



WYDZIAŁ FIZYKI TECHNICZNEJ  
I MATEMATYKI STOSOWANEJ

Politechnika Gdańska  
Wydział Fizyki Technicznej i Matematyki Stosowanej  
Matematyka – Analityk Danych

# Modelowanie Matematyczne I Symulacje Komputerowe

## Projekt 1

*Aleksandra Burczyk*  
*Monika Gmaj*  
*Kinga Jankowska*  
*Kamil Łangowski*

prowadzący:  
dr inż. Anna Szafrńska

2 lutego 2023

# Spis treści

<b>1</b>	<b>Część I</b>	<b>2</b>
1.1	Zadanie 1 . . . . .	2
1.1.1	1.1 Rozwiązanie analityczne . . . . .	2
1.1.2	1.2 Wykres dystrybuanty i dystrybuanty odwrotnej . . . . .	3
1.1.3	1.3 Miary, rozproszenia i położenia . . . . .	4
1.1.4	1.4 Histogram i wykres gęstości . . . . .	6
1.1.5	1.5 Dystrybuanta empiryczna i dystrybuanta wybranego roz- kładu . . . . .	7
1.2	Zadanie 2 . . . . .	10
1.2.1	2.1 Normalizacja . . . . .	10
1.2.2	2.2 Generowanie próbek korzystając z iteracji mapy cha- otycznej . . . . .	10
1.2.3	2.3 Histogramy wygenerowanych rozkładów . . . . .	12
1.2.4	2.4 Rozrzut punktów . . . . .	21
1.2.5	2.5 Zbliżenie do rozkładu jednorodnego . . . . .	26
1.3	Zadanie 3 . . . . .	26
1.3.1	3.1 Normalizacja . . . . .	27
1.3.2	3.2 Generowanie próbek korzystając z iteracji mapy cha- otycznej . . . . .	27
1.3.3	3.3 Histogramy wygenerowanych rozkładów . . . . .	28
1.3.4	3.4 Rozrzut punktów . . . . .	37
1.3.5	3.2 Generowanie próbek korzystając z iteracji mapy cha- otycznej . . . . .	42
1.3.6	3.3 Histogramy wygenerowanych rozkładów . . . . .	43
1.3.7	3.4 Rozrzut punktów . . . . .	52
<b>2</b>	<b>Część II</b>	<b>57</b>
2.1	Zadanie 1 . . . . .	57
2.2	Zadanie 2 . . . . .	57
2.3	Zadanie 3 . . . . .	58
2.4	Zadanie 4 . . . . .	60
2.5	Zadanie 5 . . . . .	61
2.6	Zadanie 6 . . . . .	62
2.7	Zadanie 7 . . . . .	62
2.8	Zadanie 8 . . . . .	62
2.9	Zadanie 9 . . . . .	63
2.10	Zadanie 10 . . . . .	66
2.11	Zadanie 11 . . . . .	67
<b>3</b>	<b>Kod programu</b>	<b>68</b>

# 1 Część I

## 1.1 Zadanie 1

Wykorzystując metodę dystrybuanty odwrotnej wygenerować ciąg liczb pseudolosowych o wybranym rozkładzie (Arcus Sinus, Cauchy, Wartości ekstremalnych, Rayleigh, Weibull, Logistyczny, Pareto, Trójkątny) przy zadanej gęstości  $f(x)$ .

### 1.1.1 1.1 Rozwiązanie analityczne

Wyznaczyć w sposób analityczny dystrybuantę i dystrybuantę odwrotną.

Dystrybuanta jest funkcją ściśle rosnącą, a jej wzór ma postać:

$$F : \mathbb{R} \rightarrow [0, 1]$$

$$F(x) = P(X \leq x)$$

Wówczas jej funkcja odwrotna określona jest wzorem:

$$F^{-1} : [0, 1] \rightarrow \mathbb{R}$$

$$u = F(x) \quad x \in [0, 1]$$

$$x = F^{-1}(u) \quad x \in [0, 1]$$

**Twierdzenie** Niech  $U$  będzie zmienną losową o rozkładzie jednostajnym na  $(0, 1)$  ( $U \sim U(0, 1)$ ), a  $F$  będzie ściśle rosnącą ciągłą dystrybuantą pewnego rozkładu prawdopodobieństwa. Zmienna losowa  $X := F^{-1}(U)$  ma rozkład prawdopodobieństwa o dystrybuancie  $F$ .

Do wyznaczenia dystrybuanty potrzebna jest gęstość. Funkcja dystrybuanty dla Arcus Sinus przedstawiona jest wzorem:

$$f(x) = \frac{1}{\pi \sqrt{x(1-x)}}$$

Liczmy całkę z gęstości:

$$\begin{aligned} F(x) &= \int_0^x \frac{1}{t \sqrt{t(1-t)}} dt = \frac{1}{\pi} \int_0^x \frac{1}{\sqrt{t} \sqrt{1-t}} dt = (1) = \frac{1}{\pi} \int_0^1 \frac{2ds}{\sqrt{1-s^2}} = \\ &= \frac{2 \arcsin(s)}{\pi} \Big|_0^1 = \frac{2 \arcsin(\sqrt{x})}{\pi} - \frac{2 \arcsin(\sqrt{-\infty})}{\pi} = \frac{2 \arcsin(\sqrt{x})}{\pi} \\ (1) \quad s &= \sqrt{t} \quad s^2 = t \quad 2s ds = dt \end{aligned}$$

Liczmy dystrybuantę odwrotną:

$$F(x) = y = \frac{2 \arcsin(\sqrt{x})}{\pi}$$

$$\frac{y\pi}{2} = \arcsin(\sqrt{x})$$

$$\sin\left(\frac{y\pi}{2}\right) = \sqrt{x}$$

$$\sin^2\left(\frac{y\pi}{2}\right) = x$$

Ze wzoru  $1 - \cos(\alpha) = 2\sin^2(\frac{\alpha}{2})$  mamy :

$$F^{-1}(x) = \frac{1}{2} - \frac{1}{2}\cos(\pi x)$$

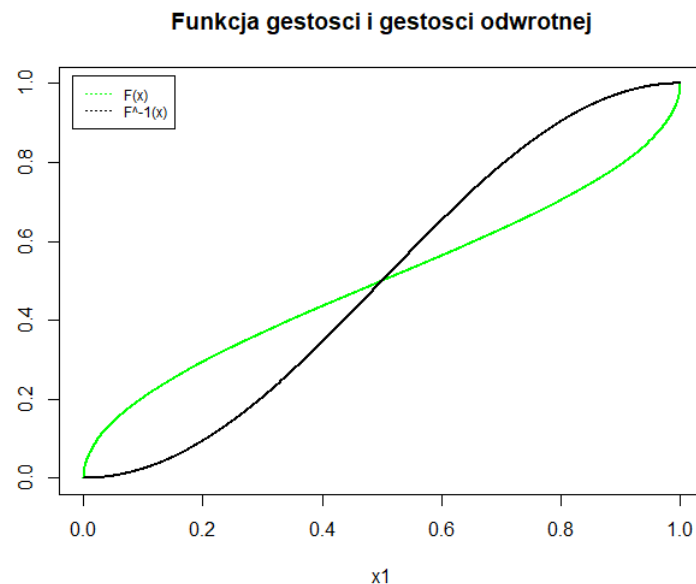
### 1.1.2 1.2 Wykres dystrybuanty i dystrybuanty odwrotnej

Przedstawić obie dystrybuanty na jednym wykresie.

```

1 dystybuanty <- function(x1,x2,d1,d2, ylimits) {
2   plot(x1,d1(x1),type = "l",ylim = ylimits,col="green",ylab = "",
3     lwd=2,
4   main = "Funkcja gestosci i gestosci odwrotnej")
5   lines(x2,d2(x2), lwd=2)
6   legend("topleft",inset=.02, legend=c("F(x)", "F^-1(x)"), col=c("
7     green", "black"),lty=3, cex=0.75, box.lty=1)
8
9   par(mfrow=c(1,1))
10  n = 1000
11  dystybuanty(seq(0,1,length=n), seq(0,1,length=n), Arcsin, inv
12    _Arcsin, c(0,1))

```



Rysunek 1: Dystrybuanta i dystrybuanta odwrotna rozkładu Arcus Sinus

### 1.1.3 1.3 Miary, rozproszenia i położenia

Wyznaczyć wybrane miary rozproszenia i położenia, następnie porównać z parametrami rozkładu teoretycznego

Aby wyznaczyć medianę przyrównujemy dystrybuantę rozkładu do 0.5.

$$F(x) = \frac{2\arcsin(\sqrt{x})}{\pi} = 0.5$$

$$\arcsin(\sqrt{x}) = \frac{\pi}{4}$$

$$\sin\left(\frac{\pi}{4}\right) = \sqrt{x}$$

$$x = \left(\frac{\sqrt{2}}{2}\right)^2$$

Obliczając wartość oczekiwaną korzystamy ze wzoru w którym  $F(x)$  jest dystrybuantą a  $f(x)$  to gęstość rozkładu.

$$\int_{-\infty}^{\infty} x dF(x) = \int_{-\infty}^{\infty} x f(x) dx$$

$$\begin{aligned} \int_0^1 \frac{x}{\pi\sqrt{x}\sqrt{1-x}} dx &= \frac{1}{\pi} \int_0^1 \frac{x}{\sqrt{x}\sqrt{1-x}} dx = (2) = \frac{2}{\pi} \int_0^1 \frac{t^2}{\sqrt{1-t^2}} dt = \\ &= \frac{2}{\pi} \left( \frac{\arcsin\sqrt{x}}{2} - \frac{\sqrt{x}\sqrt{1-x}}{2} \right) \Big|_0^1 = \frac{2}{\pi} \left( \frac{\pi}{4} - 0 \right) = \frac{1}{2} \end{aligned}$$

(2)

$$t = \sqrt{x} \quad t^2 = x \quad 2t dt = \frac{1}{\sqrt{x}} dx$$

Do wyliczenia dystrybuanty potrzebujemy pochodnej gęstości  $f(x)$ :

$$\begin{aligned} f'(x) &= \left( \frac{1}{\pi\sqrt{x}\sqrt{1-x}} \right)' = \frac{1}{\pi} \left( - \frac{\sqrt{x}\sqrt{1-x}}{(1-x)x} \right)' = - \frac{1}{\pi} \frac{\frac{(1-x)'\sqrt{x}}{2\sqrt{1-x}} + \frac{\sqrt{1-x}}{2\sqrt{x}}}{(1-x)x} = \\ &= \frac{1}{2x(1-x)\sqrt{1-x}\sqrt{x}} - \frac{1}{2\pi x\sqrt{1-x}\sqrt{x}} \end{aligned}$$

Funkcja ta osiąga swoje minimum w punkcie  $x = \frac{1}{2}$  a zatem na swoich krańcach osiągnie maksimum (są to punkty 0 i 1). Dlatego dominanta równa jest 0.1.

Kurtoza zostanie obliczona za pomocą pierwszego, drugiego i trzeciego kwantyla.

$$Q\left(\frac{1}{4}\right) = 0.15$$

$$Q\left(\frac{1}{2}\right) = 0.5$$

$$Q\left(\frac{3}{4}\right) = 0.85$$

Wariancję obliczamy ze wzoru  $Var(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$ . Z obliczeń powyżej wiemy, że  $\mathbb{E}[X] = \frac{1}{2}$

$$\mathbb{E}[X^2] = \int_{-\infty}^{\infty} x^2 f(x) dx$$

$$\int_0^1 \frac{x}{\pi \sqrt{x} \sqrt{1-x}} dx = \frac{1}{\pi} \int_0^1 \frac{2t^4}{\sqrt{1-t^2}} = \frac{1}{\pi} \left( \frac{3 \arcsin(\sqrt{x})}{4} - \frac{x^{\frac{1}{2}} \sqrt{1-x}}{2} - \frac{3\sqrt{x} \sqrt{1-x}}{4} \right) \Big|_0^1 = \frac{3}{8}$$

Stąd mamy:

$$Var(X) = \frac{3}{8} - \left(\frac{1}{2}\right)^2 = \frac{1}{8}$$

Stąd też odchylenie standardowe jest równe:

$$SD(X) = \sqrt{\frac{3}{8}} = \frac{\sqrt{6}}{4}$$

Wartość Minimalna to 0, a maksymalna 1.

Wszystkie te wartości obliczone za pomocą kodu:

```
1 Miary <- function(Dane, t_params){
2   Miary_Statystyczne <- c("Mediana", "Srednia", "Dominanta", "
   Kurtoza", "Wariancja", "Odchylenie standardowe", "Rozstep")
3   Wyniki_Z_Generatora <- c(median(Dane), mean(Dane), density(Dane)$
   x[which.max(density(Dane)$y)],
4                               kurtosis(Dane), var(Dane), std(Dane),
   max(Dane)-min(Dane))
5   Wyniki_Teoretyczne <- t_params
6   tabela <- data.frame(Miary_Statystyczne, Wyniki_Z_Generatora,
   Wyniki_Teoretyczne)
7   print(summary(Dane))
8   print(tabela)
9 }
10 Miary(Dane_Arcsin1, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
11 Miary(Dane_Arcsin2, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
12 Miary(Dane_Arcsin3, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
```

	Miary_Statystyczne	Wyniki_Z_Generatora	Wyniki_Teoretyczne
1	Mediana	0.46973326	0.5
2	Srednia	0.48596373	0.5
3	Dominanta	0.05484375	{0,1}
4	Kurtoza	1.45471078	1.5
5	Wariancja	0.13137251	0.125
6	Odchylenie standardowe	0.36245346	0.353553390593274
7	Rozstep	0.99999599	1

Rysunek 2: Zestawienie wyników miar generatora dla n=100

Wraz ze wzrostem liczebności próbki miary statystyczne zbliżają się do wartości teoretycznych.

	Miary_Statystyczne	Wyniki_Z_Generatora	Wyniki_Teoretyczne
1	Mediana	0.46973326	0.5
2	Srednia	0.48596373	0.5
3	Dominanta	0.05484375	{0,1}
4	Kurtoza	1.45471078	1.5
5	Wariancja	0.13137251	0.125
6	Odchylenie standardowe	0.36245346	0.353553390593274
7	Rozstep	0.99999599	1

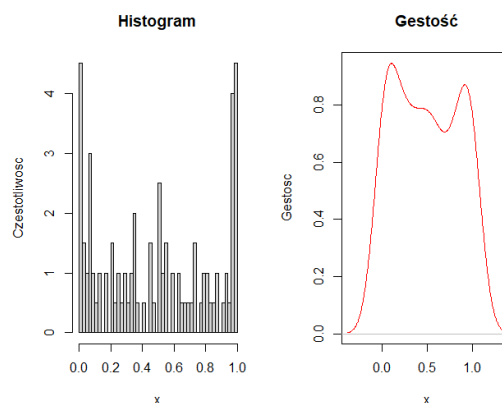
Rysunek 3: Zestawienie wyników miar generatora dla  $n=1000$

	Miary_Statystyczne	Wyniki_Z_Generatora	Wyniki_Teoretyczne
1	Mediana	0.50022484	0.5
2	Srednia	0.49970459	0.5
3	Dominanta	0.02560646	{0,1}
4	Kurtoza	1.49747065	1.5
5	Wariancja	0.12540647	0.125
6	Odchylenie standardowe	0.35412776	0.353553390593274
7	Rozstep	0.99999999	1

Rysunek 4: Zestawienie wyników miar generatora dla  $n=10000$

#### 1.1.4 1.4 Histogram i wykres gęstości

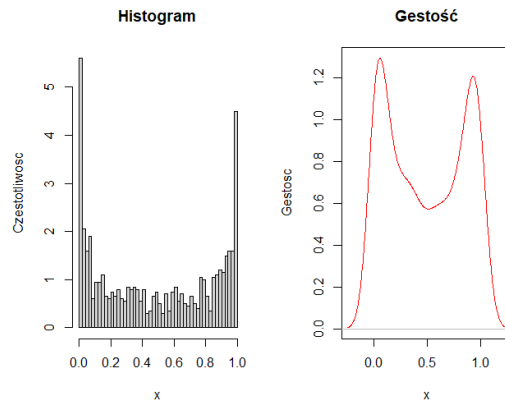
Wykreślić na jednym wykresie: histogram (wykres częstości wystąpień) i wykres gęstości rozkładu teoretycznego (wykres mas prawdopodobieństwa).



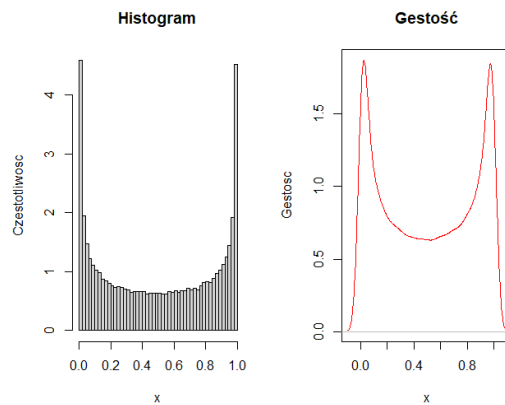
Rysunek 5: Histogram dla  $n=100$

Na histogramie z Rysunku 3 częstość bliska jest 0 i 1. Próbkę jednak jest zbyt małą aby histogram dopasował się do gęstości teoretycznej. Lepsze dopasowanie można zauważyć na Rysunku 4 dla próbki o liczebności 1000. Niemalże idealne dopasowanie zauważyć można na Rysunku 5.

Wniosek: Wraz ze wzrostem liczebności próbki dopasowanie do rozkładu teoretycznego jest coraz lepsze.



Rysunek 6: Histogram dla  $n=1000$



Rysunek 7: Histogram dla  $n=100000$

### 1.1.5 1.5 Dystrybuanta empiryczna i dystrybuanta wybranego rozkładu

Wykreślić na jednym wykresie: dystrybuantę empiryczną i dystrybuantę wybranego rozkładu

```

1 dyst_emp_teor <- function(d, d_inv, n, xlims, umin=0, umax=1, tmin
  =0, tmax=1) {
2   plot(ecdf(d_inv(runif(n,umin,umax))),do.points=FALSE,
3     main=paste("Dystrybuanta empiryczna i zgodna z rozkładem"),
4     ylab = "", xlim = xlims)
5
6   x <- seq(tmin,tmax,0.01)
7   lines(x,d(x),type = "l",col="red")
8   legend("bottomright",inset=.02, legend=c("F(x)-Dyst. emp", "F(x)
  -Dyst rozkl"),
9     col=c("black", "red"),lty=3, cex=0.75,
10    box.lty=1)
11 }

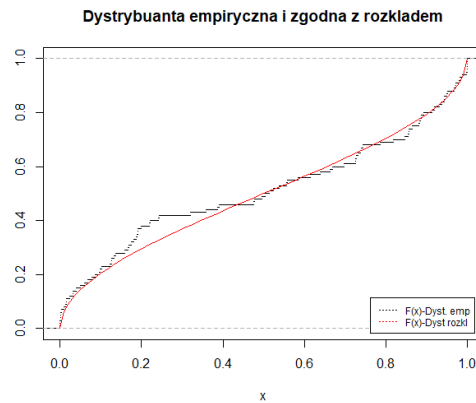
```



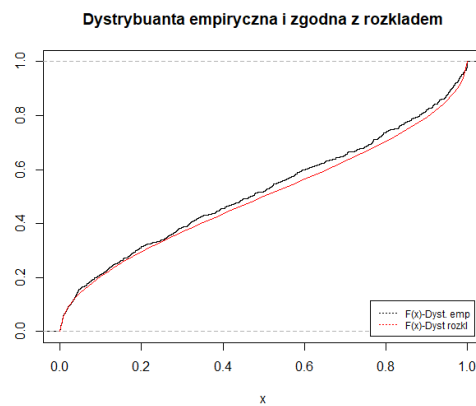
```

12
13 par(mfrow=c(1,1))
14 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^2, xlims = c(0,1))
15 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^3, xlims = c(0,1))
16 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^5, xlims = c(0,1))

```

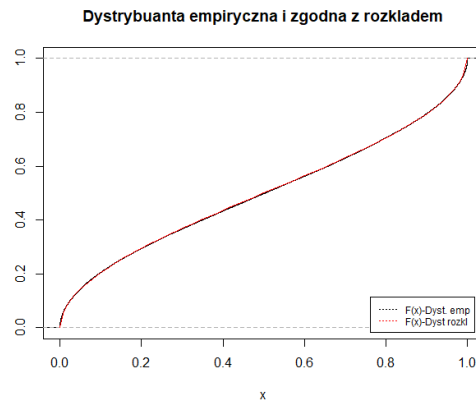


Rysunek 8: Wykres dystrybuanty dla  $n=100$



Rysunek 9: Wykres dystrybuanty dla  $n=1000$

Dla próbki o liczebności 100 dystrybuanta empiryczna nie pokrywa się z dystrybuantą teoretyczną. Wraz ze wzrostem liczebności próbki dopasowanie jest coraz dokładniejsze.



Rysunek 10: Wykres dystrybuenty dla  $n=100000$

Analogicznie do powyższych w kodzie zaimplementowane są wyniki dla:

- rozkładu Cauchyego
- rozkładu wartości ekstremalnych
- rozkładu Reyleigha
- rozkładu Weibulla
- rozkładu Logistycznego
- rozkładu Pareto

## 1.2 Zadanie 2

Dla wygenerowanych próbek z Zadania 1 ciągów liczb pseudolosowych o wybranym rozkładzie wykonujemy kolejne podpunkty.

### 1.2.1 2.1 Normalizacja

Normalizujemy próbki korzystając z metody normalizacji:

$$\text{dane znormalizowane} = \frac{\text{dane} - \text{najmniejsza wartość}}{\text{największa wartość} - \text{najmniejsza wartość}}.$$

W ten sposób otrzymujemy dane w przedziale  $[0; 1]$ . Powyższą metodę implementujemy jako funkcję oraz wykonujemy dla próbek z rozkładu arcusa sinusa oraz Rayleigha.

```
1 normalizacja <- function(x) (x - min(x))/(max(x) - min(x))
2
3 {
4 Dane_Arcsin1_normalizowane <- normalizacja(Dane_Arcsin1)
5 Dane_Arcsin2_normalizowane <- normalizacja(Dane_Arcsin2)
6 Dane_Arcsin3_normalizowane <- normalizacja(Dane_Arcsin3)
7 }
8
9 {
10 Dane_Rayleigh1_normalizowane <- normalizacja(Dane_Rayleigh1)
11 Dane_Rayleigh2_normalizowane <- normalizacja(Dane_Rayleigh2)
12 Dane_Rayleigh3_normalizowane <- normalizacja(Dane_Rayleigh3)
13 }
```

### 1.2.2 2.2 Generowanie próbek korzystając z iteracji mapy chaotycznej

Mapą chaotyczną w zadaniu jest ukośna mapa namiotu. Wyraża się ona wzorem

$$z_{k+1} = f(z_k) \begin{cases} \frac{z_k}{p}, & \text{gdy } z_k \in [0; p), \\ \frac{1-z_k}{1-p}, & \text{gdy } z_k \in [p; 1), \end{cases}$$

gdzie  $p \in (0; 1)$ , a my przyjmujemy  $p = 0.45$ ,  $k = 0, 1, 2, \dots, N$ ,  $N$  to liczba iteracji. Poprzez  $f^N$  oznaczamy  $N$ -te złożenie mapy.

Implementujemy funkcję rozważanej mapy chaotycznej, a także jej  $N$ -te złożenie.

```
1 # funkcje mapy chaotycznej z polecenia
2 ukosna_mapa_namiotu <- function(zk, p=0.45) {
3   if(zk >= 0 & zk < p) {
4     return(zk/p)
5   }definiujemy
6   if(zk >= p & zk <= 1) {
7     return((1-zk)/(1-p))
8   }
9   else{
10    print('Nieprawidlowa wartosc')
11  }
12 }
13 # definiujemy N-te zlozenie mapy chaotycznej z polecenia
14 ukosna_mapa_namiotu_N_iteracji <- function(N, x0, p = 0.45){
15   x = c()
16   for (j in 1:length(x0)) {
17     x[j] = ukosna_mapa_namiotu(x0[j], p)
18
19     if(N > 1) {
20       for(i in 2:N){
21         x[j] = ukosna_mapa_namiotu(x[j], p)
22       }
23     }
24   }
25   return(x)
26 }
```

Dla próbek z rozkładu arcusa sinusa oraz Rayleigha generujemy za pomocą kolejnych złożenia mapy chaotycznej ( $N \in \{1, 5, 12\}$ ) ciągi liczb pseudolosowych.

