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 Szafrańska

2 lutego 2023

Spis treści

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:

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

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

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:

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

Liczymy całkę z gęstości:

$$F(x) = \int_0^1 \frac{1}{t\sqrt{t(1-t)}}dt = \frac{1}{\pi}\int_0^1 \frac{1}{\sqrt{t}\sqrt{1-t}}dt = (1) = \frac{1}{\pi}\int_0^1 \frac{2ds}{\sqrt{1-s^2}} =$$

$$= \frac{2arcsin(s)}{\pi}\bigg|_0^1 = \frac{2arcsin(\sqrt{x})}{\pi} - \frac{2arcsin(\sqrt{-\infty})}{\pi} = \frac{2arcsin(\sqrt{x})}{\pi}$$

$$(1) \qquad s = \sqrt{t} \quad s^2 = t \quad 2sds = dt$$

Liczymy dystrybuantę odwrotną:

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

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

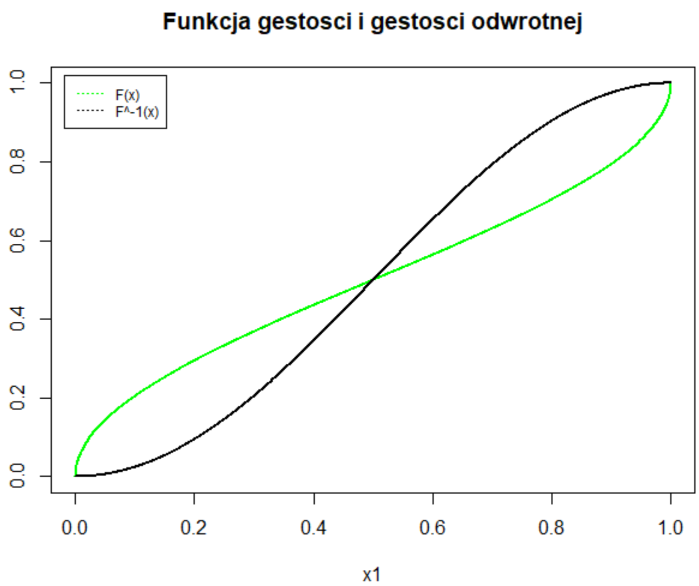
Ze wzoru $1 - cos(\alpha) = 2sin^2(\frac{\alpha}{2})$ mamy :

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

1.1.2 1.2 Wykres dystrybuanty i dystrybuanty odwrotnej

Przedstawić obie dystrybuanty na jednym wykresie.

```
dystrybuanty <- function(x1,x2,d1,d2, ylimits) {
  plot(x1,d1(x1),type = "l",ylim = ylimits,col="green",ylab = "",
    lwd=2,
  main = "Funkcja gestosci i gestosci odwrotnej")
  lines(x2,d2(x2), lwd=2)
  legend("topleft",inset=.02, legend=c("F(x)", "F^-1(x)"), col=c("
    green", "black"),lty=3, cex=0.75, box.lty=1)

  par(mfrow=c(1,1))
    n = 1000
    dystrybuanty(seq(0,1,length=n), seq(0,1,length=n), Arcsin, inv
    _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.

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

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

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

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

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

$$\int_{\infty}^{\infty} xdF(x) = \int_{\infty}^{\infty} xf(x)dx$$

$$\int_0^1 \frac{x}{\pi\sqrt{x}\sqrt{1-x}}dx = \frac{1}{\pi}\int_0^1 \frac{x}{\sqrt{x}\sqrt{1-x}}dx = (2) = \frac{2}{\pi}\int_0^1 \frac{t^2}{\sqrt{1-t^2}}dt =$$

$$= \frac{2}{\pi}\Big(\frac{arcsin\sqrt{x}}{2} - \frac{\sqrt{x}\sqrt{1-x}}{2}\Big)\Big|_0^1 = \frac{2}{\pi}(\frac{\pi}{4} - 0) = \frac{1}{2}$$

(2)

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

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

$$f'(x) = \Big(\frac{1}{\pi\sqrt{x}\sqrt{1-x}}\Big)' = \frac{1}{\pi}\Big(-\frac{\sqrt{x}\sqrt{1-x}}{(1-x)x}\Big)' = -\frac{1}{\pi}\frac{\frac{(1-x)'\sqrt{x}}{2\sqrt{1-x}} + \frac{\sqrt{1-x}}{2\sqrt{x}}}{(1-x)x} =$$

$$= \frac{1}{2x(1-x)\sqrt{1-x}\sqrt{x}} - \frac{1}{2\pi x\sqrt{1-x}\sqrt{x}}$$

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.

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

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

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

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

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

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

Stąd mamy:

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

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:

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

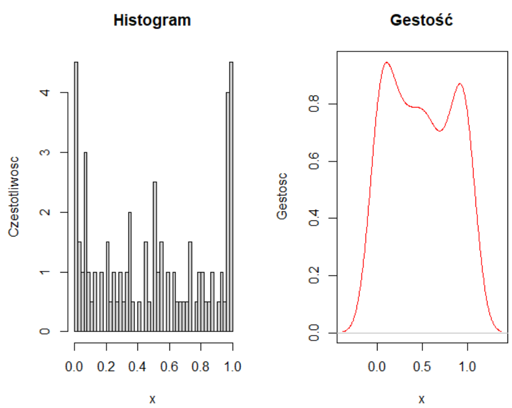
Rysunek 3: Zestawienie wyników miar generatora dla n=1000

	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

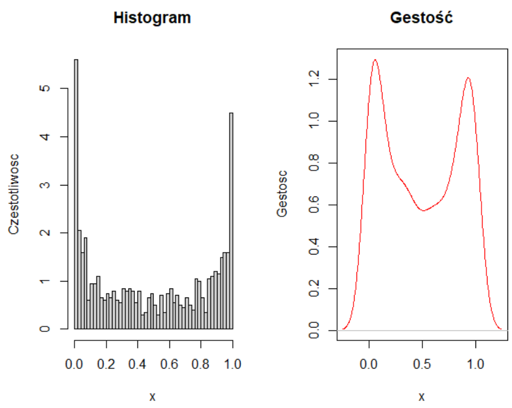
1.1.4 1.4 Histogram i wykres gęstości

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

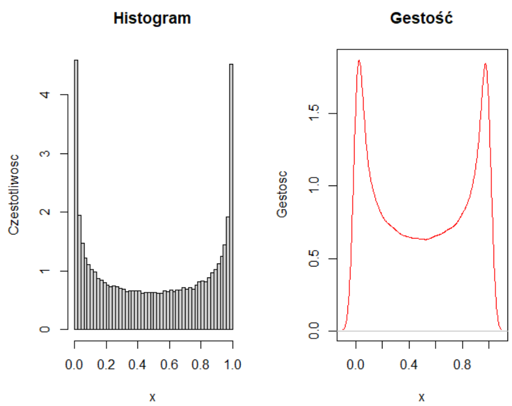


Rysunek 5: Histogram dla n=100

Na histogramie z Rysunku 3 częstotliwość bliska jest 0 i 1. Próbka jednak jest zbyt mała 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



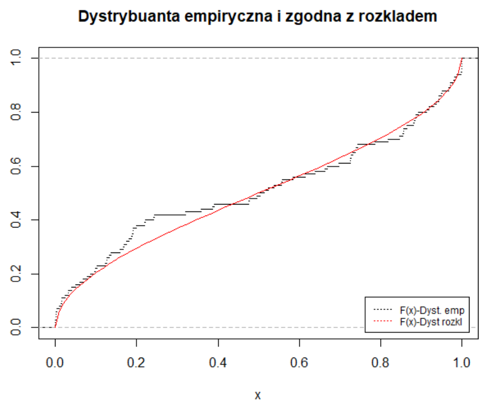
Rysunek 7: Histogram dla n=100000

1.1.5 1.5 Dystrybuanta empiryczna i dystrybuanta wybranego rozkładu

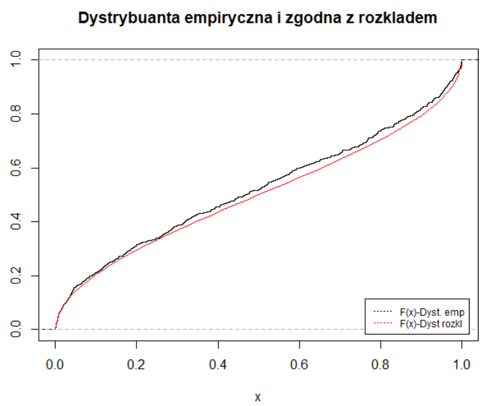
Wykreślić na jednym wykresie: dystrybuantę empiryczną i dystrybuantę wybranego rozkładu

