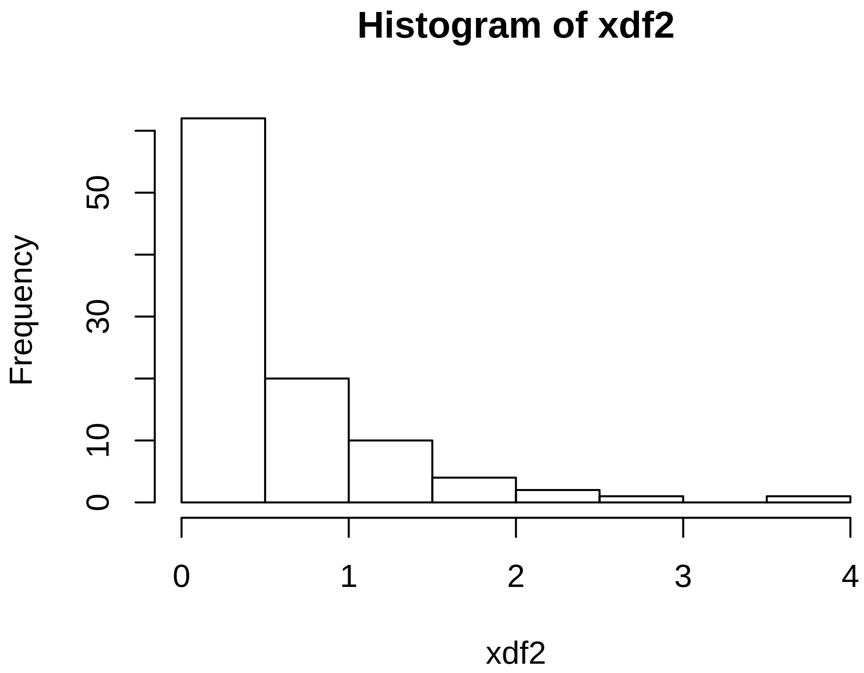
Min	1st Qu	Mediana	Średnia	3rd Qu	Max	Odch.Stand.	IQR
0.00	0.11	0.32	0.57	0.86	3.70	0.66	0.75

```
boxplot(xdf2)
```



Rysunek 3: boxplot

```
hist(xdf2)
```



Rysunek 4: histogram

2 Lista nr 2

2.1 Zadanie 1

Trzy programy do generowania danych cenzurowanych, odpowiednio I-go typu, II-go typu i losowego.

```
censvar <- function(n, fi, t0)
{
  C = rexp(n, 1/fi)
  t=rep(t0, n)
  C_1 = pmin(C, t)
  return(sort(C_1))
}

censvar2 <- function(n, fi, m)
{
  C_21 = sort(rexp(n, fi))
  C_22 = C_21[0:m]
  C_2 = c(C_21[0:m],rep(max(C_22), n-m))
  return(C_2)
}

censvar3 <- function(n, fi, ni)
```

```
{
  C_31 = rexp(n, 1/fi)
  C_32 = rexp(n, 1/ni)
  C_3 = pmin(C_31, C_32)
  delta = as.vector(as.numeric(C_31<C_32))
  results = mapply(list, C_3, delta)
  return(results)
}
```

2.2 Zadanie 2

Generuję dane cenzurowane I-go typu, używając funkcji z zadania pierwszego, i opisuję je używając wartości rozsądnych i ilustruję je wybierając odpowiednie formy wizualizacji.

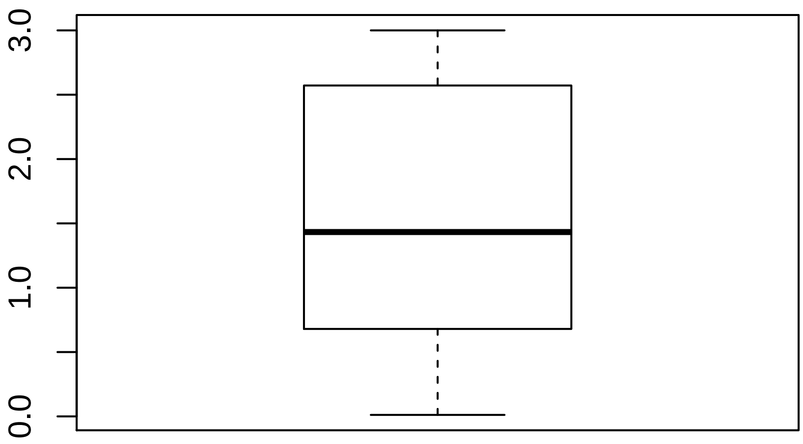
```
data2 <- censvar(100, 2, 3)
data2
```