Dla rozkładu arcusa sinusa

```
1 Dane_Arcsin1_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
   Dane_Arcsin1_normalizowane)
2 Dane_Arcsin2_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
   Dane_Arcsin2_normalizowane)
3 Dane_Arcsin3_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
   Dane_Arcsin3_normalizowane)
4 Dane_Arcsin1_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
   Dane_Arcsin1_normalizowane)
5 Dane_Arcsin2_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
   Dane_Arcsin2_normalizowane)
6 Dane_Arcsin3_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
   Dane_Arcsin3_normalizowane)
7 Dane_Arcsin1_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
   (12, Dane_Arcsin1_normalizowane)
8 Dane_Arcsin2_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
   (12, Dane_Arcsin2_normalizowane)
9 Dane_Arcsin3_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
   (12, Dane_Arcsin3_normalizowane)
```

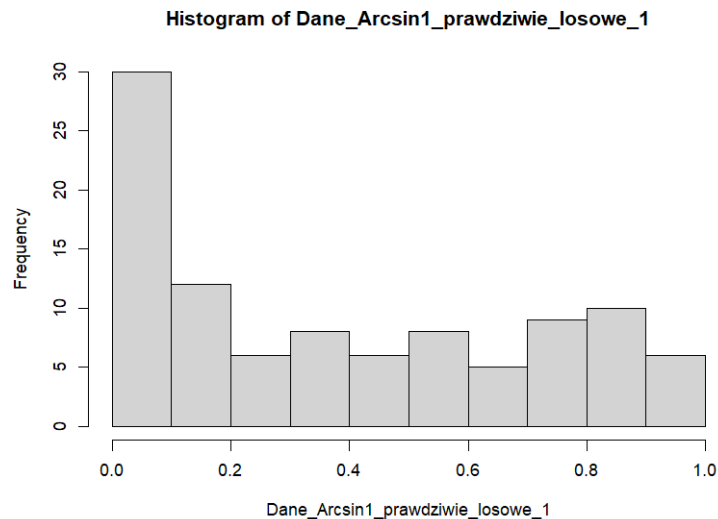
Dla rozkładu Rayleigha

```
1 Dane_Rayleigh1_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji
  (1, Dane_Rayleigh1_normalizowane)
2 Dane_Rayleigh2_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji
  (1, Dane_Rayleigh2_normalizowane)
3 Dane_Rayleigh3_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji
  (1, Dane_Rayleigh3_normalizowane)
4 Dane_Rayleigh1_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji
  (5, Dane_Rayleigh1_normalizowane)
5 Dane_Rayleigh2_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji
  (5, Dane_Rayleigh2_normalizowane)
6 Dane_Rayleigh3_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji
  (5, Dane_Rayleigh3_normalizowane)
7 Dane_Rayleigh1_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_
  iteracji(12, Dane_Rayleigh1_normalizowane)
8 Dane_Rayleigh2_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_
  iteracji(12, Dane_Rayleigh2_normalizowane)
9 Dane_Rayleigh3_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_
  iteracji(12, Dane_Rayleigh3_normalizowane)
```

### 1.2.3 2.3 Histogramy wygenerowanych rozkładów

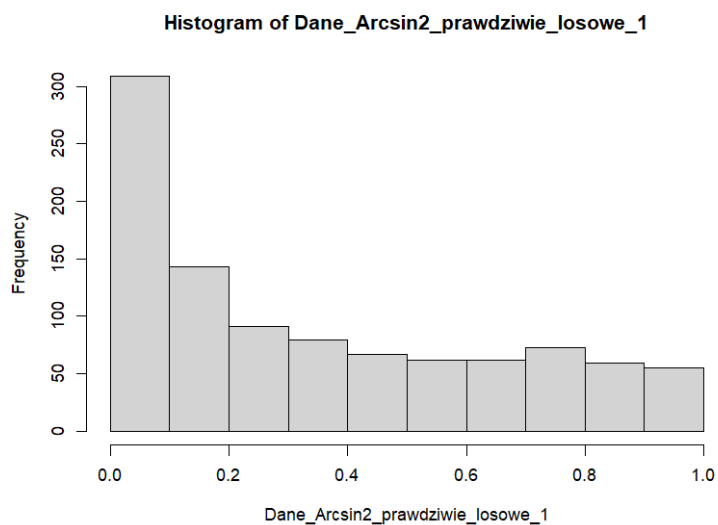
Wykreślamy histogramy wygenerowanych rozkładów.

```
1 hist(Dane_Arcsin1_prawdziwie_losowe_1)
```



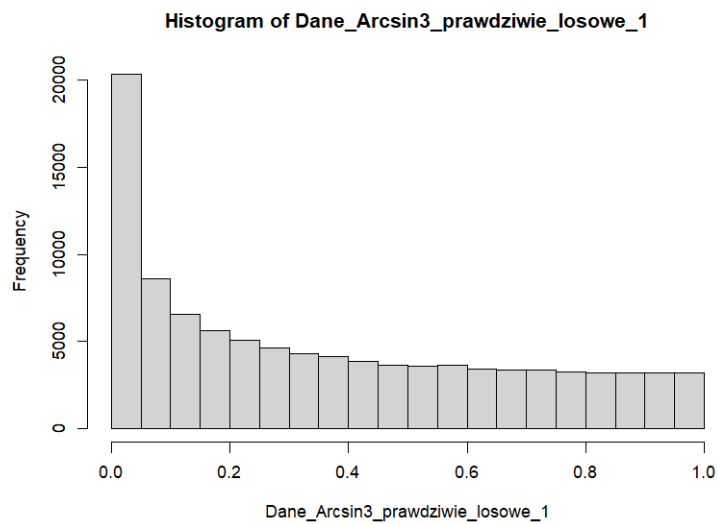
Rysunek 11: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_1)
```



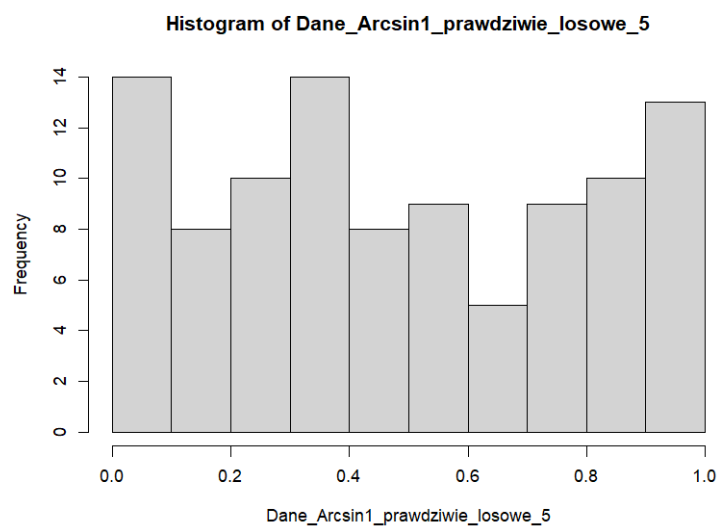
Rysunek 12: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_1)
```



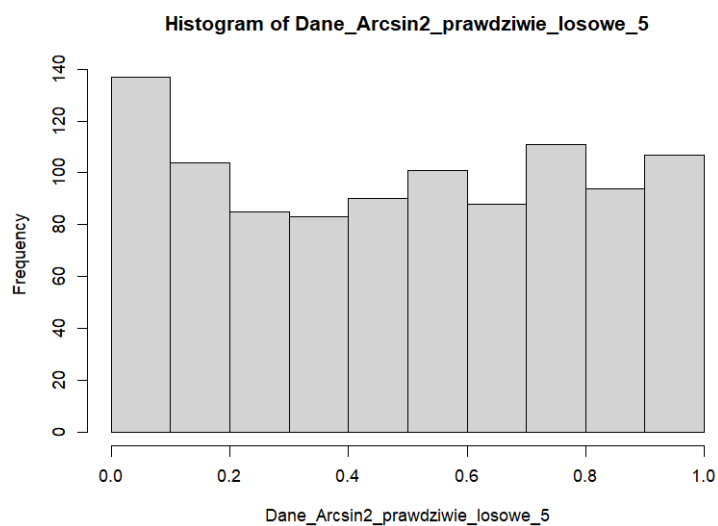
Rysunek 13: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin1_prawdziwie_losowe_5)
```



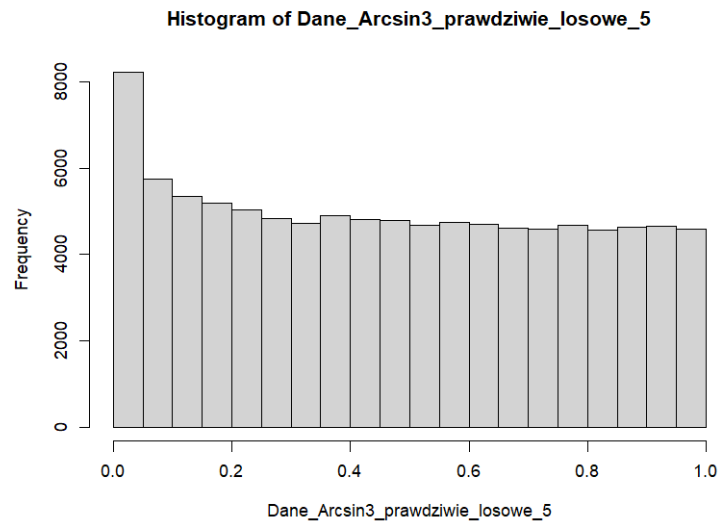
Rysunek 14: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_5)
```



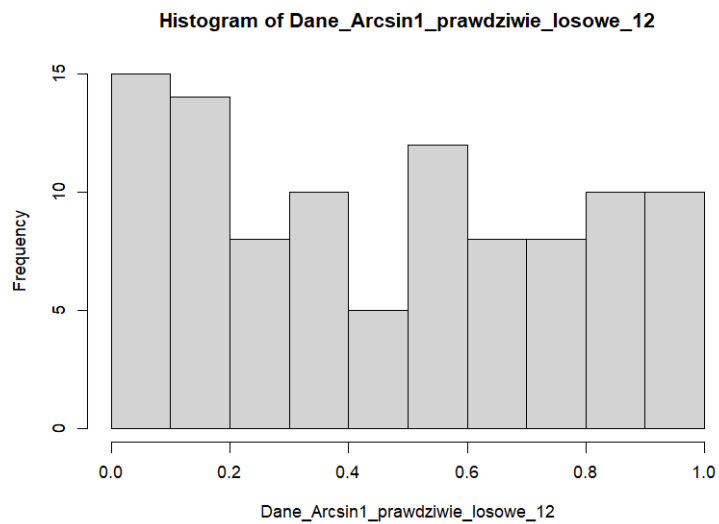
Rysunek 15: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_5)
```



Rysunek 16: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

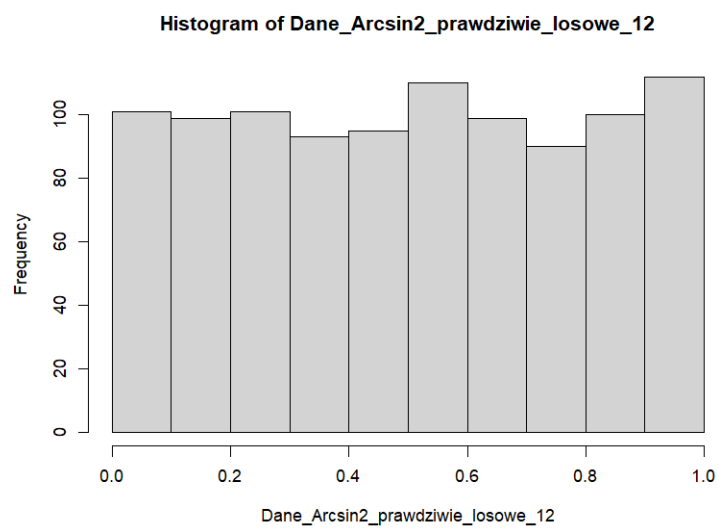
```
1 hist(Dane_Arcsin1_prawdziwie_losowe_12)
```



Rysunek 17: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

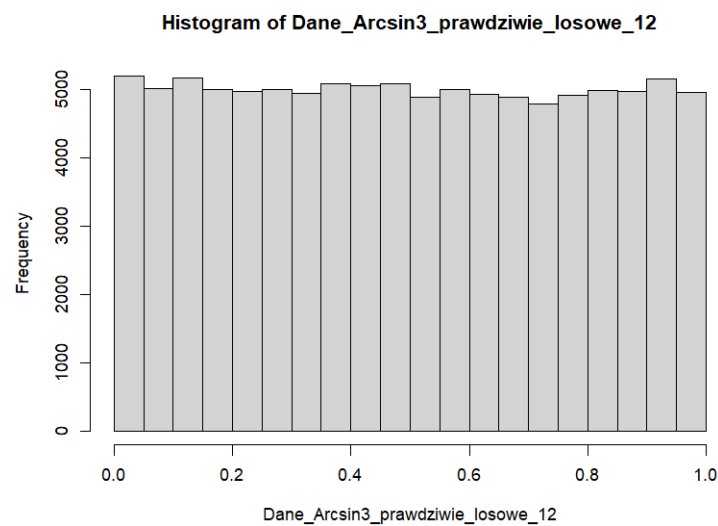


```
1 hist(Dane_Arcsin2_prawdziwie_losowe_12)
```



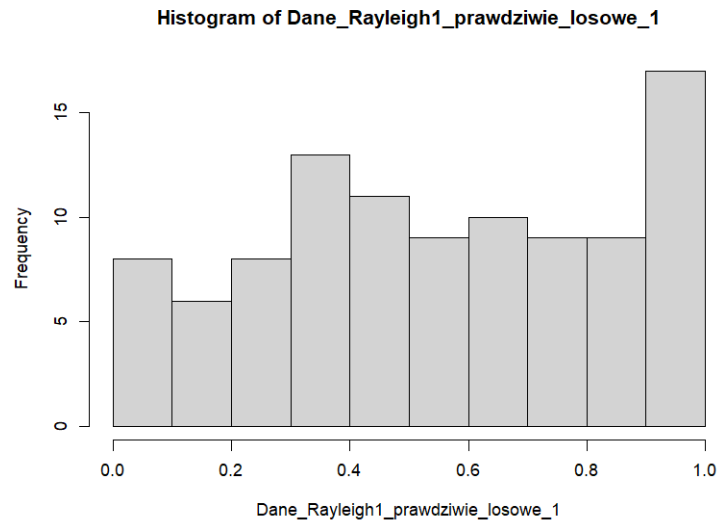
Rysunek 18: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_12)
```



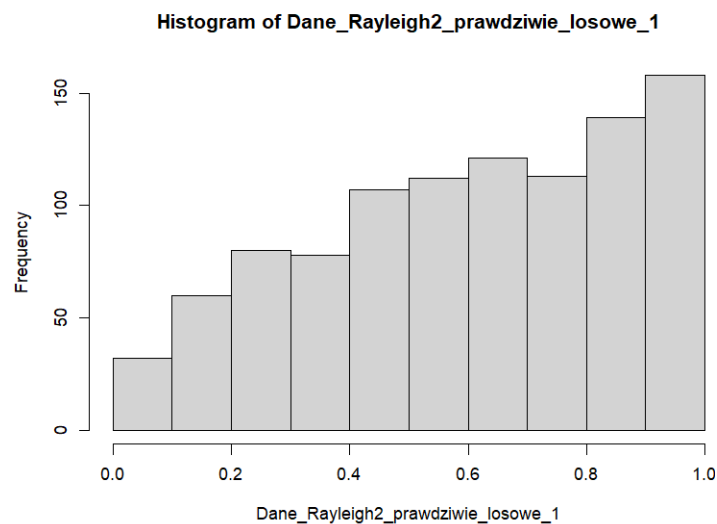
Rysunek 19: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_1)
```



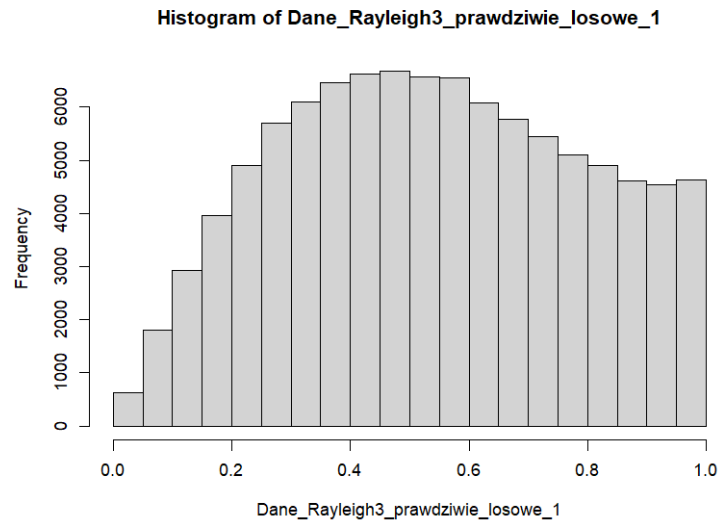
Rysunek 20: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_1)
```



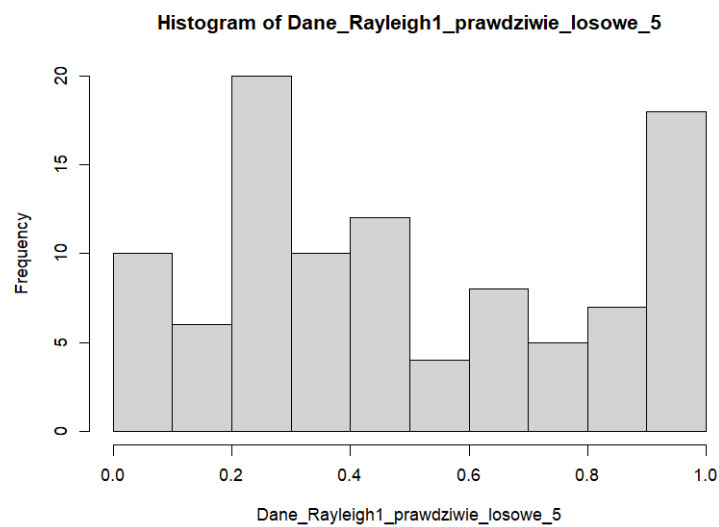
Rysunek 21: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_1)
```



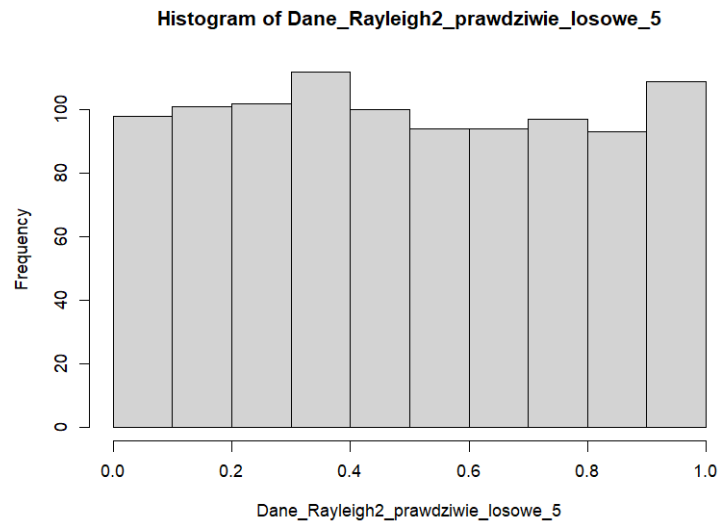
Rysunek 22: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_5)
```



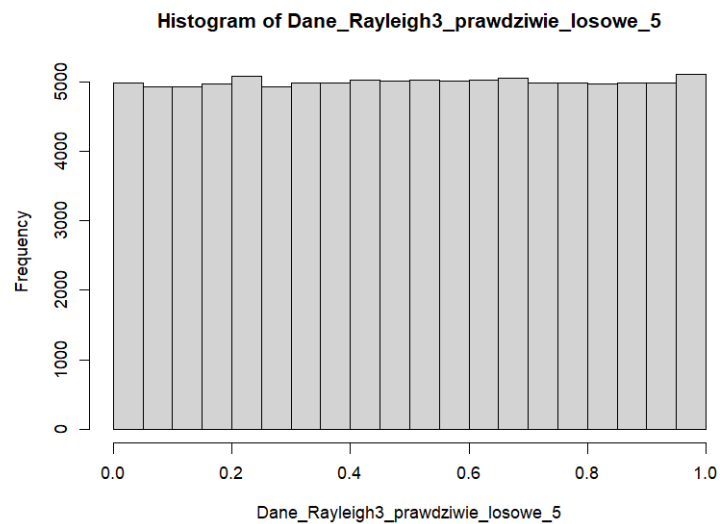
Rysunek 23: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_5)
```



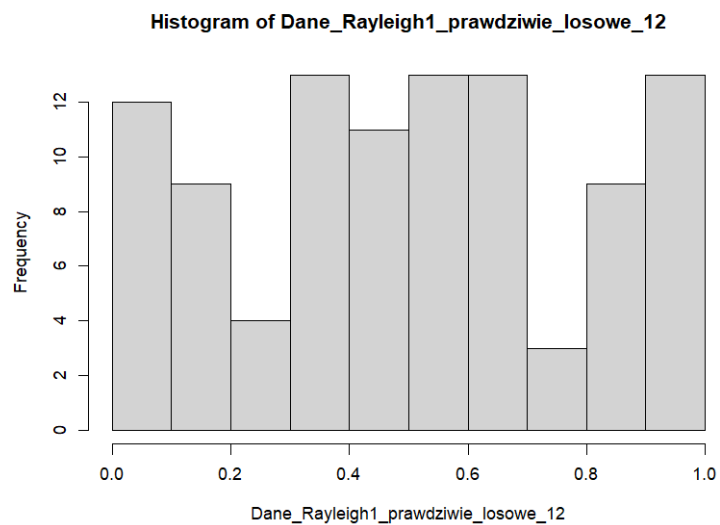
Rysunek 24: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_5)
```



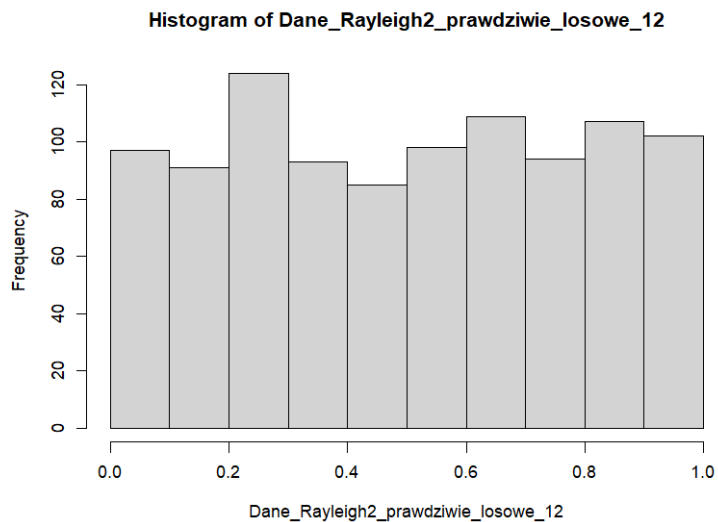
Rysunek 25: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_12)
```



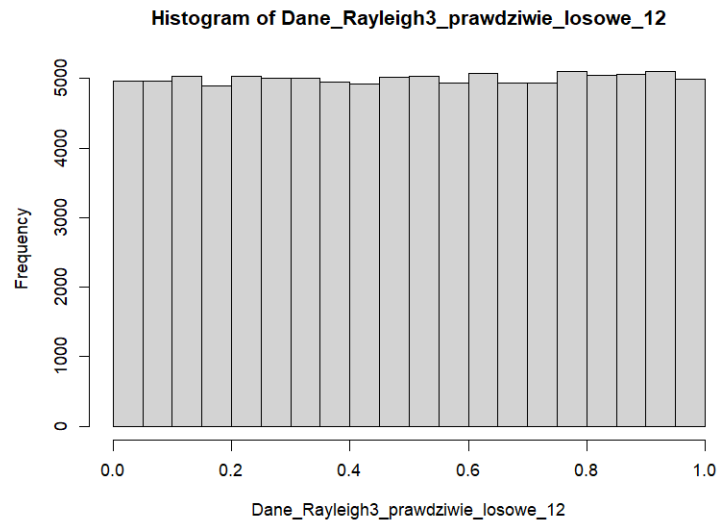
Rysunek 26: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_12)
```



Rysunek 27: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_12)
```



Rysunek 28: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

Na podstawie powyższych wykresów możemy wywnioskować:

- Im większa liczebność próbki tym mniejsze skoki pomiędzy wartościami
- Im większa liczba złożań mapy chaotycznej zastosowana do danych tym histogram (a tym samym rozkład prawdopodobieństwa) jest bardziej zbliżony do rozkładu jednostajnego.

Zatem korzystając ze złożań map chaotycznych uzyskujemy próbkę losową z rozkładu jednostajnego.

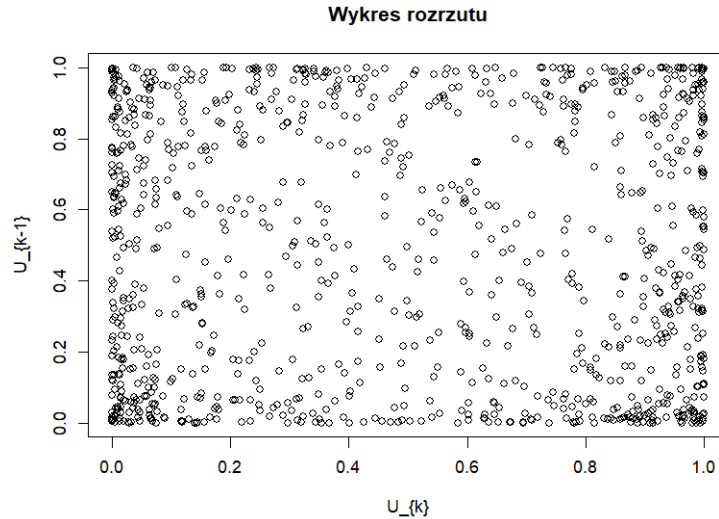
#### 1.2.4 2.4 Rozrzut punktów

Generujemy wykresy zależności par elementów z uzyskanych próbek  $(u_k, u_{k+1})$ . Implementujemy funkcję generującą wykresy.