```
dyst_emp_teor <- function(d, d_inv, n, xlims, umin=0, umax=1, tmin=0, tmax=1) {
  plot(ecdf(d_inv(runif(n,umin,umax))),do.points=FALSE,
       main=paste("Dystrybuanta empiryczna i zgodna z rozkladem"),
       ylab = "", xlim = xlims)

  x <- seq(tmin,tmax,0.01)
  lines(x,d(x),type = "l",col="red")
  legend("bottomright",inset=.02, legend=c("F(x)-Dyst. emp", "F(x)-Dyst rozkl"),
         col=c("black", "red"),lty=3, cex=0.75,
         box.lty=1)
}
```

```
par(mfrow=c(1,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^2, xlims = c(0,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^3, xlims = c(0,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^5, xlims = c(0,1))
```

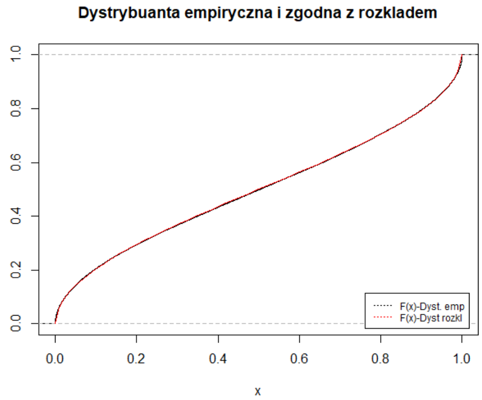


Rysunek 8: Wykres dystrybuanty dla n=100



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 dystrybuanty 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.

```
normalizacja <- function(x) (x - min(x))/(max(x) - min(x))

{
Dane_Arcsin1_normalizowane <- normalizacja(Dane_Arcsin1)
Dane_Arcsin2_normalizowane <- normalizacja(Dane_Arcsin2)
Dane_Arcsin3_normalizowane <- normalizacja(Dane_Arcsin3)
}

{
Dane_Rayleigh1_normalizowane <- normalizacja(Dane_Rayleigh1)
Dane_Rayleigh2_normalizowane <- normalizacja(Dane_Rayleigh2)
Dane_Rayleigh3_normalizowane <- normalizacja(Dane_Rayleigh3)
}
```

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 $p = 0.45$, $k = 0, 1, 2, ..., N$, N to liczba iteracji. Poprzez f^N oznaczamy N-te złożenie mapy.

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

```
#  funkcje mapy chaotycznej z polecenia
ukosna_mapa_namiotu <- function(zk, p=0.45) {
  if(zk >= 0 & zk < p) {
    return(zk/p)
  }definiujemy
  if(zk >= p & zk <= 1) {
    return((1-zk)/(1-p))
  }
  else{
    print('Nieprawidlowa wartosc')
  }
}
# definiujemy N-te zlozenie mapy chaotycznej z polecelnia
ukosna_mapa_namiotu_N_iteracji <- function(N, x0, p = 0.45){
  x = c()
  for (j in 1:length(x0)) {
    x[j] = ukosna_mapa_namiotu(x0[j], p)

    if(N > 1) {
      for(i in 2:N){
        x[j] = ukosna_mapa_namiotu(x[j], p)
      }
    }
  }
  return(x)
}
```

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

```
Dane_Arcsin1_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
     Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
     Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_1=ukosna_mapa_namiotu_N_iteracji(1,
     Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
     Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
     Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_5=ukosna_mapa_namiotu_N_iteracji(5,
     Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
    (12, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
    (12, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_12=ukosna_mapa_namiotu_N_iteracji
    (12, Dane_Arcsin3_normalizowane)
```

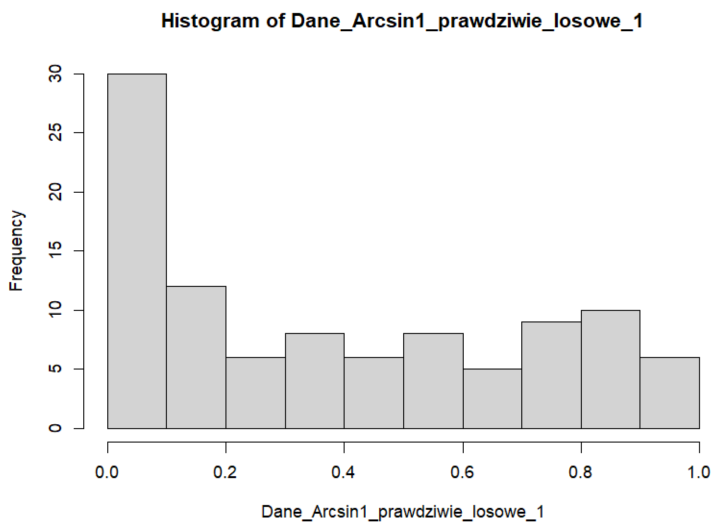
Dla rozkładu Rayleigha

```
Dane_Rayleigh1_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_1<-ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_5<-ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_12<-ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh3_normalizowane)
```

1.2.3 2.3 Histogramy wygenerowanych rozkładów

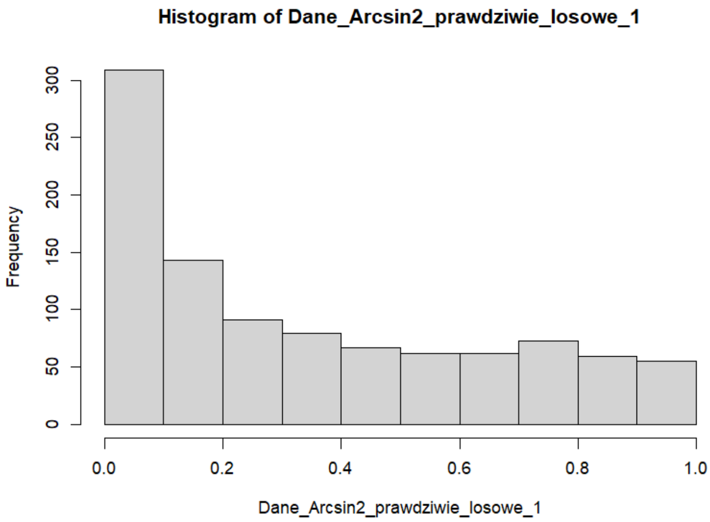
Wykreślamy histogramy wygenerowanych rozkładów.