[1] 0.01166048 0.01489999 0.02485855 0.07063021 0.08506102 0.12750401 [7] 0.13601740 0.20788120 0.25144719 0.26822806 0.30585493 0.34222348 [13] 0.36131825 0.36498311 0.38003530 0.38319352 0.42884367 0.43853539 [19] 0.43991965 0.46437855 0.49103222 0.59121992 0.59699365 0.65460832 [25] 0.67792003 0.68158888 0.70344070 0.71194280 0.72443014 0.91495218 [31] 0.92212295 0.93184225 0.95780279 0.97056761 1.00136459 1.02012885 [37] 1.02232176 1.04424964 1.09136414 1.10655327 1.13083044 1.23201003 [43] 1.30121933 1.33112789 1.36077636 1.36954623 1.37593582 1.40637795 [49] 1.41796946 1.42814060 1.43761923 1.47244385 1.50275262 1.52603688 [55] 1.59555792 1.69035984 1.73940130 1.91913771 1.93673732 1.97426228 [61] 2.00262256 2.01623837 2.07515670 2.12553266 2.16309858 2.18525733 [67] 2.21979303 2.24168314 2.30428348 2.34835738 2.40954442 2.46212228 [73] 2.47041703 2.48536473 2.55519202 2.58811665 2.71372513 2.93563621 [79] 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 [85] 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 [91] 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 3.00000000 [97] 3.00000000 3.00000000 3.00000000 3.00000000

```
library(EnvStats)
library(MASS)
library(xtable)
statopis2 <- c(min(data2), quantile(data2, 1/4), median(data2), quantile(data2, 3/4), IQ
statopis2.matrix <- matrix(statopis2, nrow = 1, ncol = length(statopis2), dimnames = lis
xtable(statopis2.matrix)
```

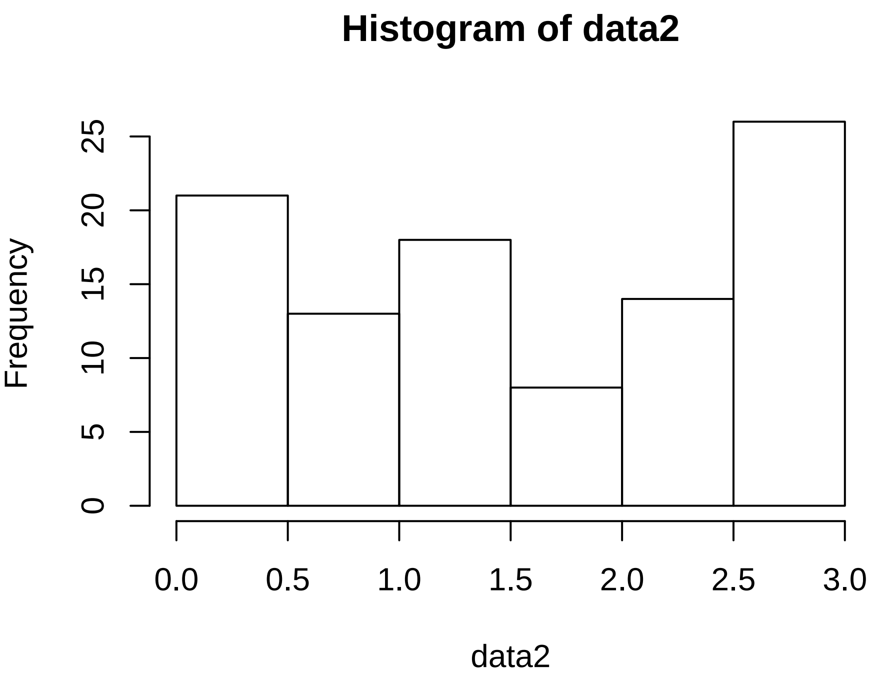
Min	1st Qu	Mediana	3rd Qu	IQR
0.01	0.68	1.43	2.56	1.88

```
boxplot(data2)
```



Rysunek 5: boxplot

```r
hist(data2)
```



Rysunek 6: histogram

3 Lista nr 3

3.1 Zadanie 1

Przeprowadzamy czas na 20 komponentach, których czasy do pierwszej awarii mają rozkład wykładniczy. W ciągu 2 lat obserwacji awarii uległo 10 komponentów. Na podstawie udokumentowanych danych podajemy oszacowanie największej wiarygodności średniego czasu do pierwszej awarii oraz wyznaczymy przedział ufności na poziomie ufności 0.99.

```
data <-  c(0.497, 0.638, 0.703, 0.839, 0.841, 0.950, 1.054, 1.103, 1.125, 1.495)
#1
n <- 20
t_0 <- 2
R <- length(data)
T_1 <- sum(data) + t_0 * (n-R)
# z wykładu
ni <- T_1/R
print(ni)
```