```
1 wykres_rozrzutu <- function(df) plot(x = df[-length(df)], y=df[-1],
  xlab='U_{k}', ylab = 'U_{k-1}', main = 'Wykres rozrzutu')
```

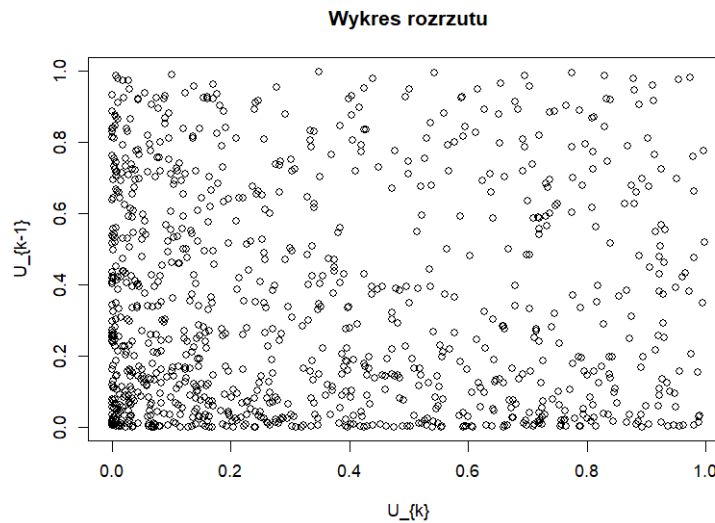
W naszych wykresach ograniczamy się tylko do próbek rozmiaru  $10^3$  dlatego, że dla  $10^2$  nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla  $10^5$  nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
1 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



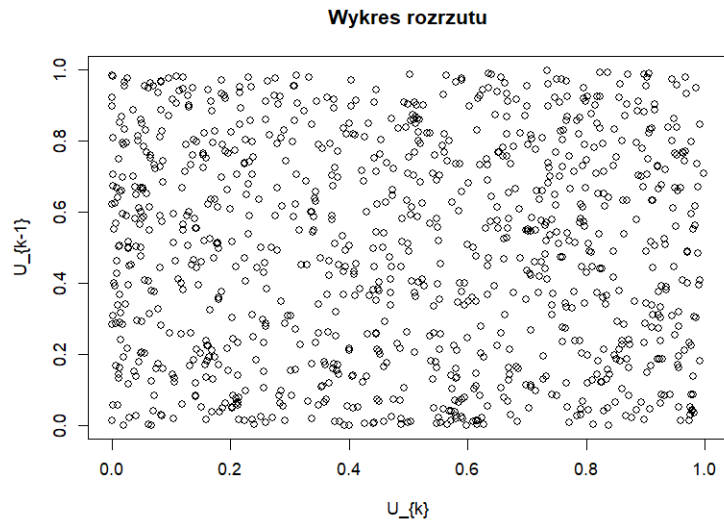
Rysunek 29: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1)
```



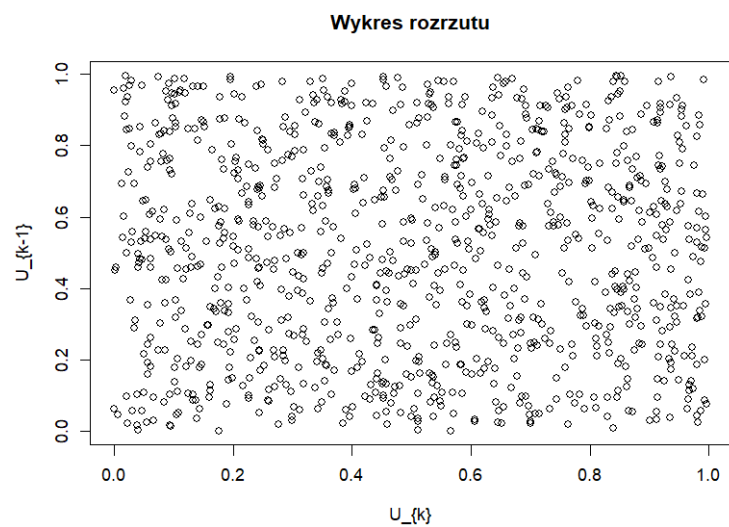
Rysunek 30: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
```



Rysunek 31: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinusów i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

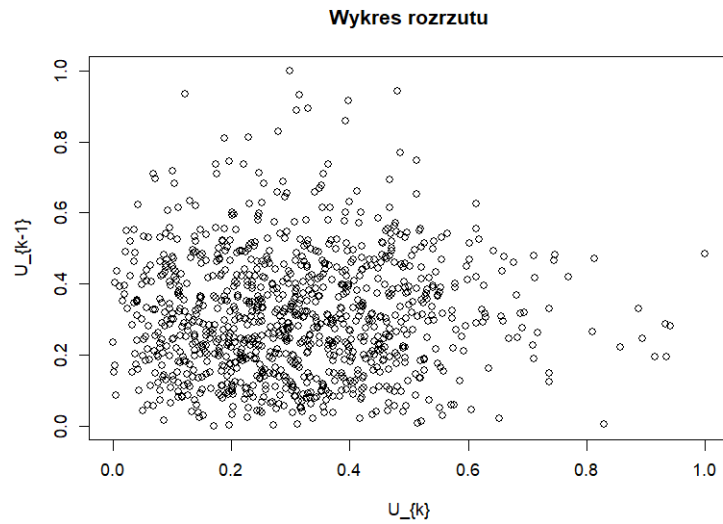
```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12)
```



Rysunek 32: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinusów i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

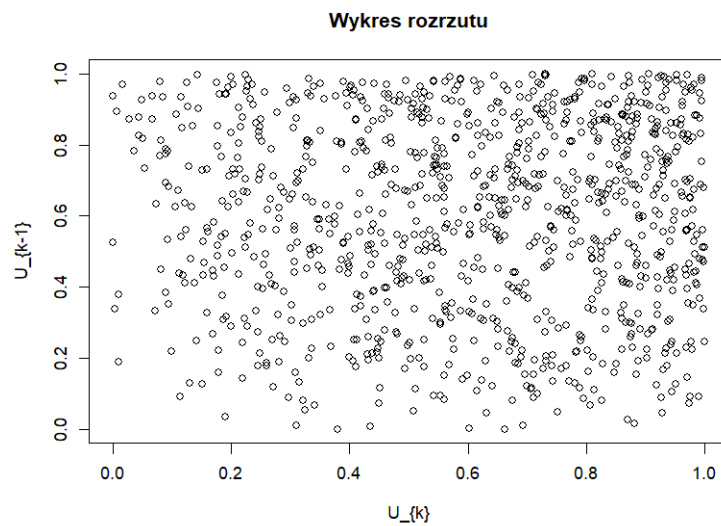


```
1 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



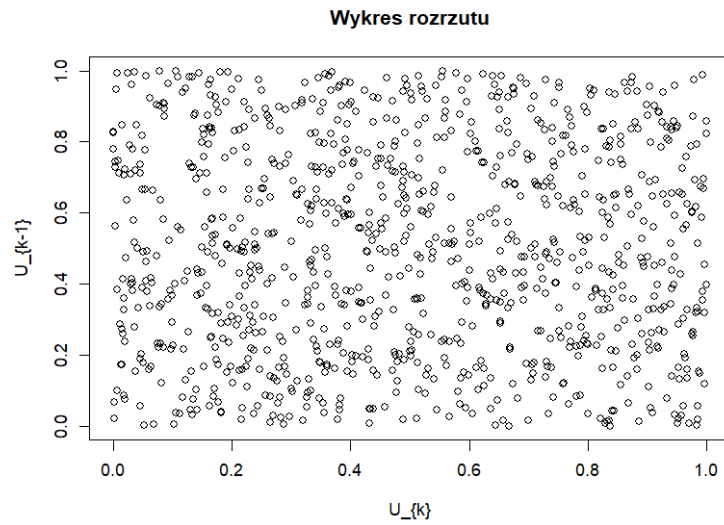
Rysunek 33: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1)
```



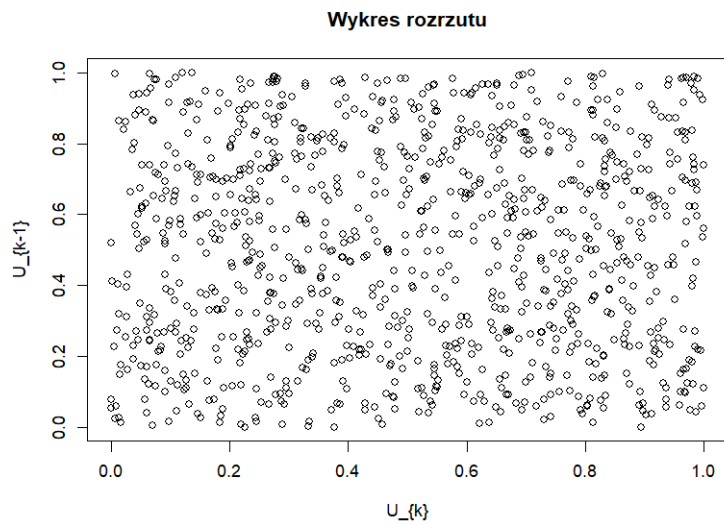
Rysunek 34: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5)
```



Rysunek 35: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12)
```



Rysunek 36: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

### 1.2.5 2.5 Zbliżenie do rozkładu jednorodnego

Obserwując powyższe wykresy możemy zauważyć, że w obu przypadkach (obu rozkładach) zbliżenie do rozkładu jednorodnego następuje dla **pięciokrotnego** złożenia mapy chaotycznej.

## 1.3 Zadanie 3

Podpunkty z zadania 2 wykonujemy dla dwóch nowych map.

Mapa logistyczna:

$$x_{n+1} = ax_n(1 - x_n).$$

Mapa kwadratowa:

$$x_{n+1} = b - ax_n^2.$$

Implementacja mapy logistycznej i kwadratowej.

```
1 # definiujemy funkcje mapy logistycznej
2 mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))
3
4
5 # definiujemy N-te zlozenie mapy chaotycznej
6 mapa_logistyczna_N_iteracji <- function(N, x0, r=3.8){
7   x = c()
8   for (j in 1:length(x0)) {
9
10     x[j] = mapa_logistyczna(x0[j], r)
11     if(N > 1){
12       for(i in 2:N){
13         x[j] = mapa_logistyczna(x[j], r)
14       }
15     }
16   }
17   return(x)
18 }
19
20
21 # definiujemy funkcje mapy kwadratowej
22 mapa_kwadratowa <- function(xk, a=4, b=0.5) return(b - a*xk^2)
23
24
25 # definiujemy N-te zlozenie mapy chaotycznej
26 mapa_kwadratowa_N_iteracji <- function(N, x0, a=4, b=0.5){
27   x = c()
28   for (j in 1:length(x0)) {
29
30     x[j] = mapa_kwadratowa(x0[j], a, b)
31     if(N>1){
32       for(i in 2:N){
33         x[j] = mapa_kwadratowa(x[j], a, b)
34       }
35     }
36   }
37   return(x)
38 }
```

### 1.3.1 3.1 Normalizacja

Dane znormalizowane mamy już z podpunktu 2.1

#### Mapa logistyczna

### 1.3.2 3.2 Generowanie próbek korzystając z iteracji mapy chaotycznej

Dla próbek z rozkładu arcusa sinusa oraz Rayleigha generujemy za pomocą kolejnych złożań mapy chaotycznej ( $N \in \{1, 5, 12\}$ ) ciągi liczb pseudolosowych.

Dla rozkładu arcusa sinusa

```
1 Dane_Arcsin1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
   _iteracji(1, Dane_Arcsin1_normalizowane)
2 Dane_Arcsin2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
   _iteracji(1, Dane_Arcsin2_normalizowane)
3 Dane_Arcsin3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
   _iteracji(1, Dane_Arcsin3_normalizowane)
4 Dane_Arcsin1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
   _iteracji(5, Dane_Arcsin1_normalizowane)
5 Dane_Arcsin2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
   _iteracji(5, Dane_Arcsin2_normalizowane)
6 Dane_Arcsin3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
   _iteracji(5, Dane_Arcsin3_normalizowane)
7 Dane_Arcsin1_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
   N_iteracji(12, Dane_Arcsin1_normalizowane)
8 Dane_Arcsin2_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
   N_iteracji(12, Dane_Arcsin2_normalizowane)
9 Dane_Arcsin3_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
   N_iteracji(12, Dane_Arcsin3_normalizowane)
10 }
```

Dla rozkładu Rayleigha

```

1 Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
  _N_iteracji(1, Dane_Rayleigh1_normalizowane)
2 Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
  _N_iteracji(1, Dane_Rayleigh2_normalizowane)
3 Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
  _N_iteracji(1, Dane_Rayleigh3_normalizowane)
4 Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
  _N_iteracji(5, Dane_Rayleigh1_normalizowane)
5 Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
  _N_iteracji(5, Dane_Rayleigh2_normalizowane)
6 Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
  _N_iteracji(5, Dane_Rayleigh3_normalizowane)
7 Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna <- mapa_
  logistyczna_N_iteracji(12, Dane_Rayleigh1_normalizowane)
8 Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna <- mapa_
  logistyczna_N_iteracji(12, Dane_Rayleigh2_normalizowane)
9 Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna <- mapa_
  logistyczna_N_iteracji(12, Dane_Rayleigh3_normalizowane)

```

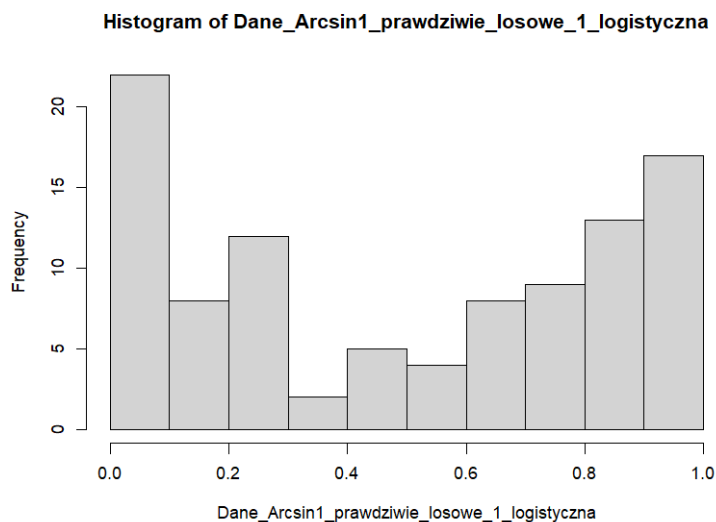
### 1.3.3 3.3 Histogramy wygenerowanych rozkładów

Wykreślamy histogramy wygenerowanych rozkładów.

```

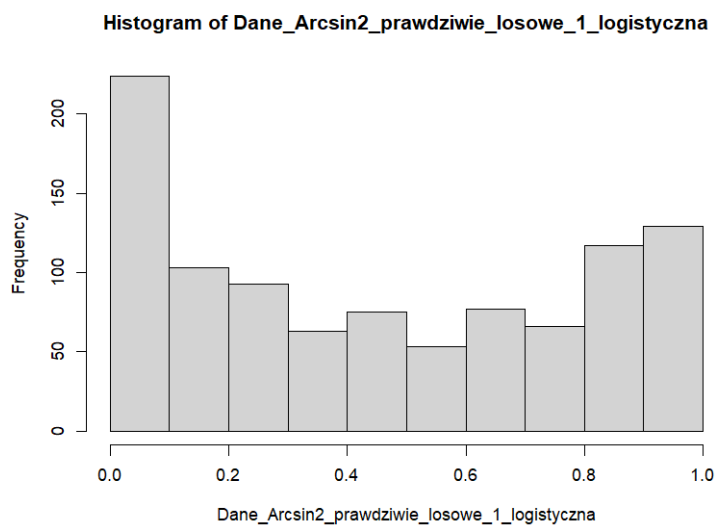
1 hist(Dane_Arcsin1_prawdziwie_losowe_1_logistyczna)

```



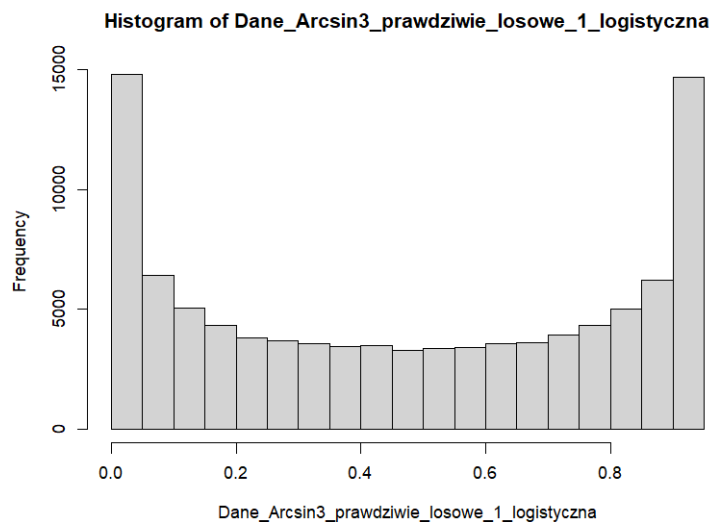
Rysunek 37: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
```



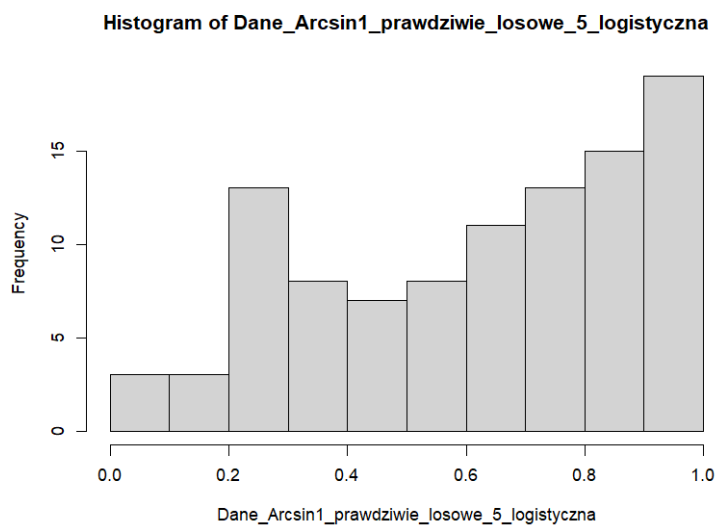
Rysunek 38: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_1_logistyczna)
```



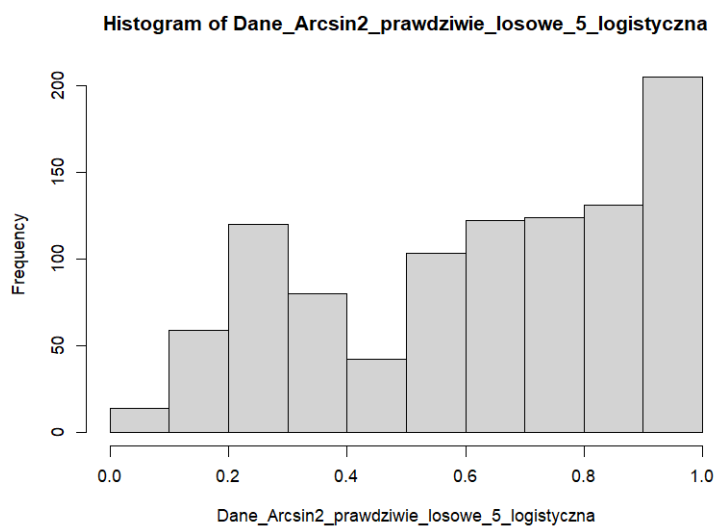
Rysunek 39: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin1_prawdziwie_losowe_5_logistyczna)
```



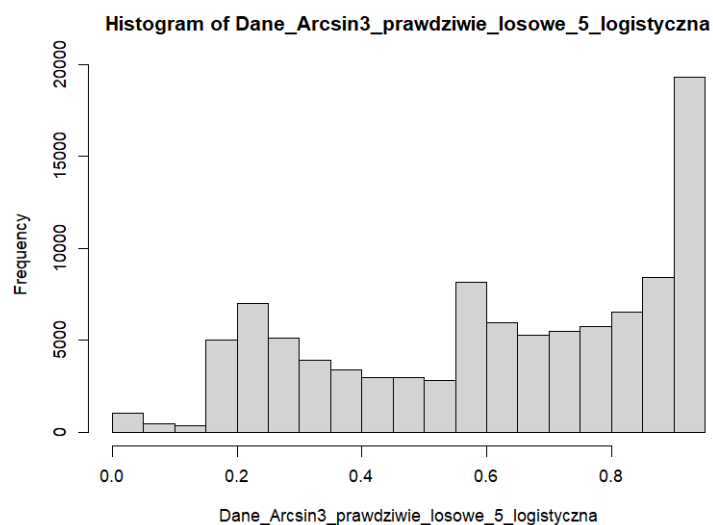
Rysunek 40: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
```



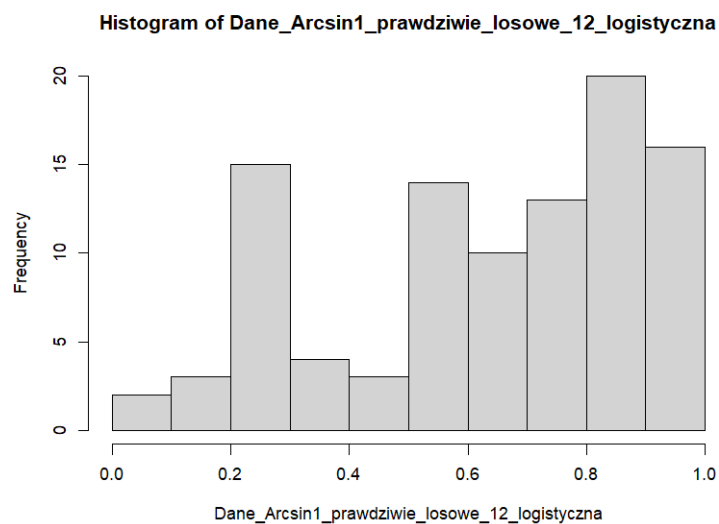
Rysunek 41: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_5_logistyczna)
```



Rysunek 42: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

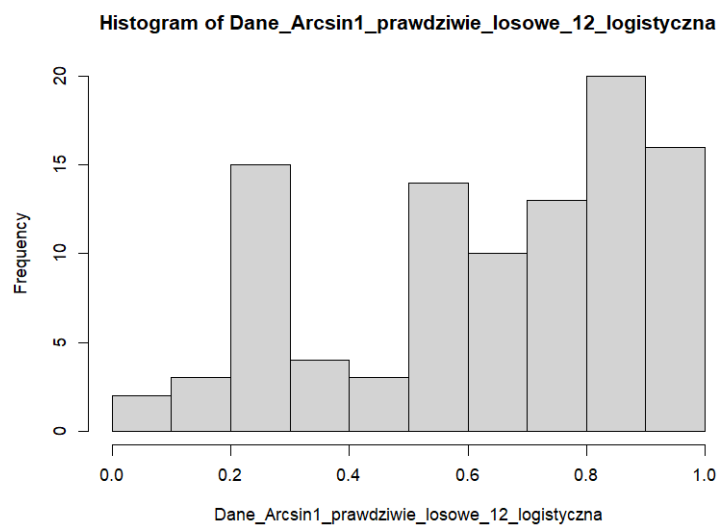
```
1 hist(Dane_Arcsin1_prawdziwie_losowe_12_logistyczna)
```