```
hist(Dane_Arcsin1_prawdziwie_losowe_1)
```



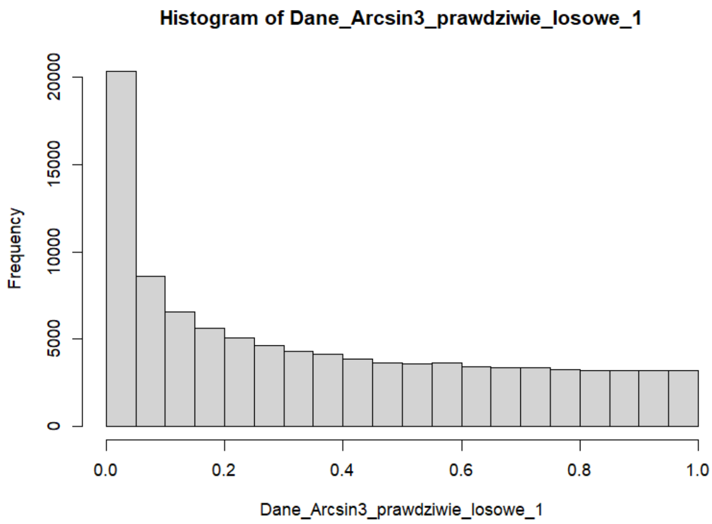
Rysunek 11: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_1)
```



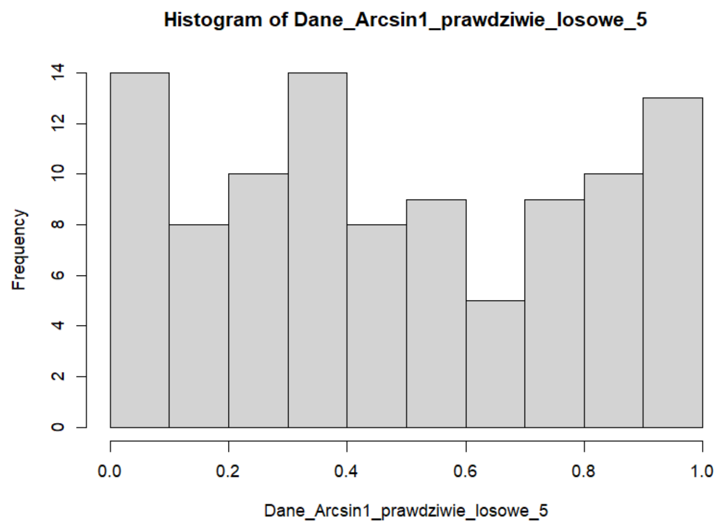
Rysunek 12: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_1)
```



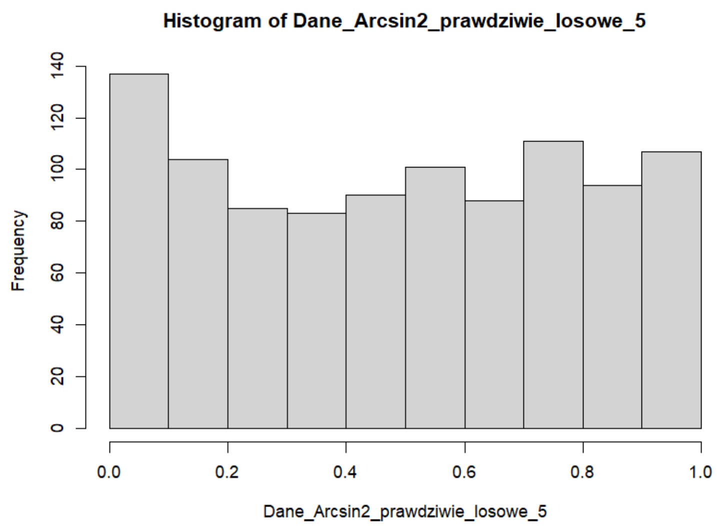
Rysunek 13: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_5)
```



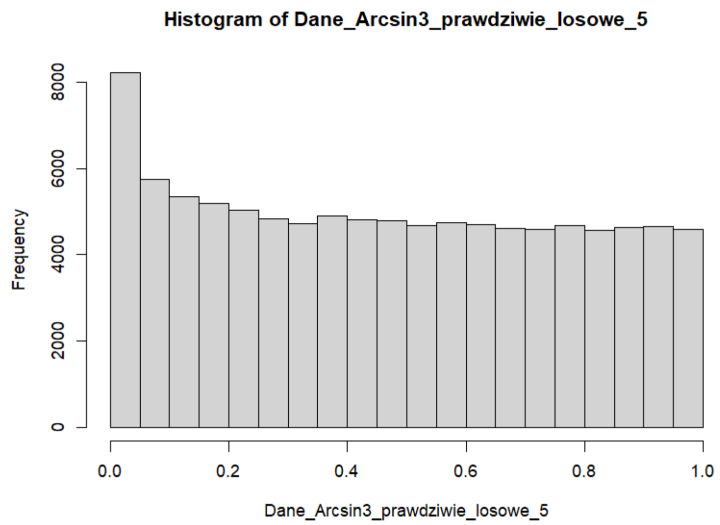
Rysunek 14: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_5)
```



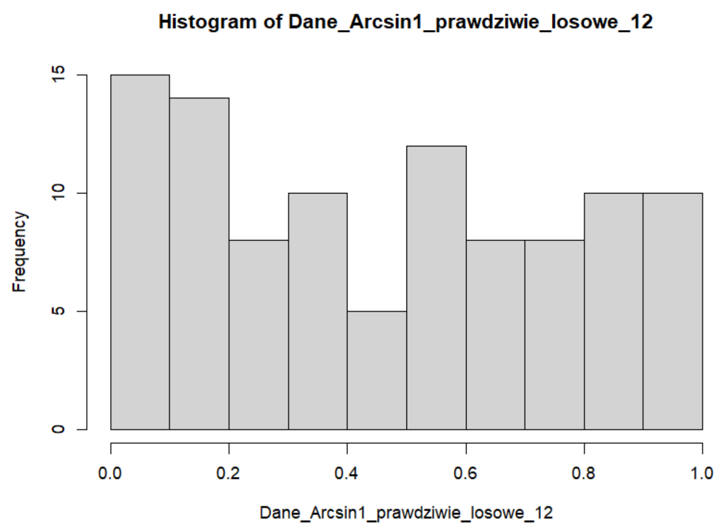
Rysunek 15: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_5)
```