[1] 2.9245

```
#z uwagi z wykładu
library(binom)
confint <- binom.confint(R, n, 0.99)
#wybierajac jeden z przedzialow
confint$philower <- 1/(-log(1-confint$upper)/t_0)
confint$phiupper <- 1/(-log(1-confint$lower)/t_0)
library(xtable)
xtable(confint)
```

	method	x	n	mean	lower	upper	philower	phiupper
1	agresti-coull	10.00	20.00	0.50	0.25	0.75	1.44	6.94
2	asymptotic	10.00	20.00	0.50	0.21	0.79	1.29	8.39
3	bayes	10.00	20.00	0.50	0.24	0.76	1.39	7.39
4	cloglog	10.00	20.00	0.50	0.20	0.74	1.49	8.78
5	exact	10.00	20.00	0.50	0.22	0.78	1.31	8.14
6	logit	10.00	20.00	0.50	0.24	0.76	1.40	7.28
7	probit	10.00	20.00	0.50	0.24	0.76	1.38	7.46
8	profile	10.00	20.00	0.50	0.23	0.77	1.38	7.49
9	lrt	10.00	20.00	0.50	0.23	0.77	1.38	7.49
10	prop.test	10.00	20.00	0.50	0.30	0.70	1.66	5.62
11	wilson	10.00	20.00	0.50	0.25	0.75	1.44	6.94

3.2 zadanie 2

Wykonujemy te sama zadania co w zadaniu 1, tylko zakładamy że nasze dane były rejestrowane do 10 pomiaru.

```
m <- length(data)
T_2 <- sum(data) + (n-m)*data[m]
ni_2 <- T_2/m
ni_2

## [1] 2.4195

#dla kwantyli 0.005 i 0.995
a <- qgamma(0.005, shape = m, scale = 1/m)
b <- qgamma(1-0.005, shape = m, scale = 1/m)

lower_2 <- data.frame("lower"=ni_2/b)
upper_2 <- data.frame("upper"=ni_2/a)
confidence2 <- data.frame(lower_2, upper_2)
confidence2

##      lower    upper
## 1 1.209845 6.509418
```

4 Lista nr 4

4.1 Zadanie 1

Bazując na danych z zadania 3.1 (cenzurowane I-go typu), zweryfikujmy hipotezę $H_0 : \mu \geq 2.9$ do hipotezy alternatywnej $H_1 : \mu < 2.9$ na poziomie istotności $\alpha = 0.01$

Wyznaczmy statystykę $\lambda(t^*)$, która wyraża się jako iloraz supremum funkcji

$L(t, \theta) = \frac{n!}{(n-r)!}\theta^r \exp -\theta[\sum_{i=1}^r x_i + t0(n-r)]$

gdzie w liczniku bierzemy supremum po tecie należącej do Θ_0, a w mianowniku po tecie należącej do $(0, \inf)$. Korzystając z tw. Wilksa, wyznaczymy postać testu asymptotycznego.

```
t_0 <- 2
n <- 20
m <- length(data)
T_1 <- sum(data) + t_0*(n-m)
teta0 <-1/2.9
teta1 <- T_1/m
teta1

## [1] 2.9245
```

```
lamb0 <- exp(-qchisq(1-0.01, 1)/2)
lamb0

## [1] 0.0362452

lamb <- (teta0 * T_1 / m)^m * exp(m - teta0 * T_1)
lamb

## [1] 0.9996452
```

Sprawdzamy, czy $\lambda > \lambda_0$

```
lamb>lamb0

## [1] TRUE
```

Zatem hipoteza H_0 jest prawdziwa.

4.2 Zadanie 2.

Robimy analogicznie rozpatrując nasze dane teraz jako cenzurowane II-go typu.

```
m <- length(data)
T_2 <- sum(data) + (n-m)*max(data)
teta2 <- T_2/m
lamb2 <- (1/2.9^10*exp(-1/2.9*T_2))/(1/teta2^10*exp(-1/teta2*T_2))
lamb2

## [1] 0.8567563
```

Sprawdzamy, czy $\lambda_2 > \lambda_0$

```
lamb2 >lamb0

## [1] TRUE
```

Zatem hipoteza H_0 jest prawdziwa.