Rysunek 43: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

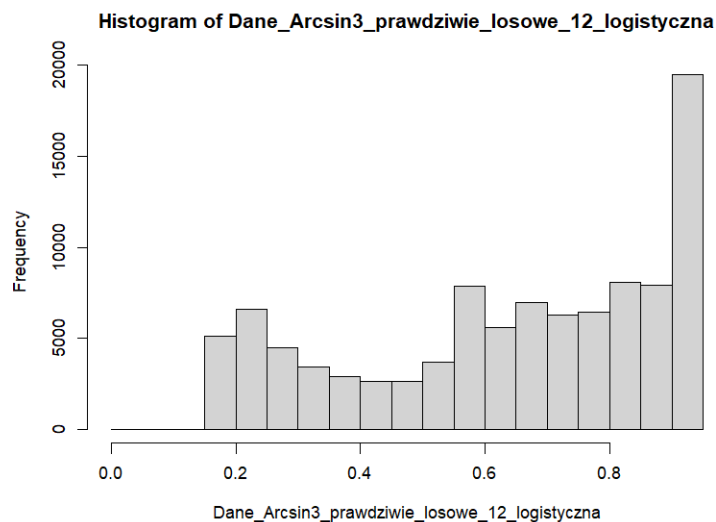


```
1 hist(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
```



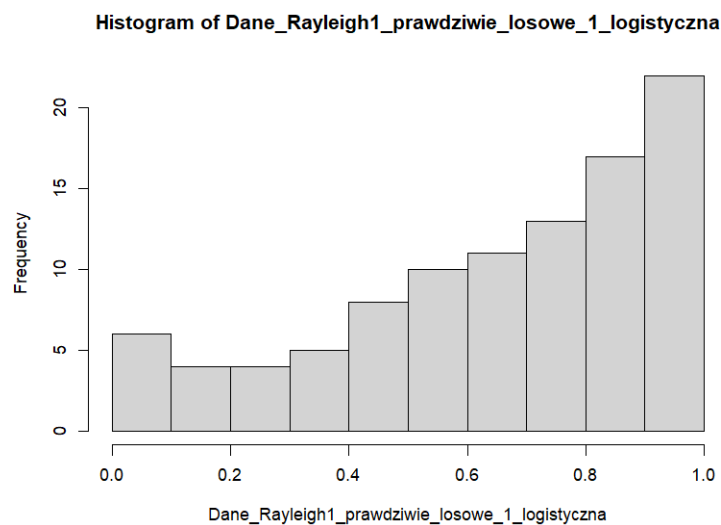
Rysunek 44: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_12_logistyczna)
```



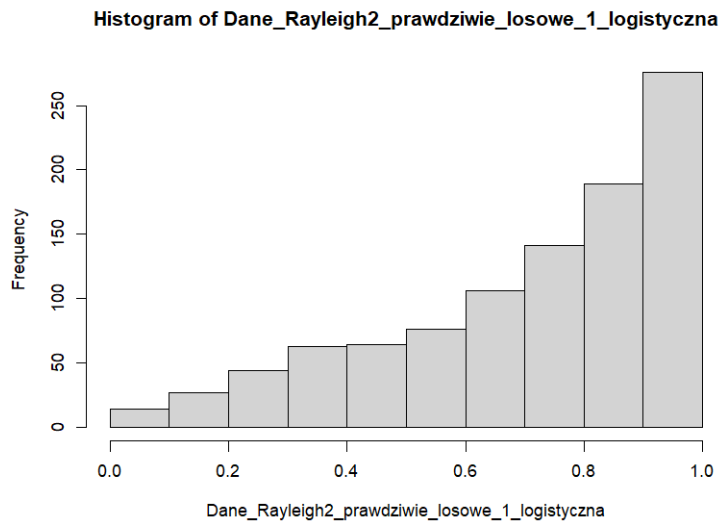
Rysunek 45: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna)
```



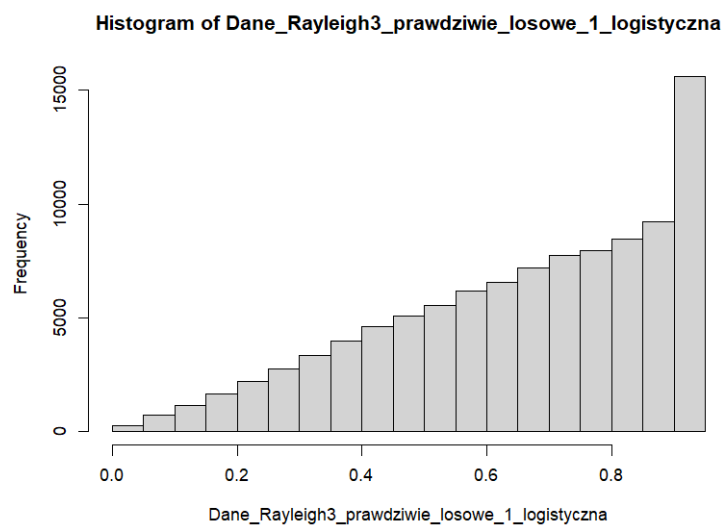
Rysunek 46: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
```



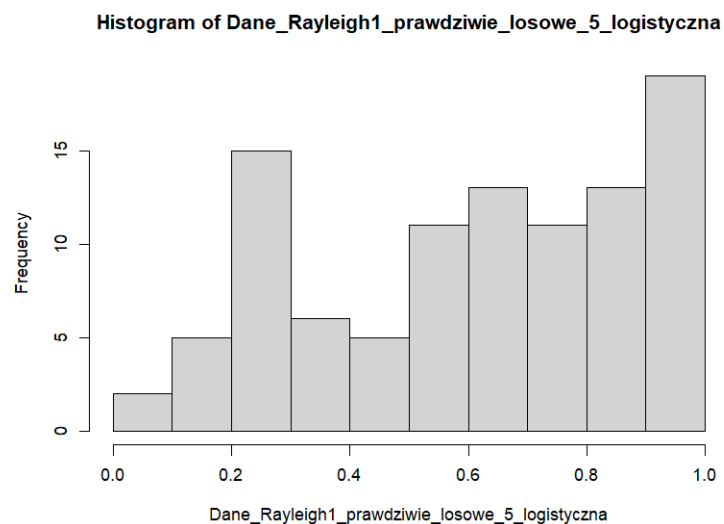
Rysunek 47: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna)
```



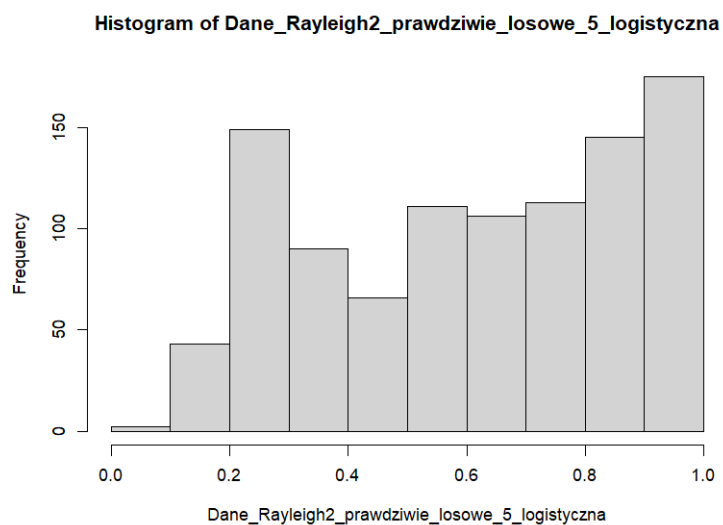
Rysunek 48: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna)
```



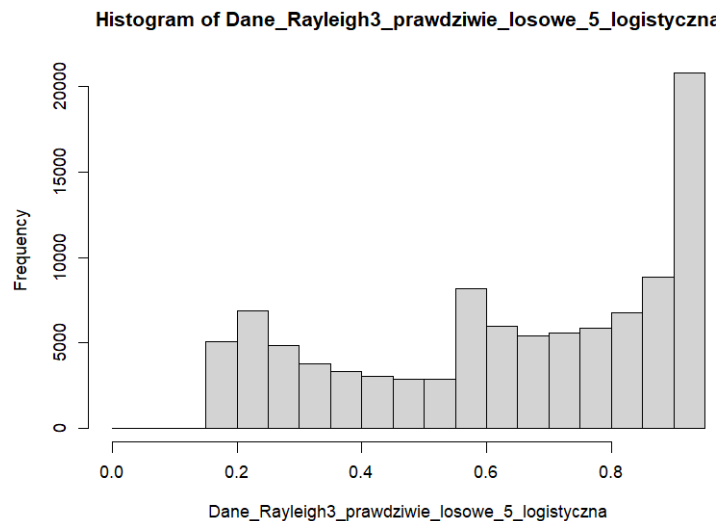
Rysunek 49: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
```



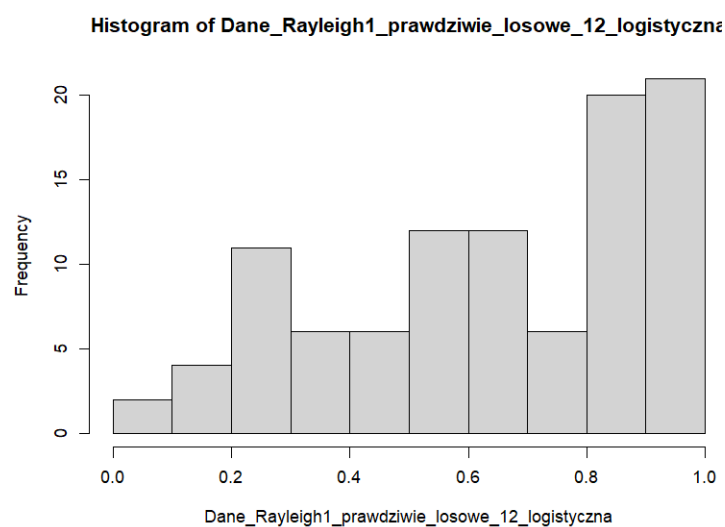
Rysunek 50: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna)
```



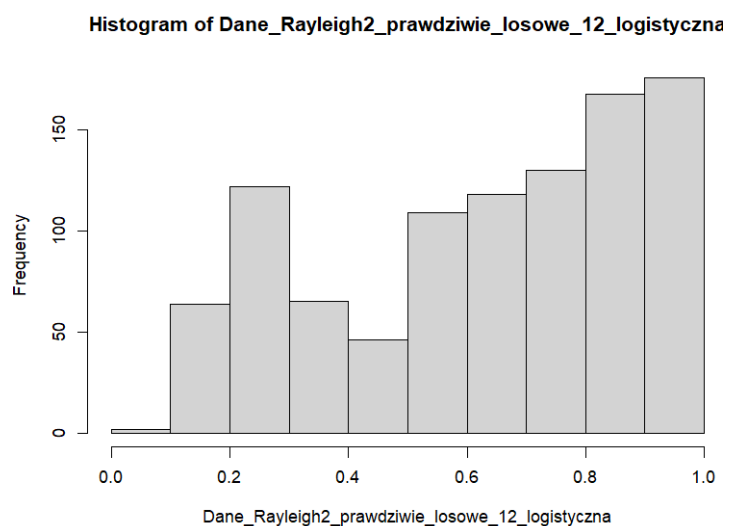
Rysunek 51: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna)
```



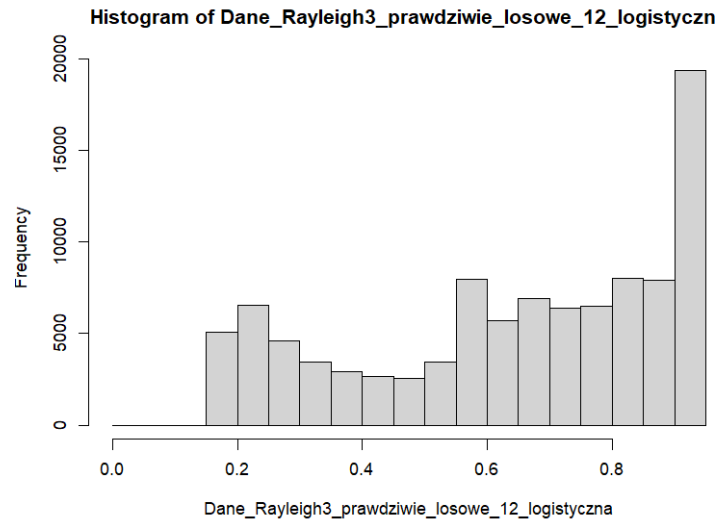
Rysunek 52: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
```



Rysunek 53: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna)
```



Rysunek 54: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

Na podstawie powyższych wykresów możemy wywnioskować:

- Im większa liczebność próbki tym mniejsze skoki pomiędzy wartościami

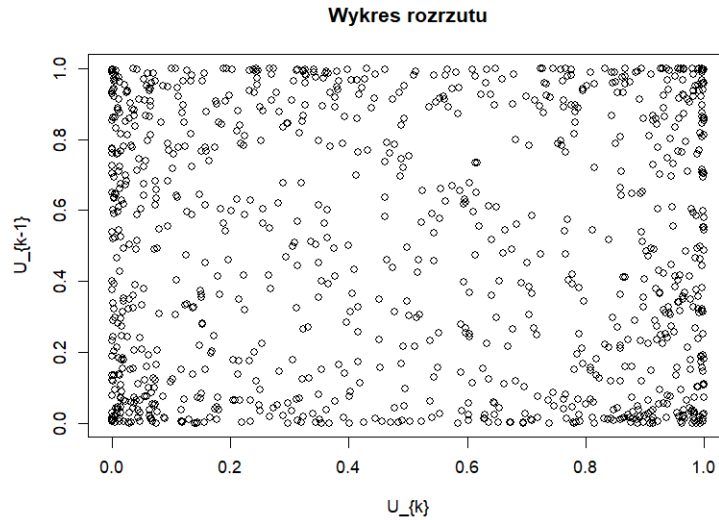
Z mapy logistycznej nie otrzymujemy rozkładu jednostajnego.

### 1.3.4 3.4 Rozrzut punktów

Generujemy wykresy zależności par elementów z uzyskanych próbek  $(u_k, u_{k+1})$ . Korzystając z zaimplementowanej funkcji generującej wykresy.

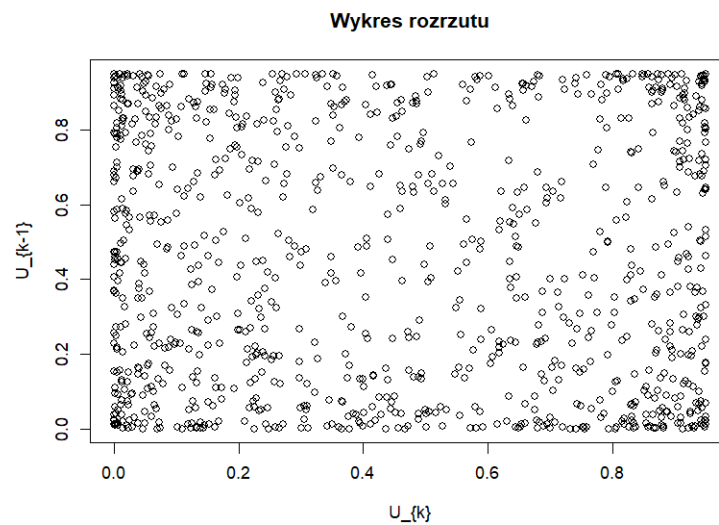
W naszych wykresach ograniczamy się tylko do próbek rozmiaru  $10^3$  dlatego, że dla  $10^2$  nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla  $10^5$  nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
1 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



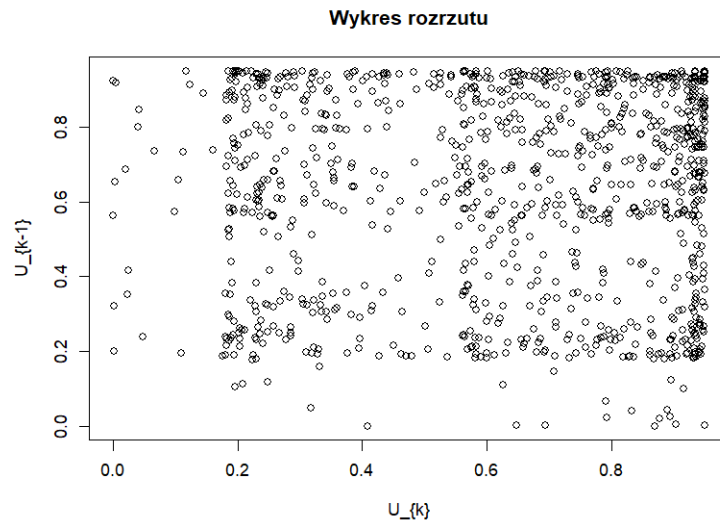
Rysunek 55: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
```



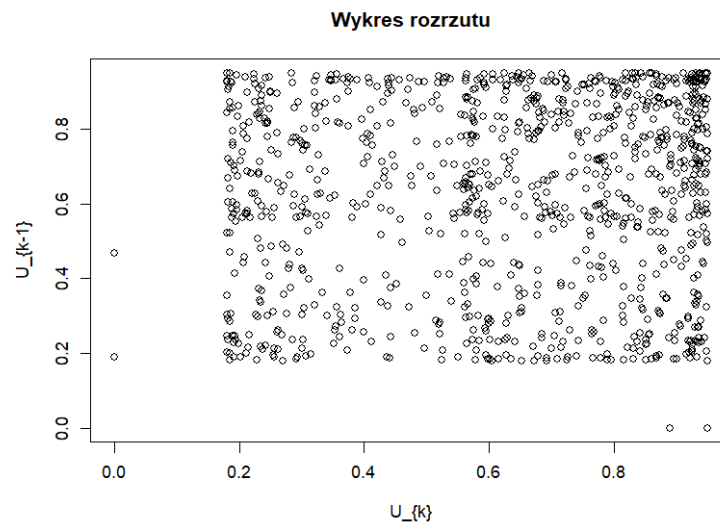
Rysunek 56: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
```



Rysunek 57: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

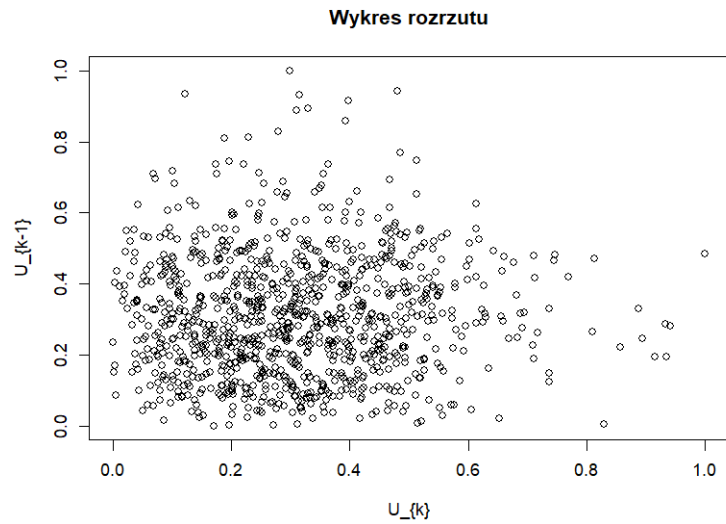
```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
```



Rysunek 58: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

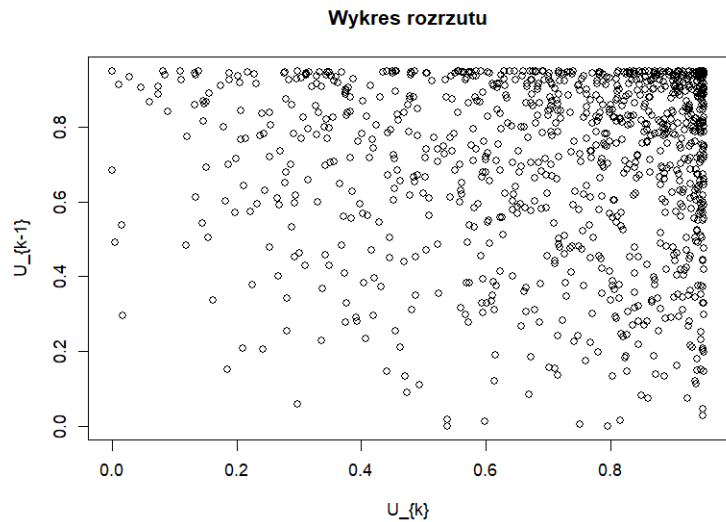


```
1 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



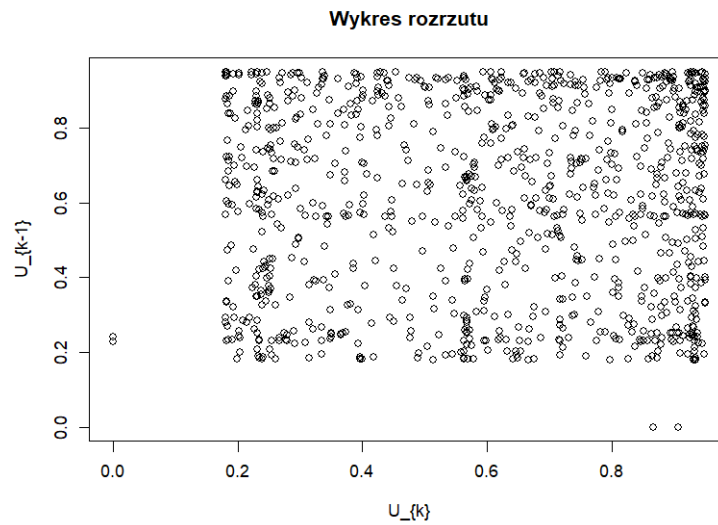
Rysunek 59: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
```



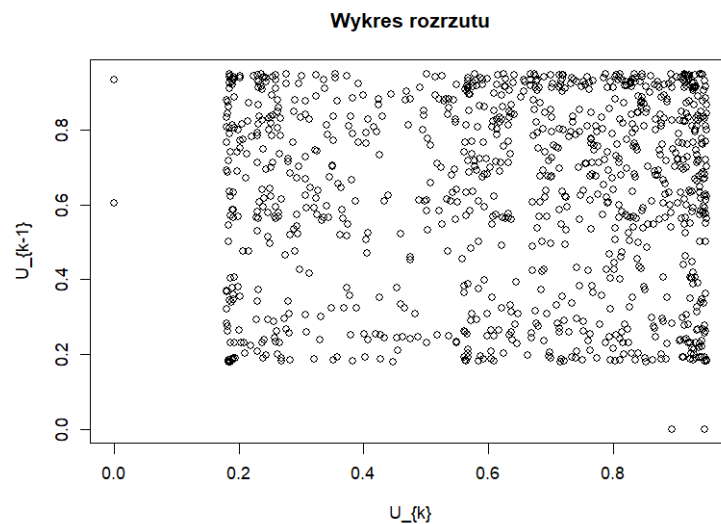
Rysunek 60: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
```



Rysunek 61: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
```



Rysunek 62: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

Widać, że dla większych iteracji uzyskujemy wartości skupione przy "bokach" obszaru  $[0.2; 1] \times [0.2; 1]$ .

## Mapa kwadratowa

### 1.3.5 3.2 Generowanie próbek korzystając z iteracji mapy chaotycznej

Dla próbek z rozkładu arcusa sinusa oraz Rayleigha generujemy za pomocą kolejnych złożań mapy chaotycznej ( $N \in \{1, 5, 12\}$ ) ciągi liczb pseudolosowych.