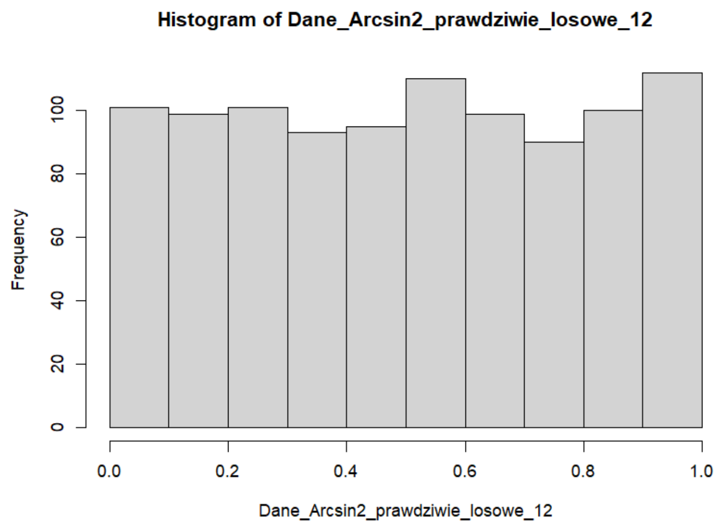
Rysunek 16: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_12)
```



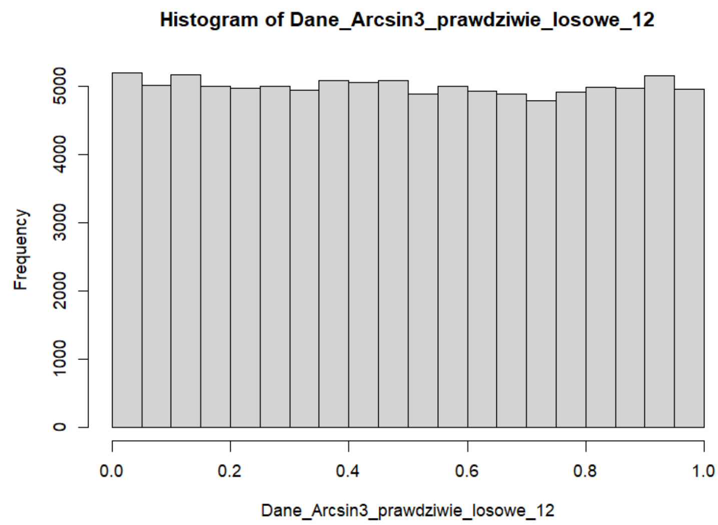
Rysunek 17: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_12)
```



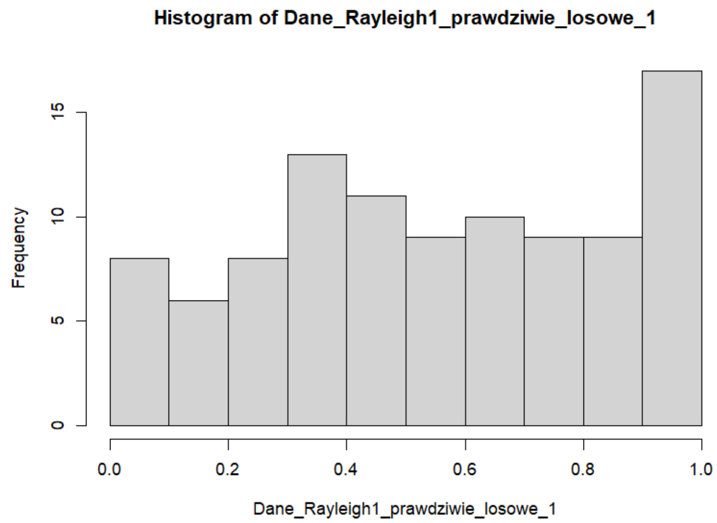
Rysunek 18: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_12)
```



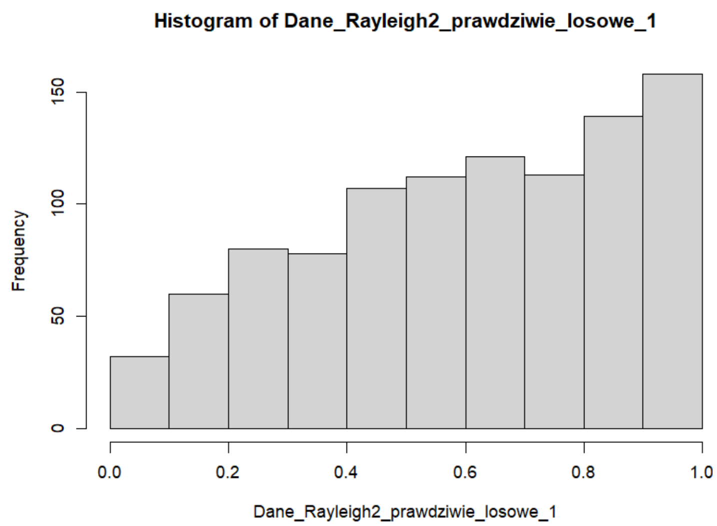
Rysunek 19: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_1)
```



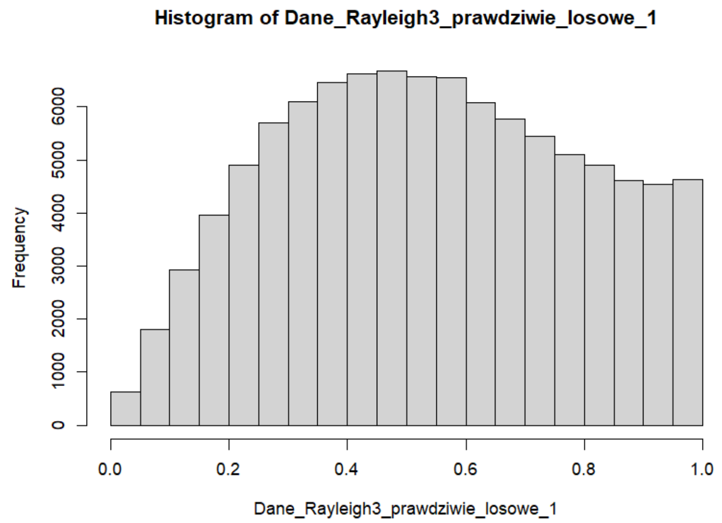
Rysunek 20: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_1)
```



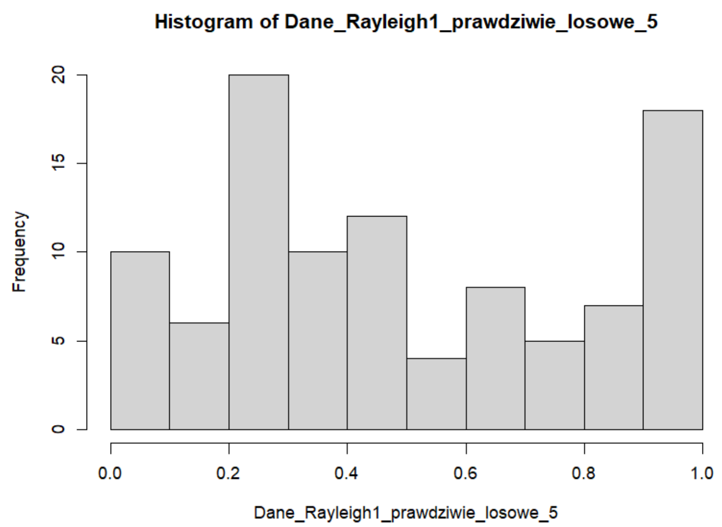
Rysunek 21: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_1)
```