Dla rozkładu arcusa sinusa

```
1 Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_
   iteracji(1, Dane_Arcsin1_normalizowane)
2 Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_
   iteracji(1, Dane_Arcsin2_normalizowane)
3 Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_
   iteracji(1, Dane_Arcsin3_normalizowane)
4 Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_
   iteracji(5, Dane_Arcsin1_normalizowane)
5 Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_
   iteracji(5, Dane_Arcsin2_normalizowane)
6 Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_
   iteracji(5, Dane_Arcsin3_normalizowane)
7 Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_
   iteracji(12, Dane_Arcsin1_normalizowane)
8 Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_
   iteracji(12, Dane_Arcsin2_normalizowane)
9 Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_
   iteracji(12, Dane_Arcsin3_normalizowane)
```

Dla rozkładu Rayleigha

```

1 Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
   _iteracji(1, Dane_Rayleigh1_normalizowane)
2 Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
   _iteracji(1, Dane_Rayleigh2_normalizowane)
3 Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
   _iteracji(1, Dane_Rayleigh3_normalizowane)
4 Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
   _iteracji(5, Dane_Rayleigh1_normalizowane)
5 Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
   _iteracji(5, Dane_Rayleigh2_normalizowane)
6 Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
   _iteracji(5, Dane_Rayleigh3_normalizowane)
7 Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N
   _iteracji(12, Dane_Rayleigh1_normalizowane)
8 Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N
   _iteracji(12, Dane_Rayleigh2_normalizowane)
9 Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N
   _iteracji(12, Dane_Rayleigh3_normalizowane)

```

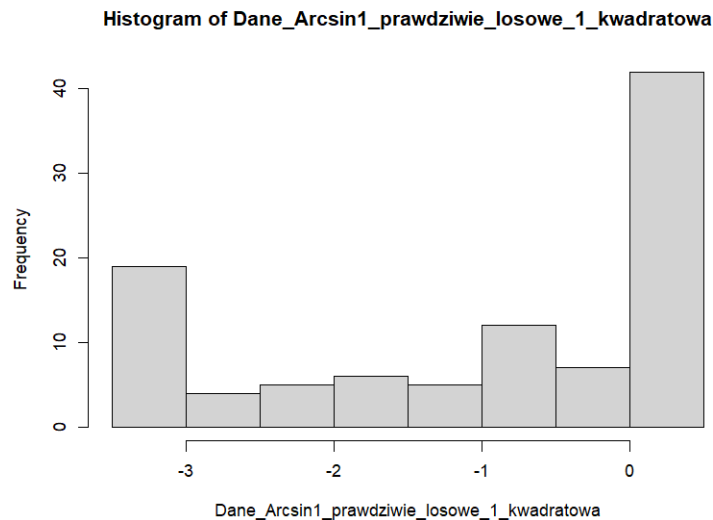
### 1.3.6 3.3 Histogramy wygenerowanych rozkładów

Wykreślamy histogramy wygenerowanych rozkładów.

```

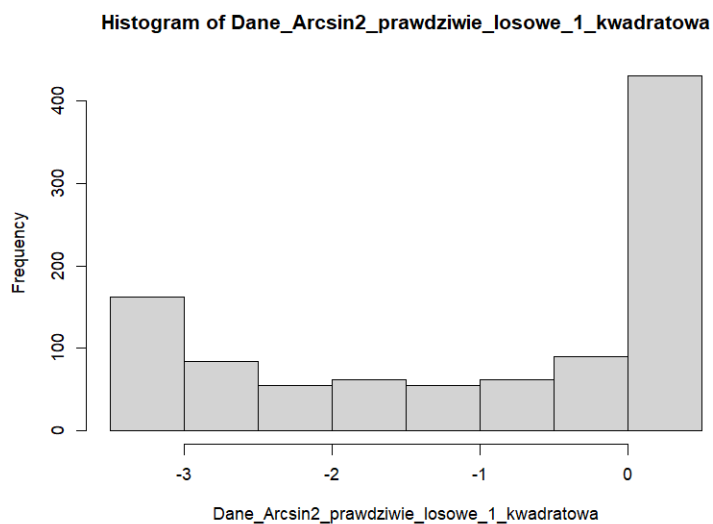
1 hist(Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa)

```



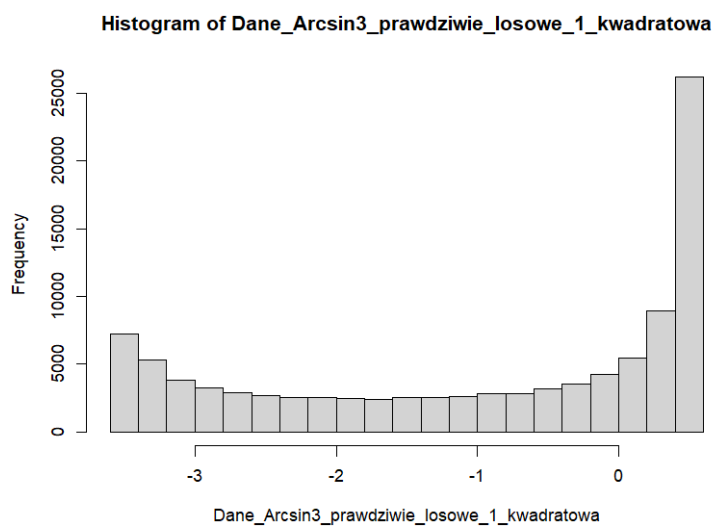
Rysunek 63: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
```



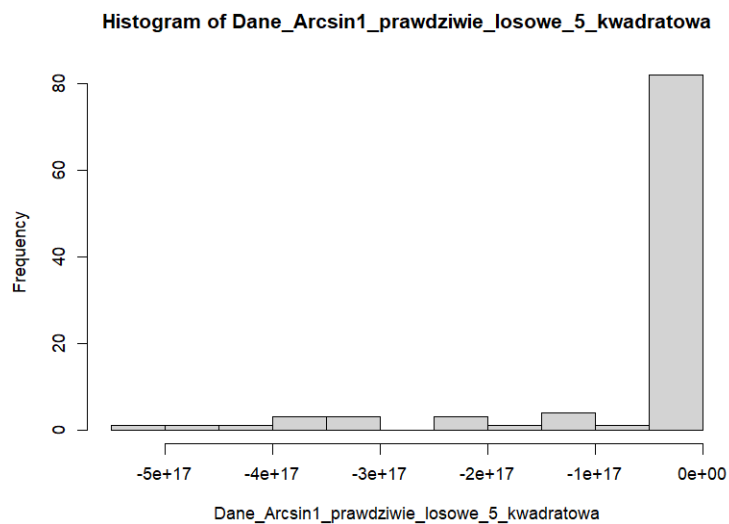
Rysunek 64: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa)
```



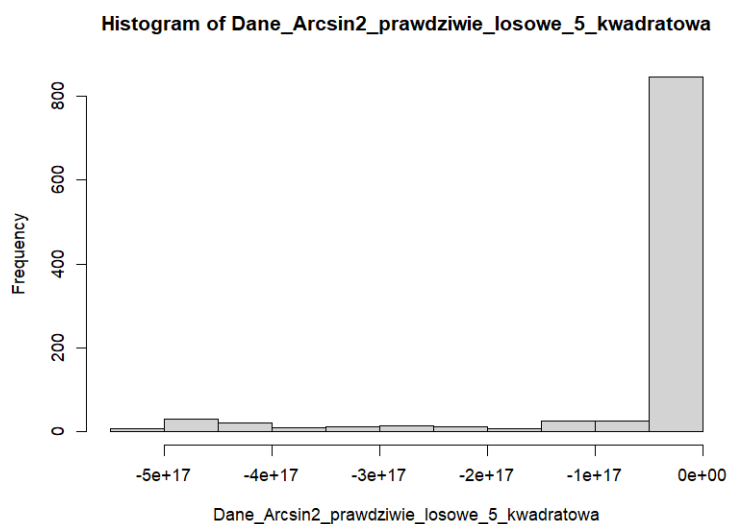
Rysunek 65: Próbkę z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pojedynczym złożeniem mapy chaotycznej.

```
1 hist(Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa)
```



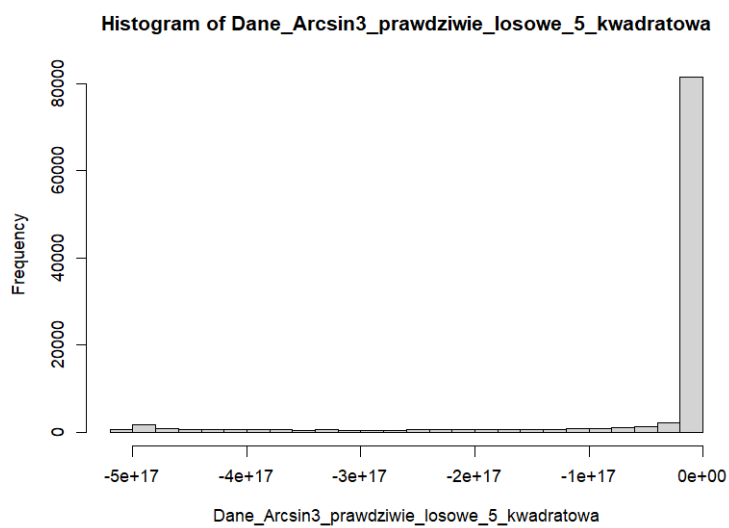
Rysunek 66: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
```



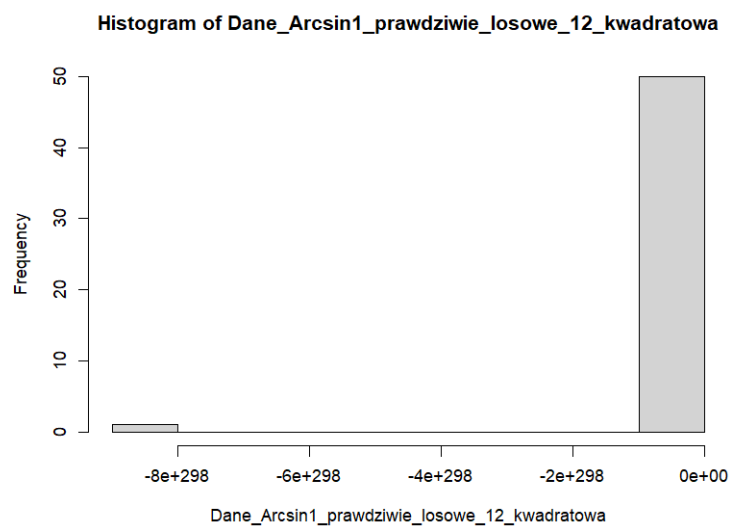
Rysunek 67: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa)
```



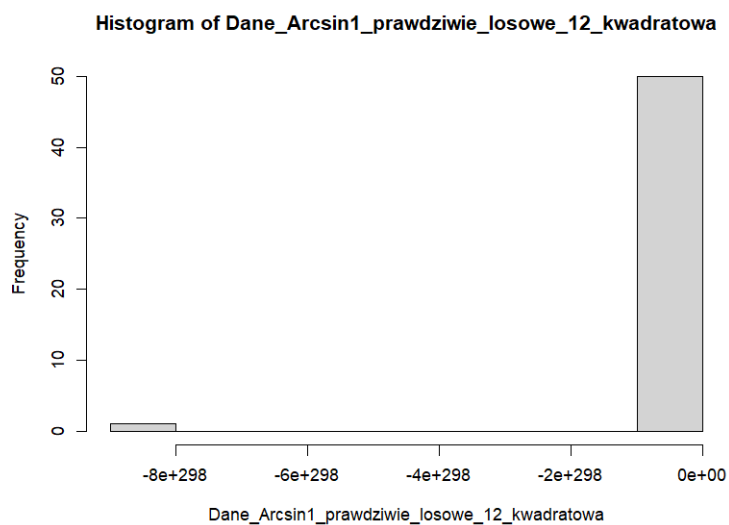
Rysunek 68: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
```



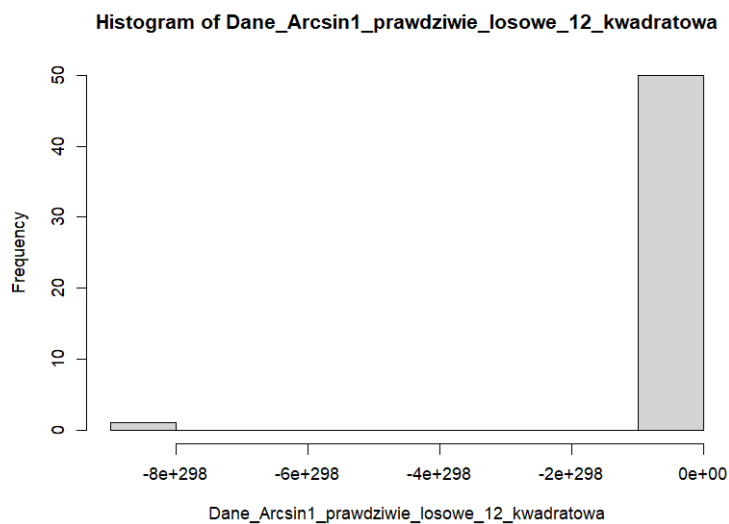
Rysunek 69: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 70: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

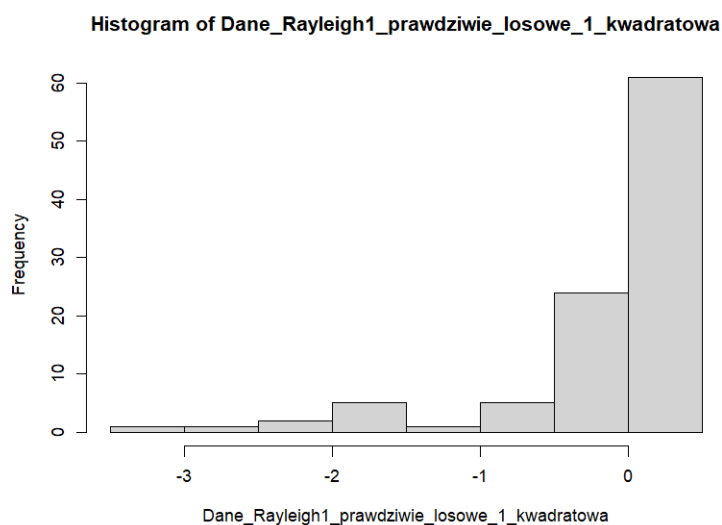
```
1 hist(Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 71: Próbką z normalizowanego rozkładu arcusa sinusa o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

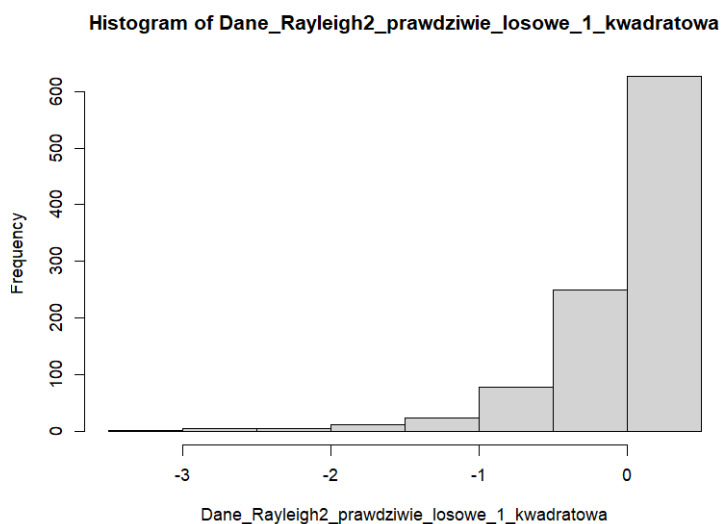


```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa)
```



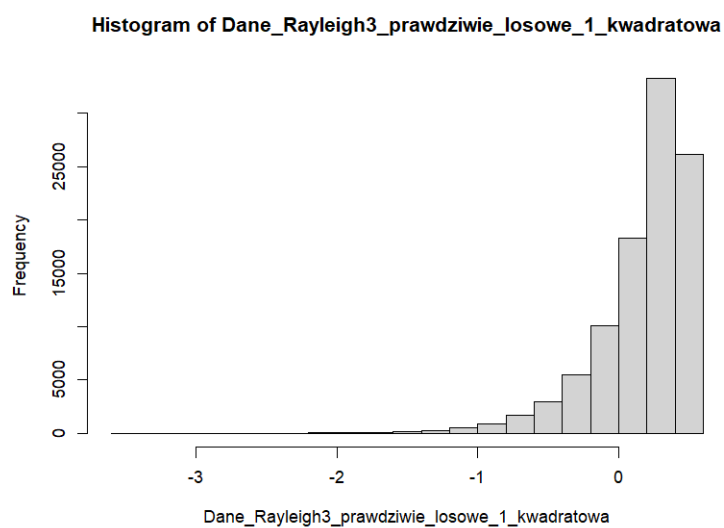
Rysunek 72: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
```



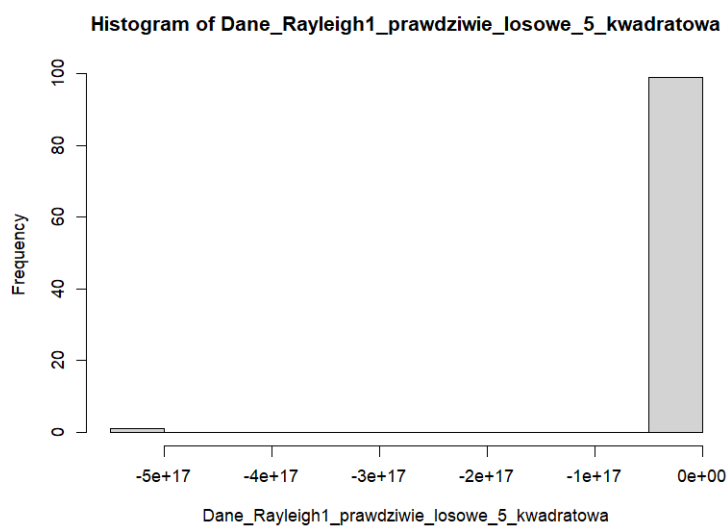
Rysunek 73: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa)
```



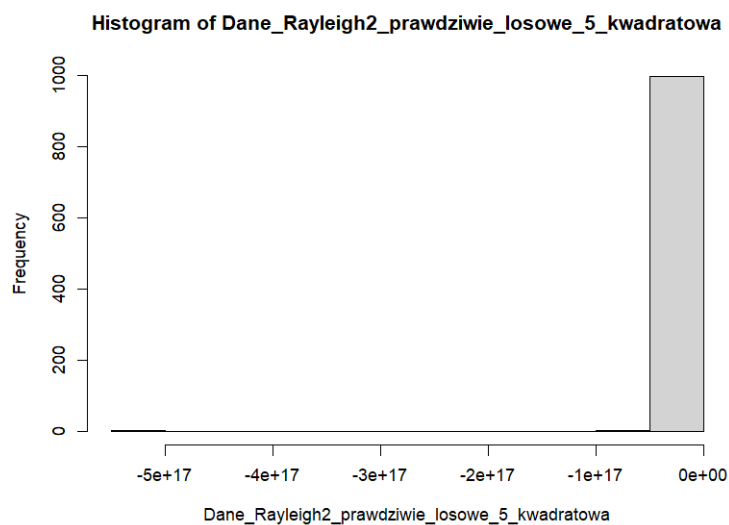
Rysunek 74: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pojedynczym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa)
```



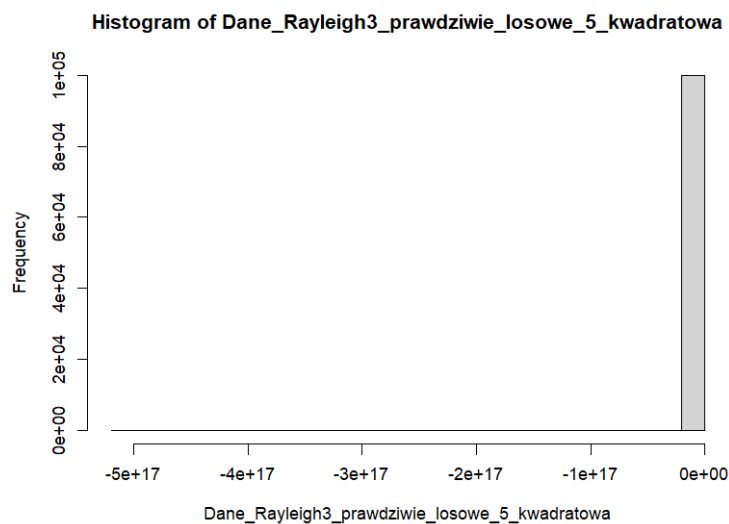
Rysunek 75: Próbką z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
```



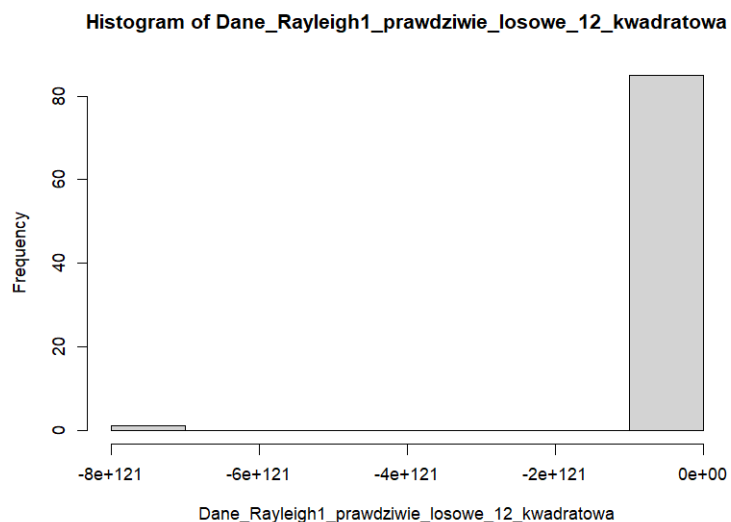
Rysunek 76: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa)
```



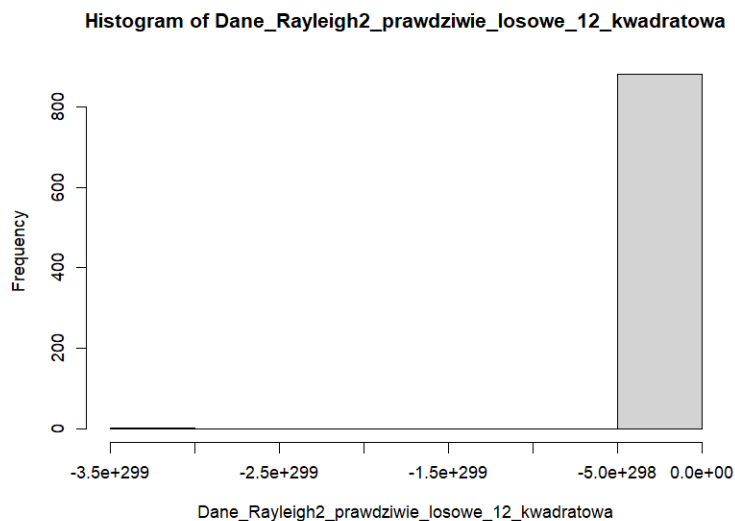
Rysunek 77: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i pięciokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa)
```