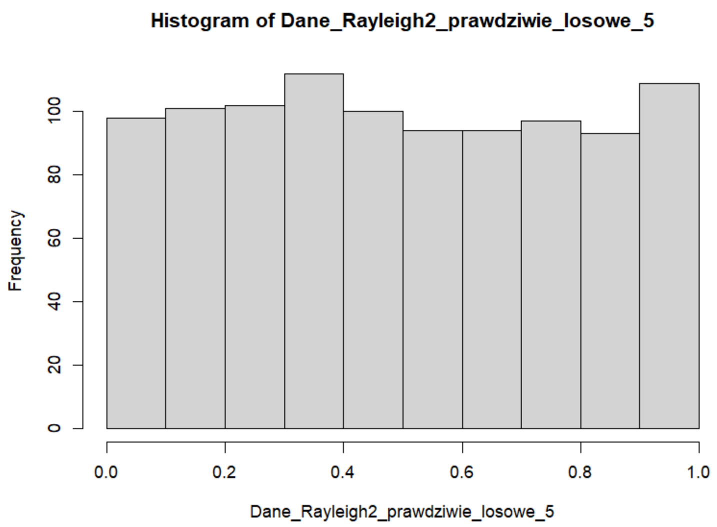
Rysunek 22: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_5)
```



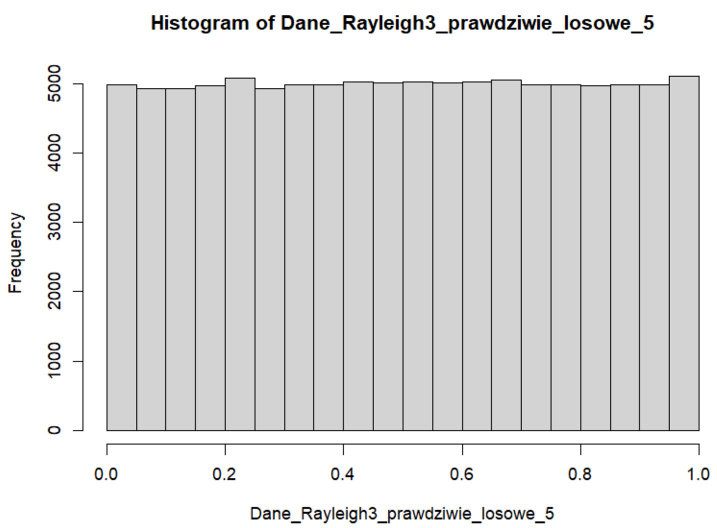
Rysunek 23: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_5)
```



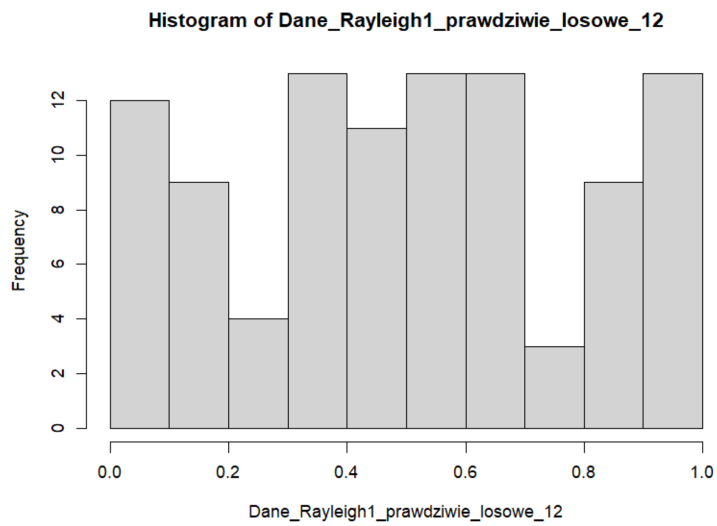
Rysunek 24: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_5)
```



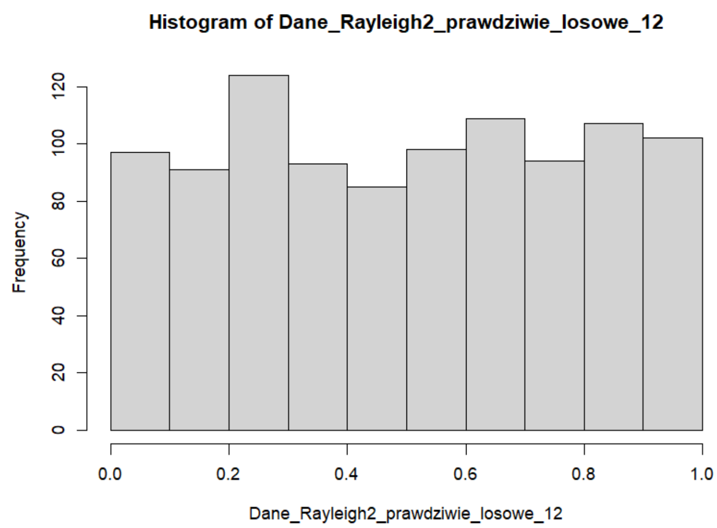
Rysunek 25: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_12)
```



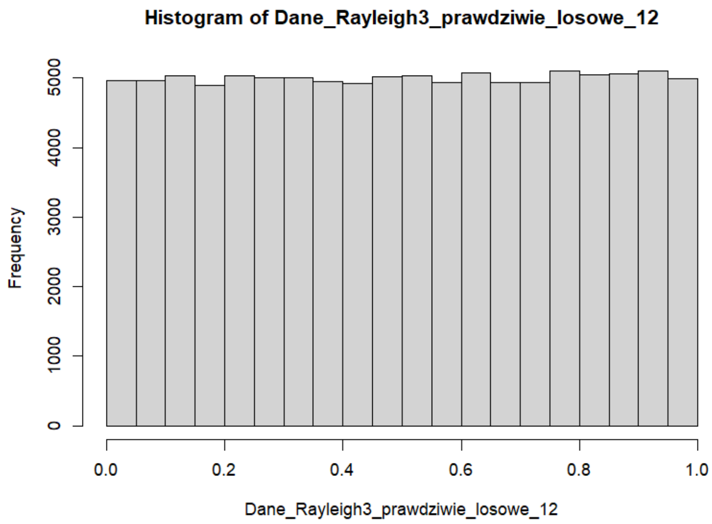
Rysunek 26: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_12)
```



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