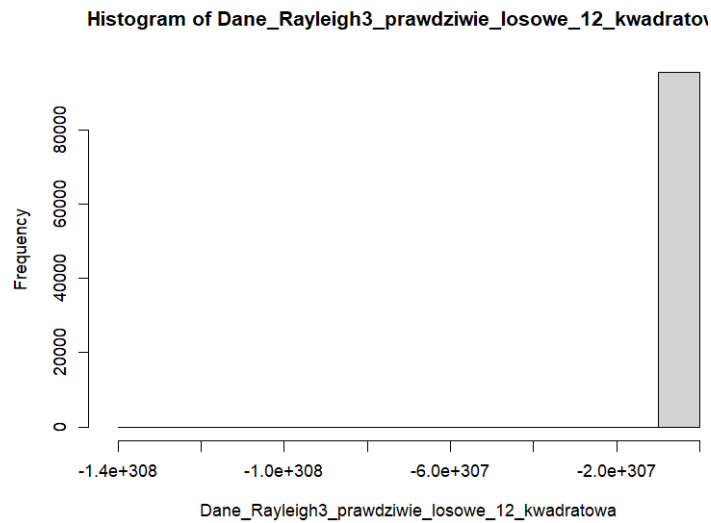
Rysunek 78: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^2$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 79: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^3$  i dwunastokrotnym złożeniu mapy chaotycznej.

```
1 hist(Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 80: Próbkę z normalizowanego rozkładu Rayleigha o rozmiarze  $10^5$  i dwunastokrotnym złożeniu mapy chaotycznej.

Na podstawie powyższych wykresów możemy wywnioskować:

- Im większa liczebność próbki tym mniejsze skoki pomiędzy wartościami

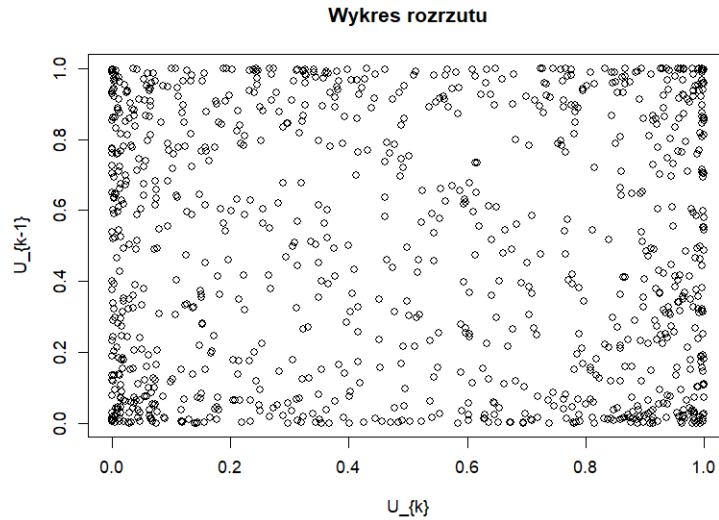
Z mapy kwadratowej nie uzyskujemy rozkładu jednostajnego, wszystkie punkty zbiegają do jednej wartości.

### 1.3.7 3.4 Rozrzut punktów

Generujemy wykresy zależności par elementów z uzyskanych próbek  $(u_k, u_{k+1})$ . Korzystając z zaimplementowanej funkcji generującej wykresy.

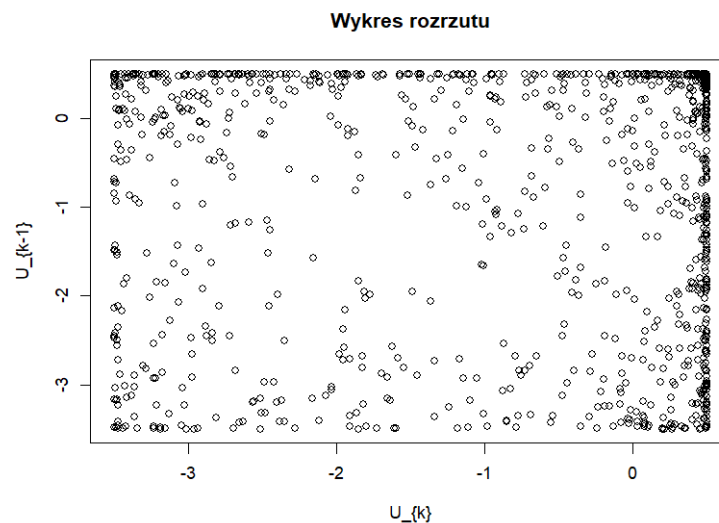
W naszych wykresach ograniczamy się tylko do próbek rozmiaru  $10^3$  dlatego, że dla  $10^2$  nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla  $10^5$  nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
1 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



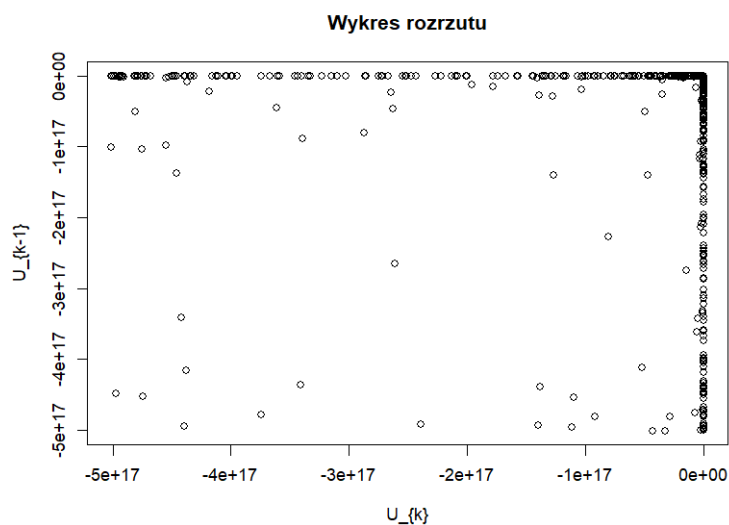
Rysunek 81: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
```



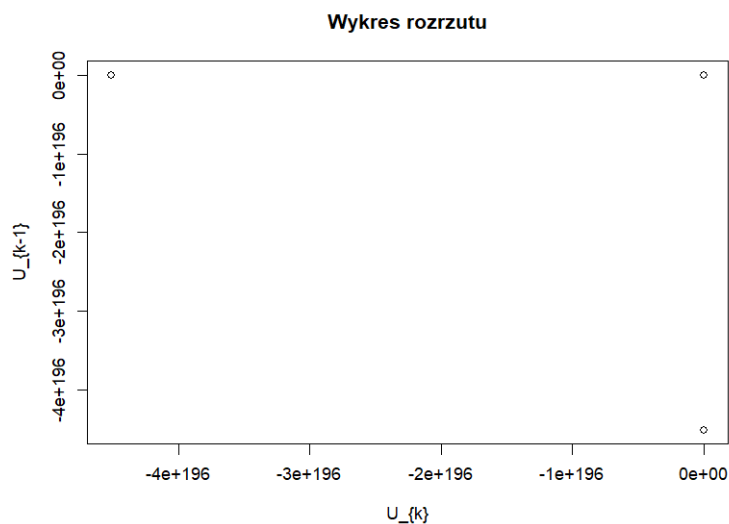
Rysunek 82: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
```



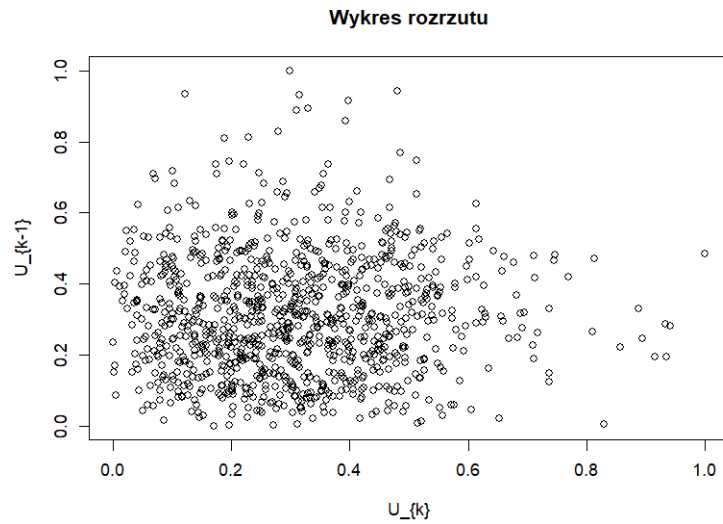
Rysunek 83: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinusua i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)
```



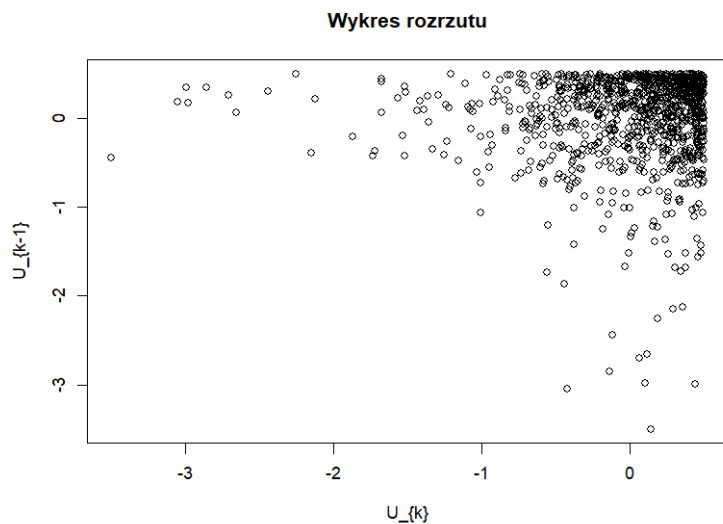
Rysunek 84: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu arcusa sinusua i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



Rysunek 85: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha.

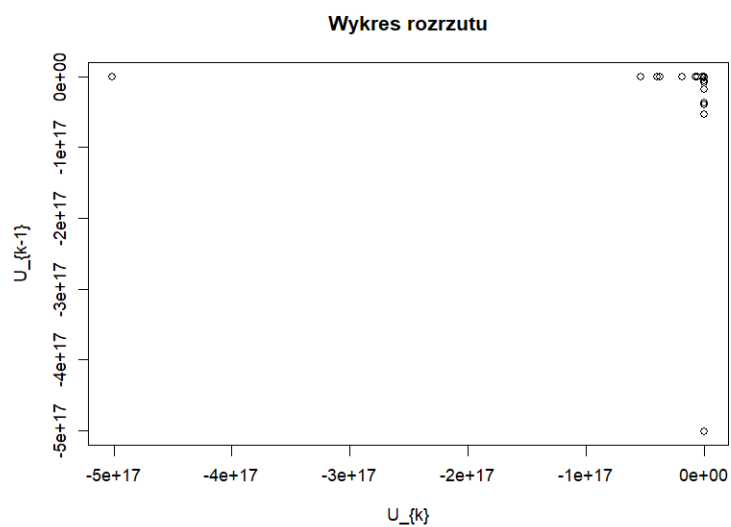
```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
```



Rysunek 86: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

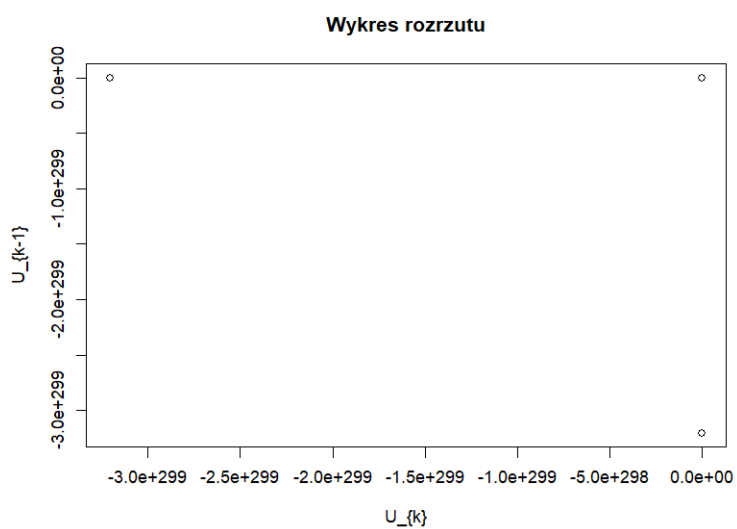


```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
```



Rysunek 87: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 88: Wykres rozrzutu znormalizowanej próbki o rozmiarze  $10^3$  z rozkładu Rayleigha i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

Widać, że dla większych iteracji uzyskujemy wartości w pojedynczych punktach.

## 2 Część II

### 2.1 Zadanie 1

Zapisujemy i ładujemy obrazek (można wykorzystać funkcję `readImage()` z pakietu `OpenImageR`, np. `img = readImage("papugi res.png")`).

Wczytujemy obrazek:

```
1 img <- readImage("papugi_res.png")
2 imageShow(img)
```



Rysunek 89: Wczytany obraz

### 2.2 Zadanie 2

Wyodrębniamy trzy macierze  $R, G, B$  (np. przy wykorzystaniu `readImage()` macierz intensywności koloru czerwonego dla każdego piksela dostajemy za pomocą `img[,1]`). Każdą macierz  $R, G, B$  zapisujemy w postaci wektora o długości  $N \cdot M$ , co daje  $\{R_w, G_w, B_w\}$

Wyodrębniamy trzy macierze  $R, G, B$

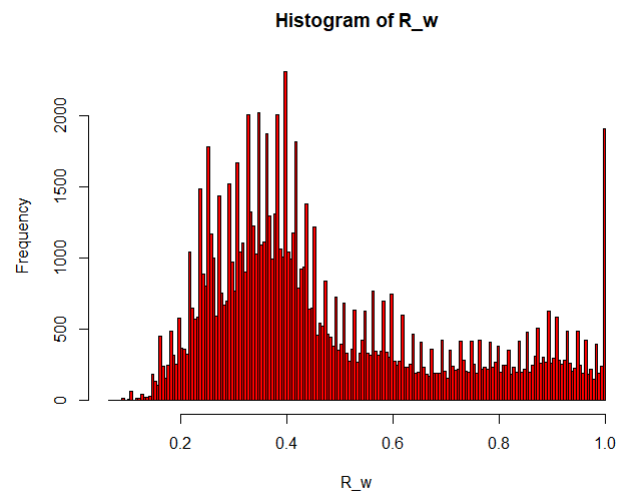
```
1 R <- img[, ,1]
2 G <- img[, ,2]
3 B <- img[, ,3]
4
5 R_w <- c(R)
6 G_w <- c(G)
```

```
7 B_w <- c(B)
```

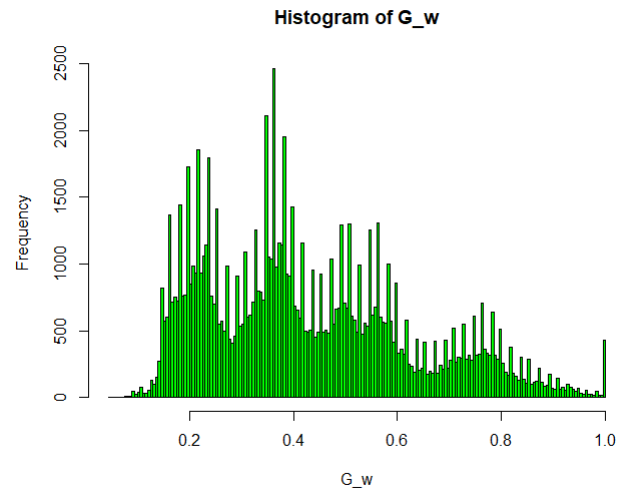
## 2.3 Zadanie 3

Dla każdego koloru oryginalnego obrazka możemy zobrazować intensywność (w formie słupkowej, przykład dla wszystkich kolorów) oraz przedstawić histogramy (przykład dla koloru czerwonego)

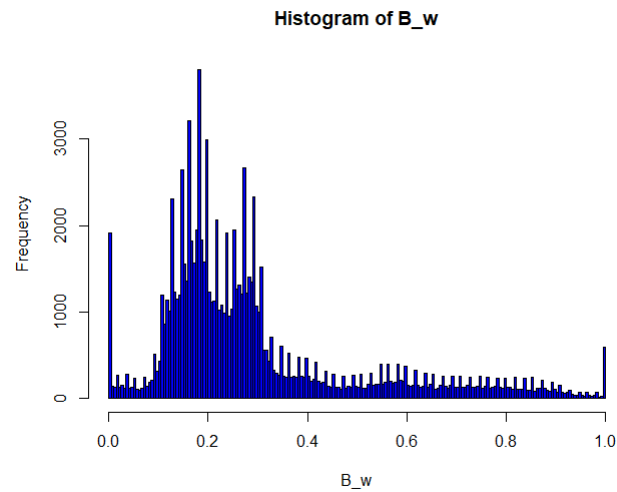
```
1 hist(R_w, col = 'red', breaks = 150)
2 hist(G_w, col = 'green', breaks = 150)
3 hist(B_w, col = 'blue', breaks = 150)
```



Rysunek 90: Histogram dla koloru  $R_w$



Rysunek 91: Histogram dla koloru  $G_w$

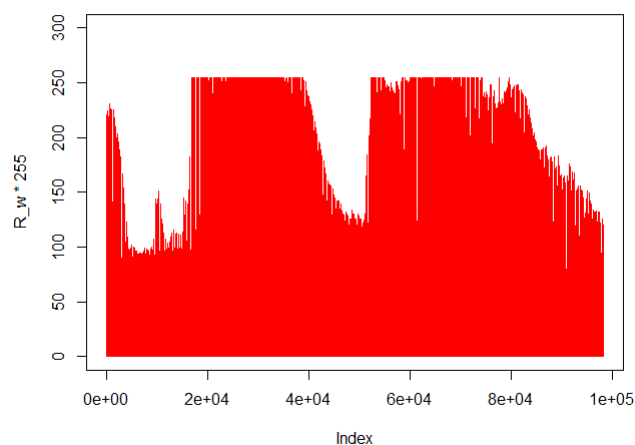


Rysunek 92: Histogram dla koloru  $B_w$

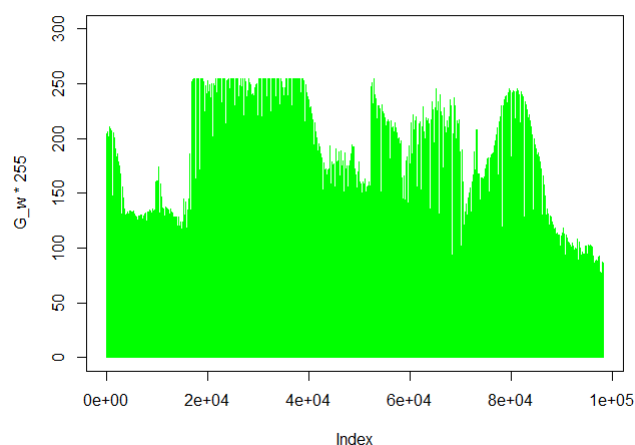
```

1 plot(R_w*255, col = 'red', ylim = c(0,300), type = 'h')
2 plot(G_w*255, col = 'green', ylim = c(0,300), type = 'h')
3 plot(B_w*255, col = 'blue', ylim = c(0,300), type = 'h')

```



Rysunek 93: Wykres intensywności koloru  $R_w$



Rysunek 94: Wykres intensywności koloru  $G_w$

## 2.4 Zadanie 4

Wykorzystując mapę logistyczną

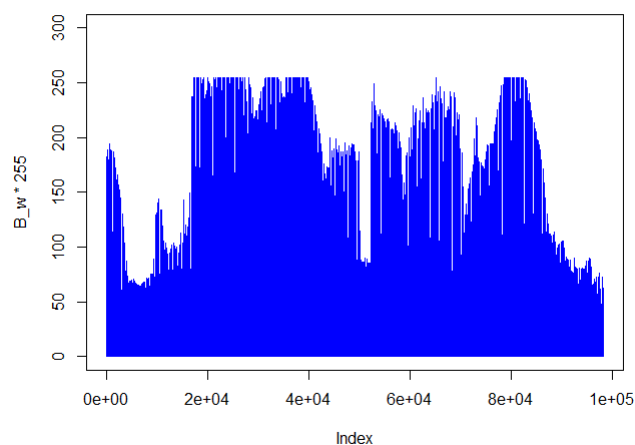
$$x_{n+1} = rx_r(1 - x_n), \quad n = 0, \dots, K$$

dla wybranego  $r \in [3.6, 4]$  i  $x_0 \in (0, 1)$  konstruujemy wektor wartości o długości  $K = 3 \cdot N \cdot M$ . Otrzymany wektor dzielimy na trzy podwektory  $\{x_R, x_G, x_B\}$ . Klucz kodowania składa się z dwóch informacji  $(x_0, r)$ .

```

1 x0 <- 0.5
2 r <- 3.8
3 mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))
4 mapa_logistyczna_wektor <- function(x, x0, r) {
5   tmp <- numeric(length(x))

```



Rysunek 95: Wykres intensywności koloru  $B_w$

```

6   tmp[1] <- x0
7   for (i in 2:length(x)) {
8       tmp[i] <- mapa_logistyczna(tmp[i-1], r)
9   }
10  return(tmp)}
11
12 x_R <- mapa_logistyczna_wektor(R_w, x0 = 0.5, r = 3.8)
13 x_G <- mapa_logistyczna_wektor(G_w, x0 = 0.5, r = 3.8)
14 x_B <- mapa_logistyczna_wektor(B_w, x0 = 0.5, r = 3.8)

```

## 2.5 Zadanie 5

Sortujemy (malejąco lub rosnąco) osobno wektory  $\{x_R, x_G, x_B\}$  otrzymując  $\{x_s^R, x_s^G, x_s^B\}$ . Następnie tworzymy trzy wektory pozycji  $\{P_R, P_G, P_B\}$ , które przechowują numery pozycji elementów z  $\{x_s^R, x_s^G, x_s^B\}$  w wektorach  $\{x_R, x_G, x_B\}$ .

Sortowanie:

```

1 x_R_s <- sort(x_R)
2 x_G_s <- sort(x_G)
3 x_B_s <- sort(x_B)

```

Wektory pozycji:

```

1 P_R <- match(x_R_s, x_R)
2 P_G <- match(x_G_s, x_G)
3 P_B <- match(x_B_s, x_B)

```

## 2.6 Zadanie 6

Dokonujemy tasowania wartości wektorów  $\{R_w, G_w, B_w\}$  za pomocą wektorów pozycji  $\{P_R, P_G, P_B\}$ , tzn. tworzymy nowe wektory  $\{T_R, T_G, T_B\}$ , do których wstawiamy wartości z  $\{R_w, G_w, B_w\}$  w następujący sposób: bierzemy element z  $i$ -tej pozycji wektora  $R_w$ , czyli  $R_w[i]$ , szukamy na jakiej pozycji stoi  $i$  w wektorze  $P_R$  (założmy, że naj), wówczas na pozycji  $j$  wektora  $T_R$ , czyli  $T_R[j]$  wstawiamy  $R_w[i]$ .

Kodowanie:

```
1 kodowanie <- function(x, P){
2   tmp <- numeric(length(x))
3   for (i in 1:length(x)){
4     x1 <- x[i]
5     x2 <- P[i]
6     tmp[x2] <- x1
7   }
8   return(tmp)}
```

Przygotowanie wektorów tasowanych:

```
1 T_R <- kodowanie(R_w, P_R)
2 T_G <- kodowanie(G_w, P_G)
3 T_B <- kodowanie(B_w, P_B)
```

## 2.7 Zadanie 7

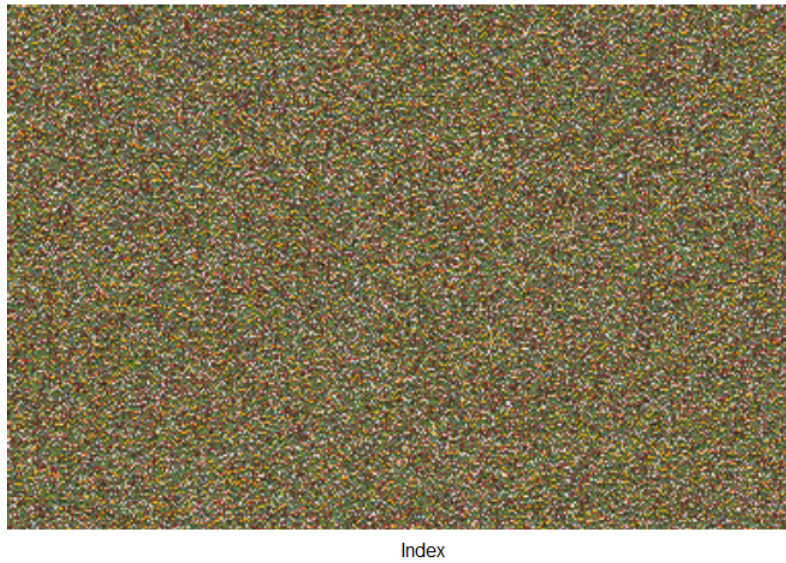
Łączymy wektory kolorów  $\{T_R, T_G, T_B\}$  w tablicę (funkcja `array()`) o wymiarze  $N \times M \times 3$ .

```
1 T <- array(data=c(T_R, T_G, T_B), dim = dim(img))
2 imageShow(T)
```

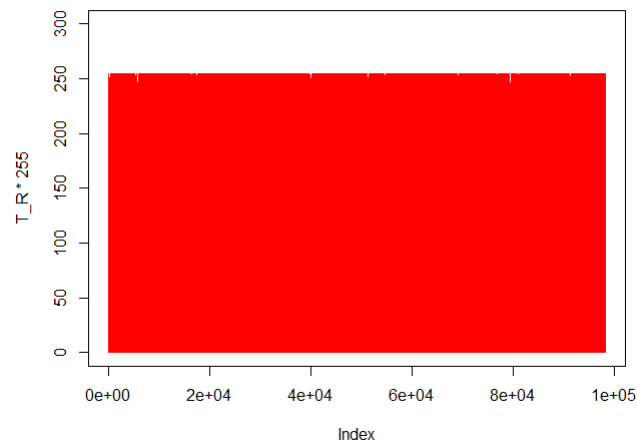
## 2.8 Zadanie 8

Wykreślamy intensywność każdego koloru zakodowanego obrazka

```
1 plot(T_R*255, col = 'red', ylim = c(0,300), type = 'h')
2 plot(T_G*255, col = 'green', ylim = c(0,300), type = 'h')
3 plot(T_B*255, col = 'blue', ylim = c(0,300), type = 'h')
```



Rysunek 96: Zakodowany obraz papug



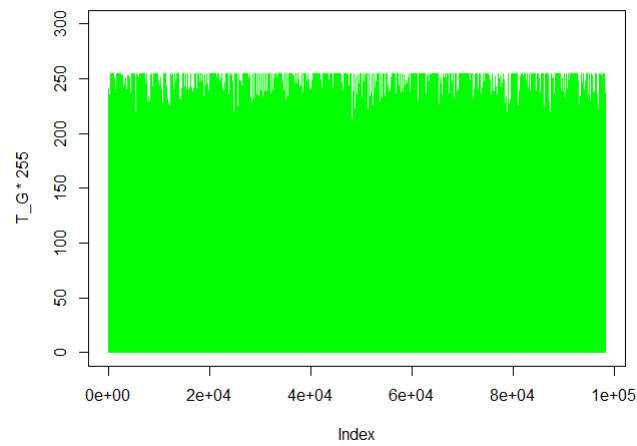
Rysunek 97: Wykres koloru R obrazu zakodowanego

## 2.9 Zadanie 9

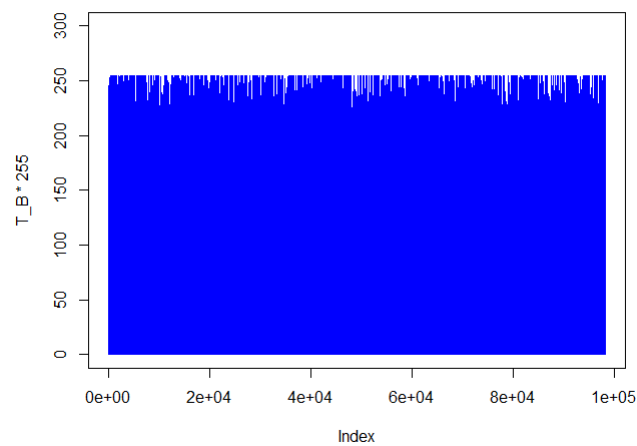
Odbiorca mając klucz kodowania  $(x_0, r)$  może stworzyć wektory pozycji  $\{P_R, P_G, P_B\}$ , które pozwolą poustawiać w nowych wektorach (odkodowane kolory)  $\{D_R, D_G, D_B\}$  wartości z  $\{T_R, T_G, T_B\}$ . Następnie łączymy  $\{D_R, D_G, D_B\}$  w tablicę uzyskując odkodowany obraz.