```
hist(Dane_Rayleigh3_prawdziwie_losowe_12)
```



Rysunek 28: Próbka 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 złożeń mapy chaotycznej zastosowana do danych tym histogram (a tym samym rozkład prawdopodobieństwa) jest bardziej zbliżony do rozkładu jednostajnego.

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

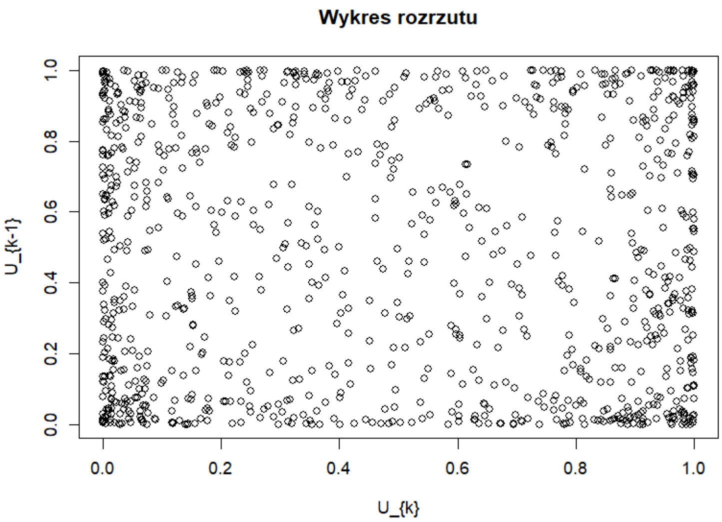
1.2.4 2.4 Rozrzut punktów

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

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

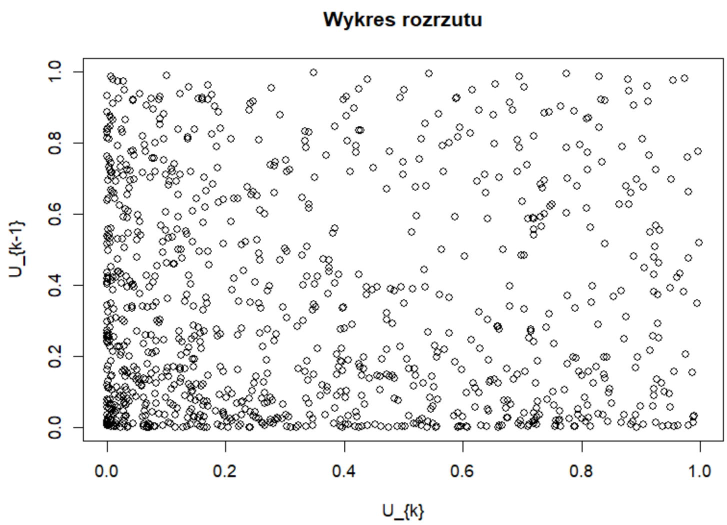
W naszych wykresach ograniczamy się tylko do próbek rozmiaru 10^3 dlatego, że dla 10^2 nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla 10^5 nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



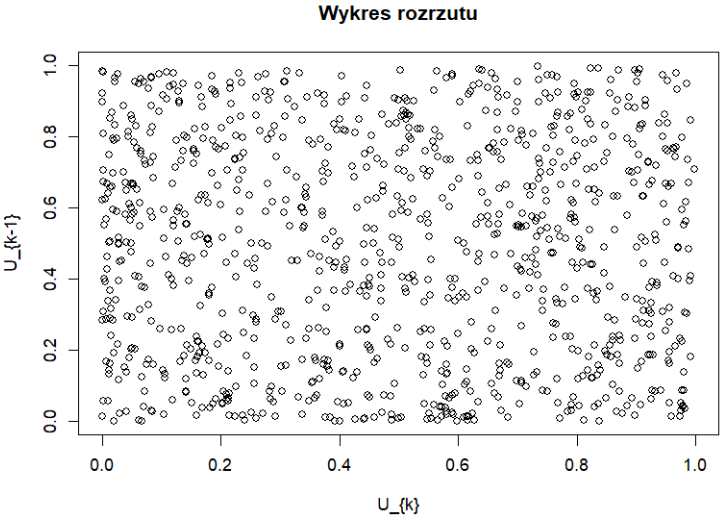
Rysunek 29: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1)
```



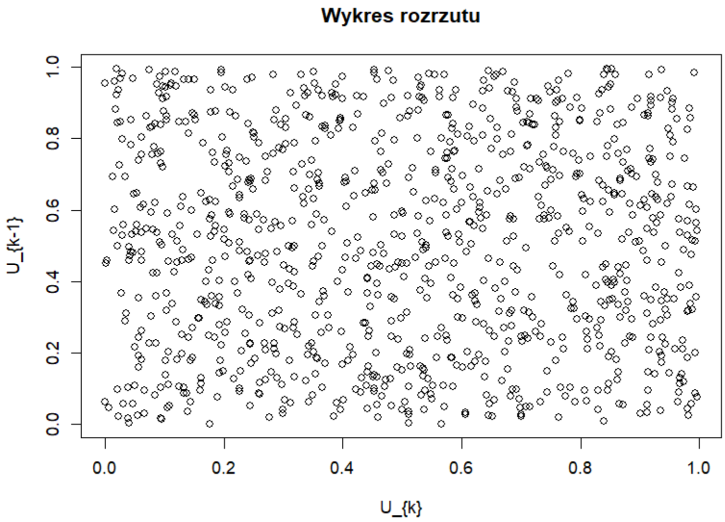
Rysunek 30: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
```



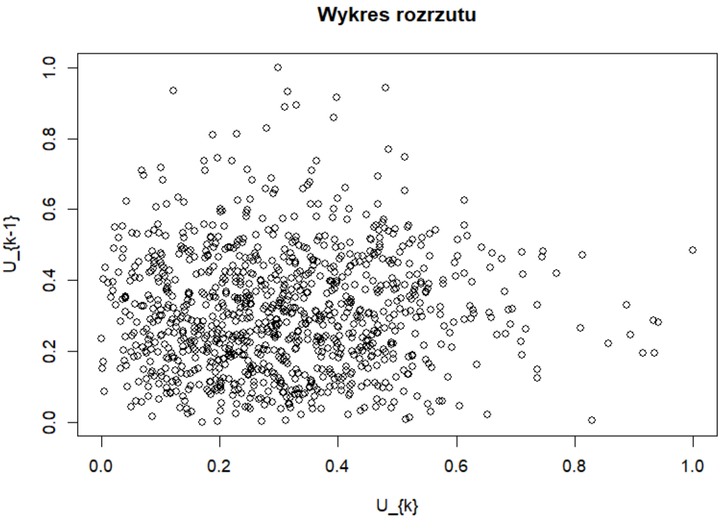
Rysunek 31: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
1 wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12)
```



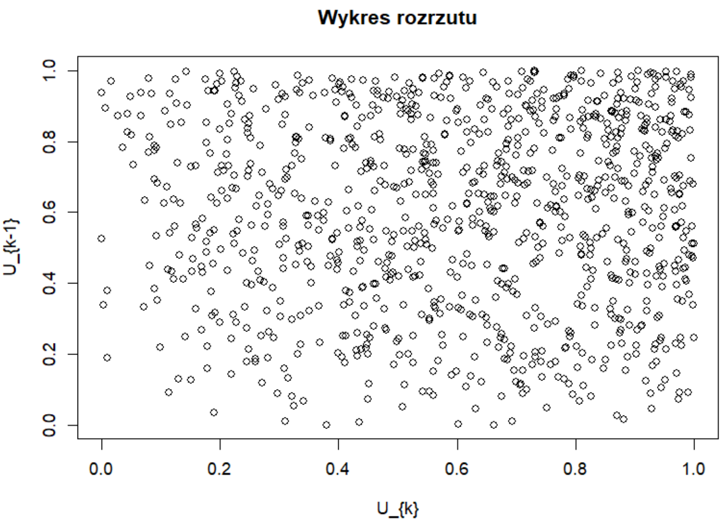
Rysunek 32: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