Klucz:  $x_0 = 0.5$  i  $r = 3.8$





Rysunek 98: Wykres koloru G obrazu zakodowanego



Rysunek 99: Wykres koloru B obrazu zakodowanego

```

1 dekodowanie <- function(x, x0, r){
2   x_mapowane <- mapa_logistyczna_wektor(x, x0, r)
3   x_mapowane_sort <- sort(x_mapowane)
4   x_mapowane_sort_pozycja <- match(x_mapowane_sort, x_mapowane)
5   tmp <- numeric(length(x))
6   for(i in (1:length(x))){
7     pozycja <- x_mapowane_sort_pozycja[i]
8     wartosc <- x[pozycja]
9     tmp[i] <- wartosc
10  }
11  return(tmp)}
12
13 D_R <- dekodowanie(T_R, 0.5, 3.8)
14 D_G <- dekodowanie(T_G, 0.5, 3.8)
15 D_B <- dekodowanie(T_B, 0.5, 3.8)

```

```
16 |  
17 D <- array(c(D_R, D_G, D_B), dim=dim(img))
```



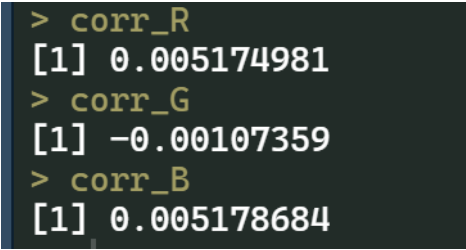
Index

Rysunek 100: Rozkodowany obraz papug

## 2.10 Zadanie 10

Wyznaczamy współczynnik korelacji między poszczególnymi kolorami oryginalnego obrazu a kolorami zakodowanego obrazu, np. dla koloru czerwonego mamy

```
1 wspolczynnik_korelacj <- function(C_w, T_C) {  
2  
3   licznik <- sum((C_w - mean(C_w))*(T_C - mean(T_C)))  
4   mianownik <- sqrt(sum(((C_w - mean(C_w)))^2))*sqrt(sum(((T_C -  
5     mean(T_C)))^2))  
6   return(licznik/mianownik)  
7 }  
8 corr_R <- wspolczynnik_korelacj(R_w, T_R)  
9 corr_G <- wspolczynnik_korelacj(G_w, T_G)  
10 corr_B <- wspolczynnik_korelacj(B_w, T_B)
```



```
> corr_R  
[1] 0.005174981  
> corr_G  
[1] -0.00107359  
> corr_B  
[1] 0.005178684
```

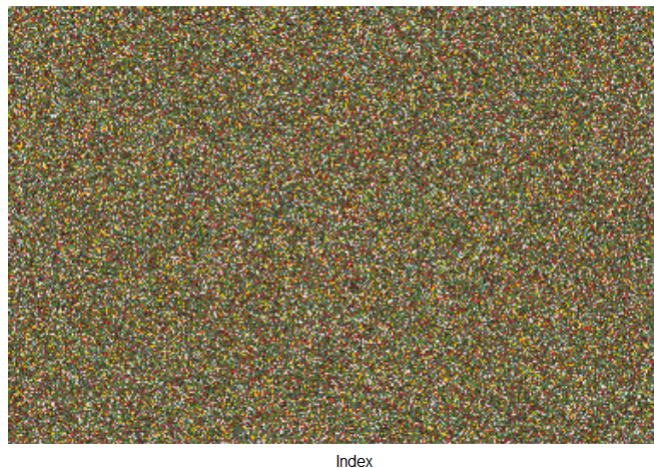
Rysunek 101: Wyniki korelacji dla kolorów R, G i B

## 2.11 Zadanie 11

Sprawdzamy wrażliwość kodowania względem doboru klucza, przykładowo dokonujemy kodowania obrazka przy kluczu ( $x_0 = 0.1, r = 3.8$ ), następnie próbujemy odkodować obrazek przy użyciu klucza ( $x_0 = 0.1000000001, r = 3.8$ ) oraz ( $x_0 = 0.1, r = 3.8000000001$ )

Sprawdzając wrażliwość kodowania względem doboru klucza przyjmujemy  $x_0 = 0.49, r = 3.79$

```
1 D_R_zmienione <- dekodowanie(T_R, x0 = 0.49, r = 3.79)
2 D_G_zmienione <- dekodowanie(T_G, x0 = 0.49, r = 3.79)
3 D_B_zmienione <- dekodowanie(T_B, x0 = 0.49, r = 3.79)
4
5 D_zmienione <- array(c(D_R_zmienione, D_G_zmienione, D_B_zmienione
6                        ), dim=dim(img))
7 imageShow(D_zmienione)
```



Rysunek 102: Rozkodowany obraz papug po zmianie klucza

Wniosek: Już niewielka zmiana klucza uniemożliwia nam rozkodowanie obrazka. Dzieje się tak, ponieważ otrzymujemy nową mapę chaotyczną, która tworzy inny wektor z pozycjami co uniemożliwia rozkodowanie obrazu.

### 3 Kod programu

```
1 # Projekt 1
2 # Aleksandra Burczyk
3 # Monika Gmaj
4 # Kinga Jankowska
5 # Kamil Langowski
6
7
8 # Cz. I
   -----
9
10
11
12
13 # Zadanie 1.
   -----
14
15
16 # Ciagi liczb pseudolosowych generowane metoda dystrybuanty
   odwrotnej dla
17 # zadanej gestosci (rozklad jednostajny U(0,1))
18
19
20 pacman::p_load(moments, pracma)
21 # Rozklad jednostajny
22 set.seed(1)
23 sim_unif <- runif(10^3,0,1)
24
25
26
27 # 1.1 wyznaczc w sposob analityczny dystrybuante i dystrybuante
   odwrotna
28
29 ##### ARCUS SINUS #####
30
31 # rozklad Arcus Sinus  $F(x) : U(0,1) \rightarrow R$ 
32 Arcsin <- function(x){ 2/pi*asin(sqrt(x))}
33
34
35 # rozklad Arcus Sinus  $F^{-1}(x) : U(0,1) \rightarrow R$ 
36 inv_Arcsin <- function(x){ 1/2 - 1/2*cos(x*pi) }
37
38 Dane_Arcsin1 <- inv_Arcsin(runif(10^2,0,1))
39 Dane_Arcsin2 <- inv_Arcsin(runif(10^3,0,1))
40 Dane_Arcsin3 <- inv_Arcsin(runif(10^5,0,1))
41
42
43 # 1.2 przedstawienie dystrybuanty na wykresie
44 dystrybuanty <- function(x1,x2,d1,d2, ylimits) {
45   plot(x1,d1(x1),type = "l",ylim = ylimits,col="green",ylab = "",
       lwd=2,
46       main = "Funkcja gestosci i gestosci odwrotnej")
47   lines(x2,d2(x2), lwd=2)
48   legend("topleft",inset=.02, legend=c("F(x)", "F-1(x)"),
49         col=c("green", "black"),lty=3, cex=0.75,
50         box.lty=1)
```

```

51 }
52
53 par(mfrow=c(1,1))
54 n = 1000
55 dystrybuanty(seq(0,1,length=n), seq(0,1,length=n), Arcsin, inv_
    Arcsin, c(0,1))
56
57 # 1.3 Miary statystyczne
58 Miary <- function(Dane, t_params){
59   Miary_Statystyczne <- c("Mediana", "Srednia", "Dominanta", "
    Kurtoza",
60                           "Wariancja", "Odchylenie standardowe", "
    Rozstep")
61   Wyniki_Z_Generatora <- c(median(Dane), mean(Dane), density(Dane)$
    x[which.max(density(Dane)$y)],
62                           kurtosis(Dane), var(Dane), std(Dane),
    max(Dane)-min(Dane))
63   Wyniki_Teoretyczne <- t_params
64   tabela <- data.frame(Miary_Statystyczne, Wyniki_Z_Generatora,
    Wyniki_Teoretyczne)
65   print(summary(Dane))
66   print(tabela)
67 }
68
69
70 Miary(Dane_Arcsin1, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
71 Miary(Dane_Arcsin2, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
72 Miary(Dane_Arcsin3, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
73
74
75
76 # 1.4 Histogram i gestosc
77 hist_PDF <- function(dane){
78   par(mfrow=c(1,2))
79   hist(dane, prob=TRUE, breaks=40, xlab = "x", main='Histogram',
    ylab = "Czestotliwosc")
80   plot(density(dane), type = "l", pch = 19,
81        col = "red", xlab = "x", ylab = "Gestosc", main='Gesto Ż '
    )
82 }
83
84 hist_PDF(Dane_Arcsin1)
85 hist_PDF(Dane_Arcsin2)
86 hist_PDF(Dane_Arcsin3)
87
88 # 1.5 Dystrybuanta i Dystrubuenta empiryczna
89 dyst_emp_teor <- function(d, d_inv, n, xlims, umin=0, umax=1, tmin
    =0, tmax=1) {
90   plot(ecdf(d_inv(runif(n,umin,umax))), do.points=FALSE,
91        main=paste("Dystrybuanta empiryczna i zgodna z rozkladem"),
92        ylab = "", xlim = xlims)
93
94   x <- seq(tmin,tmax,0.01)
95   lines(x,d(x),type = "l",col="red")
96   legend("bottomright", inset=.02, legend=c("F(x)-Dyst. emp", "F(x)
    -Dyst rozkl"),
97        col=c("black", "red"), lty=3, cex=0.75,
98        box.lty=1)

```

```

99 }
100
101 par(mfrow=c(1,1))
102 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^2, xlims = c(0,1))
103 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^3, xlims = c(0,1))
104 dyst_emp_teor(Arcsin,inv_Arcsin, n=10^5, xlims = c(0,1))
105
106 ##### CAUCHY #####
107
108 # rozklad Cauchyego  $F^*(x) : U(0,1) \rightarrow R$ 
109
110 Cauchy <- function(x,x0 = 0 ,alfa = 1) 1/2 + (1/pi)*atan((x-x0)/
    alfa)
111
112 # rozklad Cauchyego  $F^{-1}(x) : U(0,1) \rightarrow R$ 
113
114 inv_Cauchy <- function(x,x0 = 0,alfa = 1/2){ x0 - alfa*cot(pi*x)}
115
116
117 Dane_inv_Cauchy <- inv_Cauchy(sim_unif)
118 Dane_inv_Cauchy <- Dane_inv_Cauchy[Dane_inv_Cauchy > -10 & Dane_
    inv_Cauchy < 10] # usunac mocno odstajace obs
119
120 Dane_Cauchy1 <- inv_Cauchy(runif(10^2,0,1))
121 Dane_Cauchy2 <- inv_Cauchy(runif(10^3,0,1))
122 Dane_Cauchy3 <- inv_Cauchy(runif(10^5,0,1))
123
124
125
126 # 1.2 przedstawienie dystrybuanty na wykresie
127 par(mfrow = c(1,1))
128 dystrybuanty(seq(-5,5,length=n),seq(0,1,length=n), Cauchy, inv_
    Cauchy, c(-3,3))
129
130 # 1.3 Miary statystyczne
131
132
133 Miary(Dane_Cauchy1, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infy'))
134 Miary(Dane_Cauchy2, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infy'))
135 Miary(Dane_Cauchy3, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infy'))
136
137
138 # 1.4 Histogram i gestosc
139
140 hist_PDF(Dane_Cauchy1)
141 hist_PDF(Dane_Cauchy2)
142 hist_PDF(Dane_Cauchy3)
143
144 # 1.5 Dystrybuanta i Dystrubuenta empiryczna
145
146 par(mfrow=c(1,1))
147 dyst_emp_teor(Cauchy,inv_Cauchy, n=1000, xlims= c(-10,10), umin
    =-10, umax=10, tmin=-10, tmax=10)
148
149

```

```

150 ##### WARTOSCI EKSTREMALNYCH #####
151
152 # rozklad wartosci ekstremalnych  $F^*(x)$  :  $U(0,1) \rightarrow R$ 
153 extr.val <- function(x, alfa=1, beta=1/2) 1-exp(-exp(x-alfa)/beta)
154
155 # rozklad wartosci ekstremalnych  $F^{-1}(x)$  :  $U(0,1) \rightarrow R$ 
156 inv_extr.val <- function(x, alfa=1, beta=1/2) {(log(-alfa*log((1-x))
157   )/beta)}
158
159 hist(inv_extr.val(sim_unif, alfa=1, beta=1/2), prob=TRUE, breaks=40)
160 lines(density(inv_extr.val(sim_unif, alfa=1, beta=1/2)))
161
162 Dane_extr.val1 <- inv_extr.val(runif(10^2, 0, 1))
163 Dane_extr.val2 <- inv_extr.val(runif(10^3, 0, 1))
164 Dane_extr.val3 <- inv_extr.val(runif(10^5, 0, 1))
165
166 # 1.2 przedstawienie dystrybucji na wykresie
167 dystrybucji(seq(-5, 5, length=n), seq(0, 1, length=n), extr.val, inv_
168   extr.val, c(-2, 3))
169
170 # 1.3 Miary statystyczne
171
172 Miary(Dane_extr.val1, c(0, 1-1/2*-digamma(1), 1, 0, 1/2^2*pi^2/6, 0, 0))
173 Miary(Dane_extr.val2, c(0, 1-1/2*-digamma(1), 1, 0, 1/2^2*pi^2/6, 0, 0))
174 Miary(Dane_extr.val3, c(0, 1-1/2*-digamma(1), 1, 0, 1/2^2*pi^2/6, 0, 0))
175
176 # 1.4 Histogram i gestosc
177 hist_PDF(Dane_extr.val1)
178 hist_PDF(Dane_extr.val2)
179 hist_PDF(Dane_extr.val3)
180
181 # 1.5 Dystrybuanta i Dystrybuanta empiryczna
182
183 par(mfrow=c(1, 1))
184 dyst_emp_teor(extr.val, inv_extr.val, n=10000, xlims= c(-5, 5), umin
185   =-5, umax=5, tmin=-5, tmax=5)
186
187 ##### Rayleigh #####
188
189 # rozklad Reyleigha  $F^*(x)$  :  $U(0,1) \rightarrow R$ 
190 Rayleigh <- function(x, alfa = 0, beta=1) 1-exp(-((x-alfa)/beta)^2)
191
192
193 # rozklad Reyleigha  $F^{-1}(x)$  :  $U(0,1) \rightarrow R$ 
194 inv_Rayleigh <- function(x, alfa = 0, beta=1) alfa+beta*(sqrt(log(1
195   /(1-x))))
196
197 hist(inv_Rayleigh(sim_unif), prob=TRUE, breaks=40)
198 lines(density(inv_Rayleigh(sim_unif)))
199
200 Dane_Rayleigh1 <- inv_Rayleigh(runif(10^2, 0, 1))
201 Dane_Rayleigh2 <- inv_Rayleigh(runif(10^3, 0, 1))
202 Dane_Rayleigh3 <- inv_Rayleigh(runif(10^5, 0, 1))
203
204 # 1.2 przedstawienie dystrybucji na wykresie

```



```

204 dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Rayleigh, inv_
    Rayleigh, c(-2,3))
205
206 # 1.3 Miary statystyczne
207
208
209
210 Miary(Dane_Rayleigh1, c(0+1*sqrt(log(2)),0+1*sqrt(pi)/2,0+1/sqrt
    (2),(32-3*pi^2)/(4-pi)^2,1^2*(1-pi/4),sqrt(1^2*(1-pi/4)),'infty
    '))
211 Miary(Dane_Rayleigh2, c(0+1*sqrt(log(2)),0+1*sqrt(pi)/2,0+1/sqrt
    (2),(32-3*pi^2)/(4-pi)^2,1^2*(1-pi/4),sqrt(1^2*(1-pi/4)),'infty
    '))
212 Miary(Dane_Rayleigh3, c(0+1*sqrt(log(2)),0+1*sqrt(pi)/2,0+1/sqrt
    (2),(32-3*pi^2)/(4-pi)^2,1^2*(1-pi/4),sqrt(1^2*(1-pi/4)),'infty
    '))
213
214 # 1.4 Histogram i gestosc
215
216 hist_PDF(Dane_Rayleigh1)
217 hist_PDF(Dane_Rayleigh2)
218 hist_PDF(Dane_Rayleigh3)
219
220 # 1.5 Dystrybuanta i Dystrubuanta empiryczna
221
222 par(mfrow=c(1,1))
223 dyst_emp_teor(Rayleigh,inv_Rayleigh, n=1000, xlims= c(0,3), umin
    =0, umax=1 , tmax=3)
224
225
226 ##### Weibull #####
227
228 # rozklad Weibulla  $F^*(x) : U(0,1) \rightarrow R$ 
229 Weibull <- function(x,alfa=1,beta=2,c=1) 1-exp(-((x-alfa)/beta)^c
    ) # x > alfa
230
231 # rozklad Weibulla  $F^{-1}(x) : U(0,1) \rightarrow R$ 
232 inv_Weibull <- function(x,alfa=1,beta=2,c=1) alfa + beta*((-log(1-
    x))^(1/c))
233
234
235 Dane_Weibull1 <- inv_Weibull(runif(10^2,0,1))
236 Dane_Weibull2 <- inv_Weibull(runif(10^3,0,1))
237 Dane_Weibull3 <- inv_Weibull(runif(10^5,0,1))
238
239 # 1.2 przedstawienie dystrybuanty na wykresie
240 dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Weibull, inv_
    Weibull, c(-3,10))
241
242 # 1.3 Miary statystyczne
243
244
245 Miary(Dane_Weibull1, c(1+2*(log(2)^(1/1)),1+2*gamma((1+1)/1)
    ,1,(2^2/1)*(2*gamma(2)-(gamma(1))^2),(2^2)*(gamma(3)-(gamma(2))
    ^2),sqrt((2^2)*(gamma(3)-(gamma(2))^2)),'infty'))
246 Miary(Dane_Weibull2, c(1+2*(log(2)^(1/1)),1+2*gamma((1+1)/1)
    ,1,(2^2/1)*(2*gamma(2)-(gamma(1))^2),(2^2)*(gamma(3)-(gamma(2))
    ^2),sqrt((2^2)*(gamma(3)-(gamma(2))^2)),'infty'))

```

```

247 Miary(Dane_Weibull3, c(1+2*(log(2)^(1/1)),1+2*gamma((1+1)/1)
      ,1,(2^2/1)*(2*gamma(2)-(gamma(1))^2),(2^2)*(gamma(3)-(gamma(2))^
      ^2),sqrt((2^2)*(gamma(3)-(gamma(2))^2)), 'infty'))
248
249 # 1.4 Histogram i gestosc
250
251
252 hist_PDF(Dane_Weibull1)
253 hist_PDF(Dane_Weibull2)
254 hist_PDF(Dane_Weibull3)
255
256 # 1.5 Dystrybuenta i Dystrubuenta empiryczna
257
258 par(mfrow=c(1,1))
259 dyst_emp_teor(Weibull,inv_Weibull, n=100, xlims= c(0,10), umin=0,
      umax=1, tmin=1, tmax=10)
260
261
262
263 ##### LOGISTYCZNY #####
264
265 # rozklad Logistyczny  $F^*(x) : U(0,1) \rightarrow R$ 
266 Logist <- function(x,alfa = 0 ,beta = 1) 1/(1+exp(-(x-alfa)/beta))
267
268 # rozklad Logistyczny  $F^{-1}(x) : U(0,1) \rightarrow R$ 
269
270 inv_Logist <- function(x,alfa = 0,beta = 1) alfa + beta*log(x/(1-x
      ))
271
272 hist(inv_Logist(sim_unif), prob=TRUE, breaks=40)
273 lines(density(inv_Logist(sim_unif)))
274
275 Dane_Logist1 <- inv_Logist(runif(10^2,0,1))
276 Dane_Logist2 <- inv_Logist(runif(10^3,0,1))
277 Dane_Logist3 <- inv_Logist(runif(10^5,0,1))
278
279 # 1.2 przedstawienie dystrybuenty na wykresie
280 dystrybuenty(seq(-5,5,length=n),seq(0,1,length=n), Logist, inv_
      Logist, c(-3,3))
281
282 # 1.3 Miary statystyczne
283
284
285
286 Miary(Dane_Logist1, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2), '
      infty'))
287 Miary(Dane_Logist2, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2), '
      infty'))
288 Miary(Dane_Logist3, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2), '
      infty'))
289
290 # 1.4 Histogram i gestosc
291
292
293 hist_PDF(Dane_Logist1)
294 hist_PDF(Dane_Logist2)
295 hist_PDF(Dane_Logist3)
296

```

```

297 # 1.5 Dystrybuanta i Dystrubuanty empiryczna
298
299 par(mfrow=c(1,1))
300 dyst_emp_teor(Logist,inv_Logist, n=100, xlims= c(-5,7), umin=0,
    umax=1, tmin=-5, tmax=7)
301
302
303 ##### PARETO #####
304
305 # rozklad Pareto  $F^*(x) : U(0,1) \rightarrow R$ 
306 Pareto <- function(x,c = 3) 1-x^(-c)
307
308 # rozklad Pareto  $F^{-1}(x) : U(0,1) \rightarrow R$ 
309 inv_Pareto <- function(x,c = 3) (1-x)^(-1/c)
310
311 hist(inv_Pareto(sim_unif), prob=TRUE, breaks=40)
312 lines(density(inv_Pareto(sim_unif)))
313
314 Dane_Pareto1 <- inv_Pareto(runif(10^2,0,1))
315 Dane_Pareto2 <- inv_Pareto(runif(10^3,0,1))
316 Dane_Pareto3 <- inv_Pareto(runif(10^5,0,1))
317
318 # 1.2 przedstawienie dystrybuanty na wykresie
319 par(mfrow=c(1,1))
320 dystrybuanty(seq(0,1,length=n),seq(0,1,length=n),Pareto, inv_
    Pareto, c(-3,3))
321
322 # 1.3 Miary statystyczne
323
324
325 Miary(Dane_Pareto1, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infy'))
326 Miary(Dane_Pareto2, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infy'))
327 Miary(Dane_Pareto3, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infy'))
328
329 # 1.4 Histogram i gestosc
330
331 hist_PDF(Dane_Pareto1)
332 hist_PDF(Dane_Pareto2)
333 hist_PDF(Dane_Pareto3)
334
335 # 1.5 Dystrybuanta i Dystrubuanty empiryczna
336
337 par(mfrow=c(1,1))
338 dyst_emp_teor(Pareto,inv_Pareto, n=1000, xlims= c(0,50), umin=0,
    umax=1, tmin=-5, tmax=50)
339
340
341
342 ##### TROJKATNY #####
343
344 # rozklad Pareto  $F^*(x) : U(0,1) \rightarrow R$ 
345
346 Triangular_base <- function(x,a = 0,c = 1/2, b= 1) {
347   if (x < a) value <- 0
348   else if (x <= c) value <- (x-a)^2/((b-a)*(c-a))