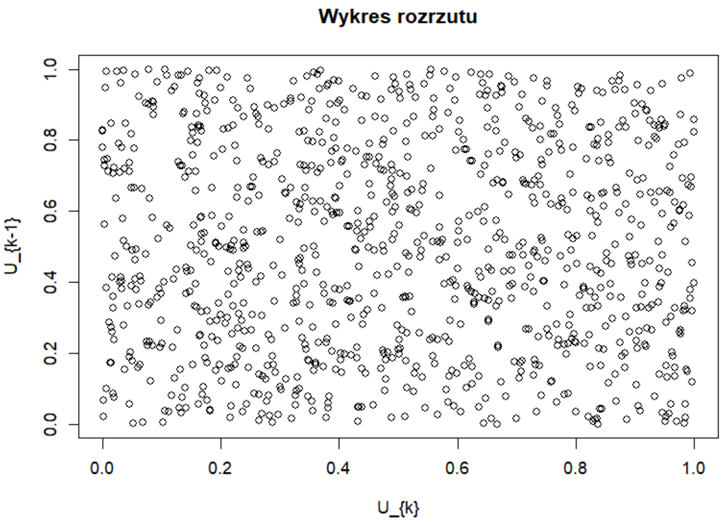
Rysunek 33: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha.

```
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1)
```



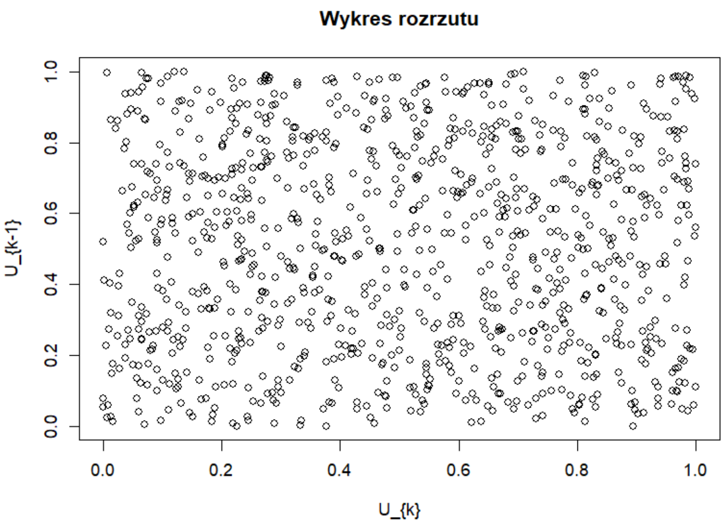
Rysunek 34: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
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.

```
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.

```
# definiujemy funkcje mapy logistycznej
mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))


# definiujemy N-te zlozenie mapy chaotycznej
mapa_logistyczna_N_iteracji <- function(N, x0, r=3.8){
  x = c()
  for (j in 1:length(x0)) {

    x[j] = mapa_logistyczna(x0[j], r)
    if(N > 1){
      for(i in 2:N){
        x[j] = mapa_logistyczna(x[j], r)
      }
    }
  }
  return(x)
}


# definiujemy funkcje mapy kwadratowej
mapa_kwadratowa <- function(xk, a=4, b=0.5) return(b - a*xk^2)


# definiujemy N-te zlozenie mapy chaotycznej
mapa_kwadratowa_N_iteracji <- function(N, x0, a=4, b=0.5){
  x = c()
  for (j in 1:length(x0)) {

    x[j] = mapa_kwadratowa(x0[j], a, b)
    if(N>1){
      for(i in 2:N){
        x[j] = mapa_kwadratowa(x[j], a, b)
      }
    }
  }
  return(x)
}
```

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 złożeń mapy chaotycznej ($N \in \{1, 5, 12\}$) ciągi liczb pseudolosowych.
Dla rozkładu arcusa sinusa

```
Dane_Arcsin1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
    _iteracji(1, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
    _iteracji(1, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N
    _iteracji(1, Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
    _iteracji(5, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
    _iteracji(5, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N
    _iteracji(5, Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
    N_iteracji(12, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
    N_iteracji(12, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_
    N_iteracji(12, Dane_Arcsin3_normalizowane)
}
```

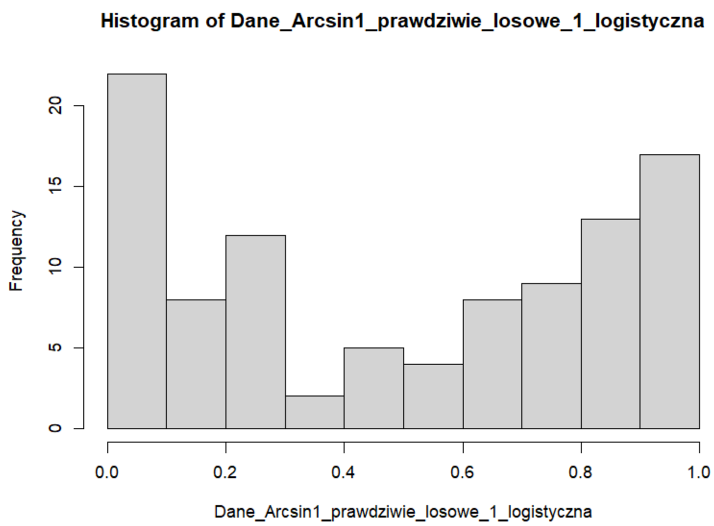
Dla rozkładu Rayleigha

```
Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh3_normalizowane)
```

1.3.3 3.3 Histogramy wygenerowanych rozkładów

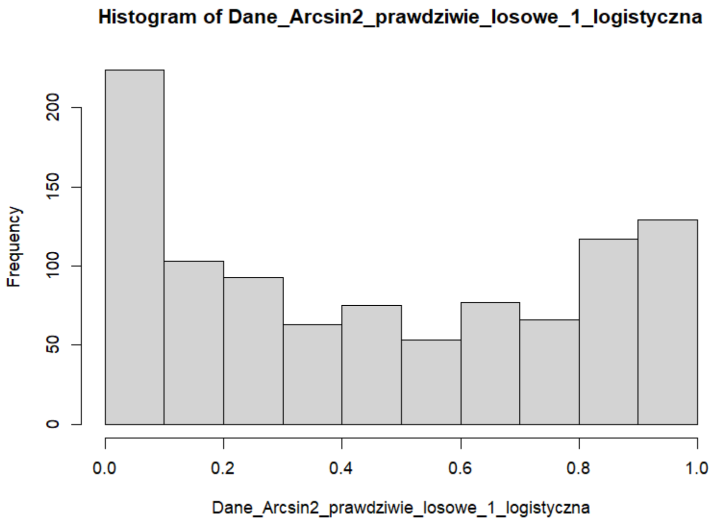
Wykreślamy histogramy wygenerowanych rozkładów.