```

```

349 else if (x < b) value <- 1 - (b-x)^2/((b-a)*(b-c))
350 else if (b <= x) value <- 1
351
352 return(value)
353 }
354
355
356 Triangular <- function(x, ...) {
357   sapply(x, Triangular_base, ...)
358 }
359
360
361 # rozklad Trojkatny  $F^{-1}(x) : U(0,1) \rightarrow R$ 
362
363 inv_Triangular_base <- function(x,a = 0,c = 1/2, b= 1) {
364   if (x < (c - a) / (b - a)) value <- a + sqrt(max(0, (c - a) * (b
365     - a) * x))
366   else if (x >= (c - a) / (b - a)) value <- b - sqrt(max(0, (b - a
367     ) * (b - c) * (1 - x)))
368   return(value)
369 }
370 inv_Triangular <- function(x, ...) {
371   sapply(x, inv_Triangular_base, ...)
372 }
373
374 hist(inv_Triangular(sim_unif), prob=TRUE, breaks=40)
375 lines(density(inv_Triangular(sim_unif)))
376
377 Dane_Triangular1 <- inv_Triangular(runif(10^2,0,1))
378 Dane_Triangular2 <- inv_Triangular(runif(10^3,0,1))
379 Dane_Triangular3 <- inv_Triangular(runif(10^5,0,1))
380 # 1.2 przedstawienie dystrybuanty na wykresie
381 dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Triangular, inv_
382   Triangular, c(0,1))
383
384 # 1.3 Miary statystyczne
385
386 Miary(Dane_Triangular1, c(min(Dane_Triangular3)+sqrt((max(Dane_
387   Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
388   ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
389   2,-3/5,(3*(max(Dane_Triangular3)-min(Dane_Triangular3))^2+max(
390   Dane_Triangular3)+min(Dane_Triangular3)-1)^2/72,sqrt((3*(max(
391   Dane_Triangular3)-min(Dane_Triangular3))^2+max(Dane_Triangular3
392   )+min(Dane_Triangular3)-1)^2/72),max(Dane_Triangular3)-min(Dane
393   _Triangular3)))
394
395 Miary(Dane_Triangular2, c(min(Dane_Triangular3)+sqrt((max(Dane_
396   Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
397   ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
398   2,-3/5,(3*(max(Dane_Triangular3)-min(Dane_Triangular3))^2+max(
399   Dane_Triangular3)+min(Dane_Triangular3)-1)^2/72,sqrt((3*(max(
400   Dane_Triangular3)-min(Dane_Triangular3))^2+max(Dane_Triangular3
401   )+min(Dane_Triangular3)-1)^2/72),max(Dane_Triangular3)-min(Dane
402   _Triangular3)))
403
404 Miary(Dane_Triangular3, c(min(Dane_Triangular3)+sqrt((max(Dane_
405   Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
406   ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
407   2,-3/5,(3*(max(Dane_Triangular3)-min(Dane_Triangular3))^2+max(

```

```

Dane_Triangular3)+min(Dane_Triangular3)-1)^2/72,sqrt((3*(max(
Dane_Triangular3)-min(Dane_Triangular3))^2+max(Dane_Triangular3
)+min(Dane_Triangular3)-1)^2/72),max(Dane_Triangular3)-min(Dane
_Triangular3)))
387
388 # 1.4 Histogram i gestosc
389
390 hist_PDF(Dane_Triangular1)
391 hist_PDF(Dane_Triangular2)
392 hist_PDF(Dane_Triangular3)
393
394 # 1.5 Dystrybuanta i Dystrubuenta empiryczna
395
396 par(mfrow=c(1,1))
397 dyst_emp_teor(Triangular,inv_Triangular, n=1000, xlims= c(0,1),
    umin=0, umax=1, tmin=0, tmax=1)
398
399
400
401 # Zadanie 2
    -----
402
403 # Dla wygenerowanych pr bek z Zadania 1 ci g w liczb
    pseudolosowych o wybranym rozk Ćadzie
404 # 2.1 Znormalizowa pr bki
405
406 normalizacja <- function(x) (x - min(x))/(max(x) - min(x))
407
408 {
409     Dane_Arcsin1_normalizowane <- normalizacja(Dane_Arcsin1)
410     Dane_Arcsin2_normalizowane <- normalizacja(Dane_Arcsin2)
411     Dane_Arcsin3_normalizowane <- normalizacja(Dane_Arcsin3)
412 }
413
414 {
415     Dane_Rayleigh1_normalizowane <- normalizacja(Dane_Rayleigh1)
416     Dane_Rayleigh2_normalizowane <- normalizacja(Dane_Rayleigh2)
417     Dane_Rayleigh3_normalizowane <- normalizacja(Dane_Rayleigh3)
418 }
419
420
421
422
423 # 2.2
424 # Dla ka dej znormalizowanej pr bki wygenerowa ci g liczb
    pseudolosowych U wykorzystuj c (2),
425 # gdzie fN oznacza N-t iteracj Ć mapy chaotycznej (1) dla p =
    0.45.
426
427 # definiujemy funkcje mapy chaotycznej z polecenia
428 ukosna_mapa_namiotu <- function(zk, p=0.45) {
429     if(zk >= 0 & zk < p) {
430         return(zk/p)
431     }
432     if(zk >= p & zk <= 1) {
433         return((1-zk)/(1-p))
434     }
435     else{

```

```

436     print('Nieprawidłowa wartość ')
437 }
438 }
439
440
441 # definiujemy N-te złożenie mapy chaotycznej z polecenia
442 ukosna_mapa_namiotu_N_iteracji <- function(N, x0, p = 0.45){
443   x = c()
444   for (j in 1:length(x0)) {
445     x[j] = ukosna_mapa_namiotu(x0[j], p)
446
447     if(N > 1) {
448       for(i in 2:N){
449         x[j] = ukosna_mapa_namiotu(x[j], p)
450       }
451     }
452   }
453   return(x)
454 }
455
456
457
458 {
459   Dane_Arcsin1_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Arcsin1_normalizowane)
460   Dane_Arcsin2_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Arcsin2_normalizowane)
461   Dane_Arcsin3_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Arcsin3_normalizowane)
462 }
463
464 {
465   Dane_Arcsin1_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Arcsin1_normalizowane)
466   Dane_Arcsin2_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Arcsin2_normalizowane)
467   Dane_Arcsin3_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Arcsin3_normalizowane)
468 }
469
470 {
471   Dane_Arcsin1_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Arcsin1_normalizowane)
472   Dane_Arcsin2_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Arcsin2_normalizowane)
473   Dane_Arcsin3_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Arcsin3_normalizowane)
474 }
475
476
477 {
478   Dane_Rayleigh1_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Rayleigh1_normalizowane)
479   Dane_Rayleigh2_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Rayleigh2_normalizowane)
480   Dane_Rayleigh3_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_
iteracji(1, Dane_Rayleigh3_normalizowane)
481 }

```

```

482 {
483 {
484 Dane_Rayleigh1_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Rayleigh1_normalizowane)
485 Dane_Rayleigh2_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Rayleigh2_normalizowane)
486 Dane_Rayleigh3_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_
iteracji(5, Dane_Rayleigh3_normalizowane)
487 }
488
489 {
490 {
491 Dane_Rayleigh1_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Rayleigh1_normalizowane)
492 Dane_Rayleigh2_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Rayleigh2_normalizowane)
493 Dane_Rayleigh3_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_
iteracji(12, Dane_Rayleigh3_normalizowane)
494 }
495
496 # 2.3 Wygenerowa histogramy rozkładu U dla każdej z próbek,
dla trzech wartości N = {1, 5, 12}
497
498 hist(Dane_Arcsin1_prawdziwie_losowe_1)
499 hist(Dane_Arcsin2_prawdziwie_losowe_1)
500 hist(Dane_Arcsin3_prawdziwie_losowe_1)
501
502 hist(Dane_Arcsin1_prawdziwie_losowe_5)
503 hist(Dane_Arcsin2_prawdziwie_losowe_5)
504 hist(Dane_Arcsin3_prawdziwie_losowe_5)
505
506 hist(Dane_Arcsin1_prawdziwie_losowe_12)
507 hist(Dane_Arcsin2_prawdziwie_losowe_12)
508 hist(Dane_Arcsin3_prawdziwie_losowe_12)
509
510
511
512
513 hist(Dane_Rayleigh1_prawdziwie_losowe_1)
514 hist(Dane_Rayleigh2_prawdziwie_losowe_1)
515 hist(Dane_Rayleigh3_prawdziwie_losowe_1)
516
517 hist(Dane_Rayleigh1_prawdziwie_losowe_5)
518 hist(Dane_Rayleigh2_prawdziwie_losowe_5)
519 hist(Dane_Rayleigh3_prawdziwie_losowe_5)
520
521 hist(Dane_Rayleigh1_prawdziwie_losowe_12)
522 hist(Dane_Rayleigh2_prawdziwie_losowe_12)
523 hist(Dane_Rayleigh3_prawdziwie_losowe_12)
524
525
526 # 2.4 W każdym przypadku wygenerowa wykresy zależności par (
uk, uk+1)
527 # (rozrzut punkt w na płaszczyźnie (uk, uk+1)).
528
529 wykres_rozrzutu <- function(df) plot(x = df[-length(df)], y = df
[-1],
530                                     xlab='U_{k}',

```

```

531                                     ylab = 'U_{k-1}',
532                                     main = 'Wykres rozrzutu')
533
534 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
535 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1)
536 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
537 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12)
538
539
540
541 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
542 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1)
543 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5)
544 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12)
545
546
547
548
549
550 # Zadanie 3
551 -----
552 # Dla wygenerowanych pr bek z Zadania 1 ci g w liczb
553   pseudolosowych o wybranym rozk Ćadzie
554 # wykona punkty 2.1-2.4 dla dw ch dowolnie wybranych z [1]
555   map chaotycznych.
556
557 #
558 # definiujemy funkcje mapy logistycznej
559 mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))
560
561 # definiujemy N-te zlozenie mapy chaotycznej
562 mapa_logistyczna_N_iteracji <- function(N, x0, r=3.8){
563   x = c()
564   for (j in 1:length(x0)) {
565     x[j] = mapa_logistyczna(x0[j], r)
566     if(N > 1){
567       for(i in 2:N){
568         x[j] = mapa_logistyczna(x[j], r)
569       }
570     }
571   }
572   return(x)
573 }
574
575 # definiujemy funkcje mapy kwadratowej
576 mapa_kwadratowa <- function(xk, a=4, b=0.5) return(b - a*xk^2)
577
578 # definiujemy N-te zlozenie mapy chaotycznej
579 mapa_kwadratowa_N_iteracji <- function(N, x0, a=4, b=0.5){
580   x = c()
581   for (j in 1:length(x0)) {
582     x[j] = mapa_kwadratowa(x0[j], a, b)
583     if(N>1){

```



```

586     for(i in 2:N){
587         x[j] = mapa_kwadratowa(x[j], a, b)
588     }
589 }
590 }
591 return(x)
592 }
593
594 ##### MAPA LOGISTYCZNA #####
595
596
597 # 3.2
598 # Dla ka de j znormalizowanej pr bki wygenerowa ci g liczb
   pseudolosowych U wykorzystuj c (2),
599 # gdzie fN oznacza N-t iteracj Ĺ mapy chaotycznej (1) dla p =
   0.45.
600
601 {
602     Dane_Arcsin1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
        _N_iteracji(1, Dane_Arcsin1_normalizowane)
603     Dane_Arcsin2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
        _N_iteracji(1, Dane_Arcsin2_normalizowane)
604     Dane_Arcsin3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna
        _N_iteracji(1, Dane_Arcsin3_normalizowane)
605 }
606
607 {
608     Dane_Arcsin1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
        _N_iteracji(5, Dane_Arcsin1_normalizowane)
609     Dane_Arcsin2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
        _N_iteracji(5, Dane_Arcsin2_normalizowane)
610     Dane_Arcsin3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna
        _N_iteracji(5, Dane_Arcsin3_normalizowane)
611 }
612
613 {
614     Dane_Arcsin1_prawdziwie_losowe_12_logistyczna <- mapa_
        logistyczna_N_iteracji(12, Dane_Arcsin1_normalizowane)
615     Dane_Arcsin2_prawdziwie_losowe_12_logistyczna <- mapa_
        logistyczna_N_iteracji(12, Dane_Arcsin2_normalizowane)
616     Dane_Arcsin3_prawdziwie_losowe_12_logistyczna <- mapa_
        logistyczna_N_iteracji(12, Dane_Arcsin3_normalizowane)
617 }
618
619
620 {
621     Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna <- mapa_
        logistyczna_N_iteracji(1, Dane_Rayleigh1_normalizowane)
622     Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna <- mapa_
        logistyczna_N_iteracji(1, Dane_Rayleigh2_normalizowane)
623     Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna <- mapa_
        logistyczna_N_iteracji(1, Dane_Rayleigh3_normalizowane)
624 }
625
626 {
627     Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna <- mapa_
        logistyczna_N_iteracji(5, Dane_Rayleigh1_normalizowane)

```

```

628 Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna <- mapa_
    logistyczna_N_iteracji(5, Dane_Rayleigh2_normalizowane)
629 Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna <- mapa_
    logistyczna_N_iteracji(5, Dane_Rayleigh3_normalizowane)
630 }
631
632 {
633 Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna <- mapa_
    logistyczna_N_iteracji(12, Dane_Rayleigh1_normalizowane)
634 Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna <- mapa_
    logistyczna_N_iteracji(12, Dane_Rayleigh2_normalizowane)
635 Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna <- mapa_
    logistyczna_N_iteracji(12, Dane_Rayleigh3_normalizowane)
636 }
637
638
639 # 3.3 Wygenerowa histogramy rozkładu U dla każdej z próbek,
    dla trzech wartości N = {1, 5, 12}
640
641 hist(Dane_Arcsin1_prawdziwie_losowe_1_logistyczna)
642 hist(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
643 hist(Dane_Arcsin3_prawdziwie_losowe_1_logistyczna)
644
645 hist(Dane_Arcsin1_prawdziwie_losowe_5_logistyczna)
646 hist(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
647 hist(Dane_Arcsin3_prawdziwie_losowe_5_logistyczna)
648
649 hist(Dane_Arcsin1_prawdziwie_losowe_12_logistyczna)
650 hist(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
651 hist(Dane_Arcsin3_prawdziwie_losowe_12_logistyczna)
652
653
654
655
656 hist(Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna)
657 hist(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
658 hist(Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna)
659
660 hist(Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna)
661 hist(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
662 hist(Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna)
663
664 hist(Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna)
665 hist(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
666 hist(Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna)
667
668 # 3.4 W każdym przypadku wygenerowa wykresy zależności par (
    uk, uk+1)
669 # (rozrzut punkt w na płaszczyźnie (uk, uk+1)).
670
671
672 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
673 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
674 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
675 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
676
677
678

```

```

679 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
680 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
681 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
682 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
683
684
685
686 ##### MAPA KWADRATOWA #####
687
688 # 3.2
689 # Dla ka dej znormalizowanej pr bki wygenerowa ci g liczb
    pseudolosowych U wykorzystuj c (2),
690 # gdzie fN oznacza N-t iteracj Ż mapy chaotycznej (1) dla p =
    0.45.
691
692 {
693     Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
        _iteracji(1, Dane_Arcsin1_normalizowane)
694     Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
        _iteracji(1, Dane_Arcsin2_normalizowane)
695     Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
        _iteracji(1, Dane_Arcsin3_normalizowane)
696 }
697
698 {
699     Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
        _iteracji(5, Dane_Arcsin1_normalizowane)
700     Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
        _iteracji(5, Dane_Arcsin2_normalizowane)
701     Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
        _iteracji(5, Dane_Arcsin3_normalizowane)
702 }
703
704 {
705     Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
        N_iteracji(12, Dane_Arcsin1_normalizowane)
706     Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
        N_iteracji(12, Dane_Arcsin2_normalizowane)
707     Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
        N_iteracji(12, Dane_Arcsin3_normalizowane)
708 }
709
710
711 {
712     Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa
        _N_iteracji(1, Dane_Rayleigh1_normalizowane)
713     Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa
        _N_iteracji(1, Dane_Rayleigh2_normalizowane)
714     Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa
        _N_iteracji(1, Dane_Rayleigh3_normalizowane)
715 }
716
717 {
718     Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa
        _N_iteracji(5, Dane_Rayleigh1_normalizowane)
719     Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa
        _N_iteracji(5, Dane_Rayleigh2_normalizowane)

```

```

720 Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa
      _N_iteracji(5, Dane_Rayleigh3_normalizowane)
721 }
722
723 {
724   Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa <- mapa_
      kwadratowa_N_iteracji(12, Dane_Rayleigh1_normalizowane)
725   Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa <- mapa_
      kwadratowa_N_iteracji(12, Dane_Rayleigh2_normalizowane)
726   Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa <- mapa_
      kwadratowa_N_iteracji(12, Dane_Rayleigh3_normalizowane)
727 }
728
729
730 # 3.3 Wygenerowa histogramy rozkładu U dla każdej z próbek,
      dla trzech wartości N = {1, 5, 12}
731
732 hist(Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa)
733 hist(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
734 hist(Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa)
735
736 hist(Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa)
737 hist(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
738 hist(Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa)
739
740 hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
741 hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
742 hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
743
744
745
746 hist(Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa)
747 hist(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
748 hist(Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa)
749
750 hist(Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa)
751 hist(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
752 hist(Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa)
753
754 hist(Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa)
755 hist(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
756 hist(Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa)
757
758 # 3.4 W każdym przypadku wygenerowa wykresy zależności par (
      uk, uk+1)
759 # (rozrzut punktów w na płaszczyźnie (uk, uk+1)).
760
761
762 wykres_rozrzutu(Dane_Arcsin2_normalizowane)
763 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
764 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
765 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)
766
767
768
769 wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
770 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
771 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)

```

```

772 wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
773
774 # Cz. II
775 -----
776
775 library(OpenImageR)
776
777 # 1. Zapisujemy i Ćadujemy obrazek
778 img <- readImage("papugi_res.png")
779 imageShow(img)
780
781 # 2. Wyodr Żbniamy trzy macierze {R, G,B}
782
783 R <- img[, ,1]
784 G <- img[, ,2]
785 B <- img[, ,3]
786 R_w <- c(R)
787 G_w <- c(G)
788 B_w <- c(B)
789
790 # 3. Dla ka dego koloru oryginalnego obrazka mo emy zobrazowa
791 intensywno Ź
792 # oraz przedstawi histogramy
793
793 hist(R_w, col = 'red', breaks = 150)
794 hist(G_w, col = 'green', breaks = 150)
795 hist(B_w, col = 'blue', breaks = 150)
796
797 plot(R_w*255, col = 'red', ylim = c(0,300), type = 'h')
798 plot(G_w*255, col = 'green', ylim = c(0,300), type = 'h')
799 plot(B_w*255, col = 'blue', ylim = c(0,300), type = 'h')
800
801 # 4. Wykorzystuj c map Ź logistyczn dla wybranego r [3.6,
802 4] i x0 (0, 1)
803 # konstruujemy wektor warto Źci o d Ćugo Źci K = 3 N M .
804 # Otrzymany wektor dzielimy na trzy podwektory {xR, xG, xB}.
805 # Klucz kodowania sk Ćada si Ź z dw ch informacji (x0, r).
806
806 # przyjmujemy x0 = 0.5 i r = 3.8
807
808 x0 <- 0.5
809 r <- 3.8
810 mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))
811 mapa_logistyczna_wektor <- function(x, x0, r) {
812
813 tmp <- numeric(length(x))
814
815 tmp[1] <- x0
816
817 for (i in 2:length(x)) {
818
819 tmp[i] <- mapa_logistyczna(tmp[i-1], r)
820
821 }
822
823 return(tmp)
824 }
825

```

```

826 x_R <- mapa_logistyczna_wektor(R_w, x0 = 0.5, r =3.8)
827 x_G <- mapa_logistyczna_wektor(G_w, x0 = 0.5, r =3.8)
828 x_B <- mapa_logistyczna_wektor(B_w, x0 = 0.5, r =3.8)
829
830
831 # 5. Sortujemy (malej co lub rosn co) osobno wektory {xR, xG, xB
      } otrzymuj c {xsR, xsG, xsB}.
832 # Nast Ępnie tworzymy trzy wektory pozycji {PR, PG, PB}, kt re
      przechowuj numery pozycji
833 # element w z {xsR, xsG, xsB} w wektorach {xR, xG, xB}.
834
835
836
837
838 # sortowanie
839 x_R_s <- sort(x_R)
840 x_G_s <- sort(x_G)
841 x_B_s <- sort(x_B)
842
843
844
845
846 # wektory pozycji
847 P_R <- match(x_R_s, x_R)
848 P_G <- match(x_G_s, x_G)
849 P_B <- match(x_B_s, x_B)
850
851 # 6. Kodowanie
852 kodowanie <- function(x, P){
853
854     tmp <- numeric(length(x))
855
856     for (i in 1:length(x)){
857         x1 <- x[i]
858         x2 <- P[i]
859         tmp[x2] <- x1
860     }
861
862     return(tmp)
863 }
864 # przygotowanie wektor w tasowanych
865
866 T_R <- kodowanie(R_w, P_R)
867 T_G <- kodowanie(G_w, P_G)
868 T_B <- kodowanie(B_w, P_B)
869
870 # 7. Ączenie wektor w
871
872 T <- array(data=c(T_R, T_G, T_B), dim = dim(img))
873
874 # wy Ĺwietlenie zakodowanego obrazka
875 imageShow(T)
876
877 # 8. Wykre Ĺlamy intensywno Ĺ ka dego koloru zakodowanego
      obrazka
878
879 plot(T_R*255, col = 'red', ylim = c(0,300), type = 'h')
880 plot(T_G*255, col = 'green', ylim = c(0,300), type = 'h')

```

```

881 plot(T_B*255, col = 'blue', ylim = c(0,300), type = 'h')
882
883
884 # 9. Dekodowanie
885
886 # klucz: x0 = 0.5 i r = 3.8
887
888
889 dekodowanie <- function(x, x0, r){
890   x_mapowane <- mapa_logistyczna_wektor(x, x0, r)
891   x_mapowane_sort <- sort(x_mapowane)
892   x_mapowane_sort_pozycja <- match(x_mapowane_sort, x_mapowane)
893   tmp <- numeric(length(x))
894   for(i in (1:length(x))){
895     pozycja <- x_mapowane_sort_pozycja[i]
896     wartosc <- x[pozycja]
897     tmp[i] <- wartosc
898   }
899   return(tmp)
900 }
901
902
903 D_R <- dekodowanie(T_R, 0.5, 3.8)
904 D_G <- dekodowanie(T_G, 0.5, 3.8)
905 D_B <- dekodowanie(T_B, 0.5, 3.8)
906
907 D <- array(c(D_R, D_G, D_B), dim=dim(img))
908
909
910 imageShow(D)
911
912
913 # 10 Wyznaczamy wsp  czynnik korelacji mi  dzy poszczeg  lnymi
914   kolorami oryginalnego obrazu
915 # a kolorami zakodowanego obrazu
916
917 wspolczynnik_korelacj <- function(C_w, T_C) {
918
919   licznik <- sum((C_w - mean(C_w))*(T_C - mean(T_C)))
920   mianownik <- sqrt(sum(((C_w - mean(C_w)))^2))*sqrt(sum(((T_C -
921     mean(T_C)))^2))
922   return(licznik/mianownik)
923 }
924
925 corr_R <- wspolczynnik_korelacj(R_w, T_R)
926 corr_G <- wspolczynnik_korelacj(G_w, T_G)
927 corr_B <- wspolczynnik_korelacj(B_w, T_B)
928
929 # 12. Sprawdzamy wra  liwo   kodowania wzgl  dem doboru klucza.
930 # przyjmujemy x0 = 0.49, r = 3.79
931
932 D_R_zmienione <- dekodowanie(T_R, x0 = 0.49, r = 3.79)
933 D_G_zmienione <- dekodowanie(T_G, x0 = 0.49, r = 3.79)
934 D_B_zmienione <- dekodowanie(T_B, x0 = 0.49, r = 3.79)
935
936 D_zmienione <- array(c(D_R_zmienione, D_G_zmienione, D_B_zmienione
937   ), dim=dim(img))

```

```
936  
937 imageShow(D_zmienione)
```