```
hist(Dane_Arcsin1_prawdziwie_losowe_1_logistyczna)
```



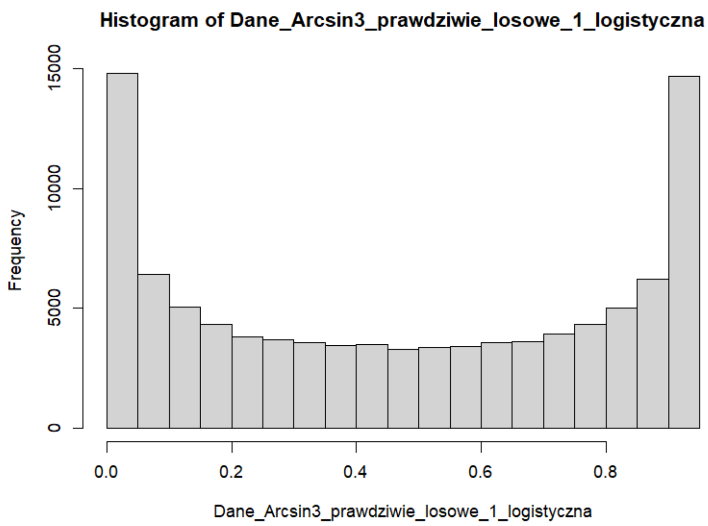
Rysunek 37: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
```



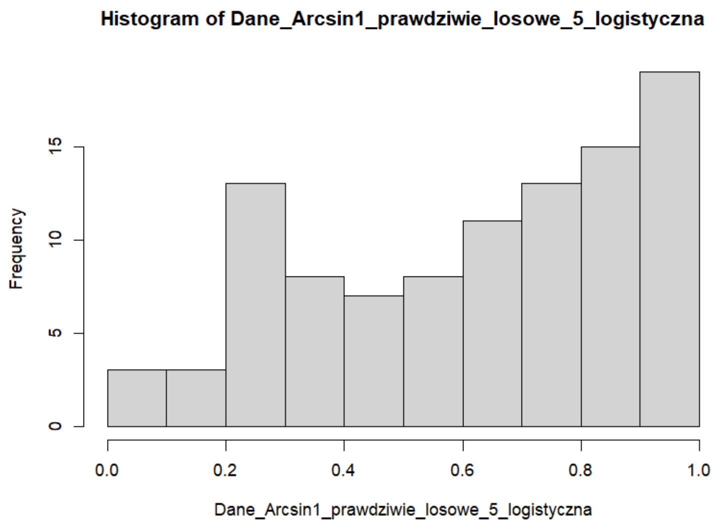
Rysunek 38: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_1_logistyczna)
```



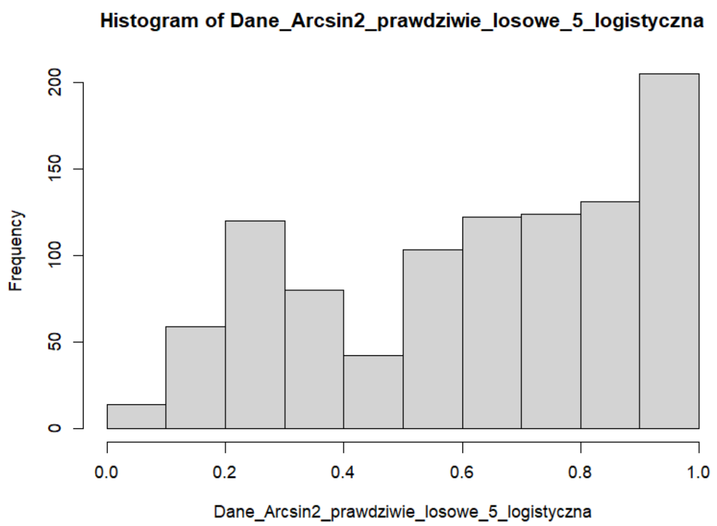
Rysunek 39: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_5_logistyczna)
```



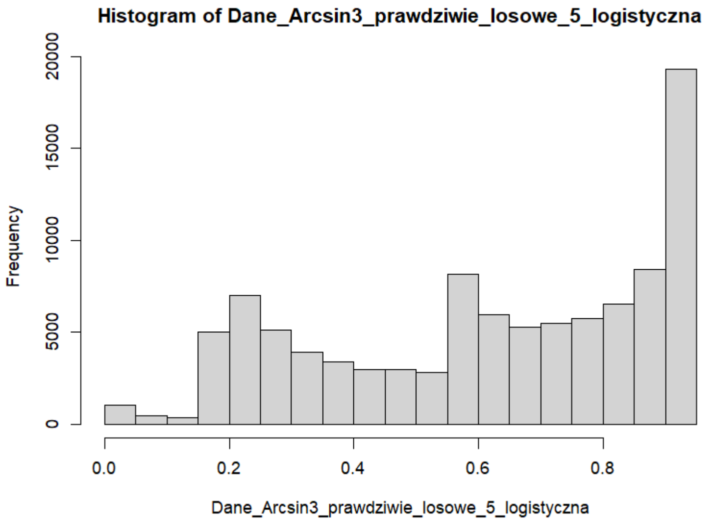
Rysunek 40: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
```



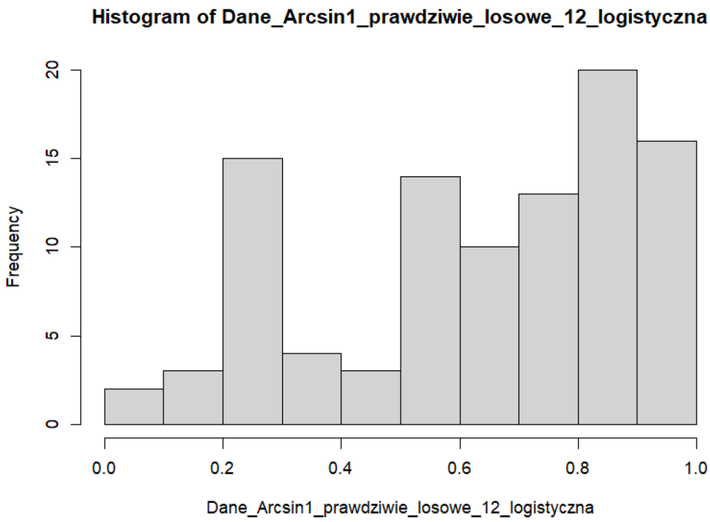
Rysunek 41: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_5_logistyczna)
```



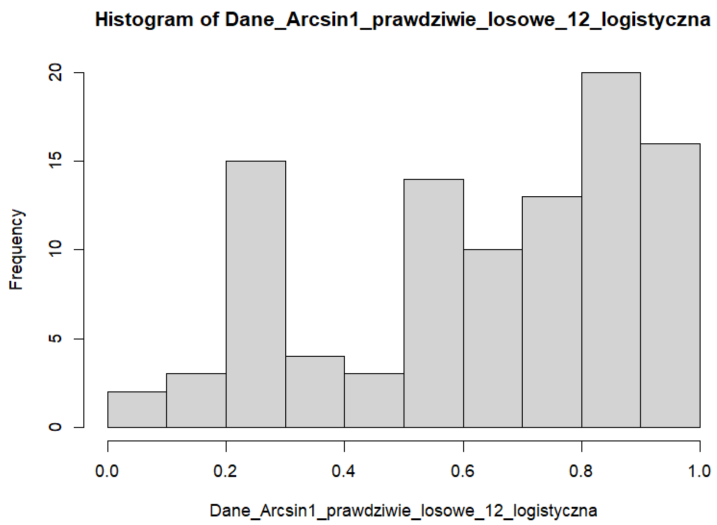
Rysunek 42: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_12_logistyczna)
```



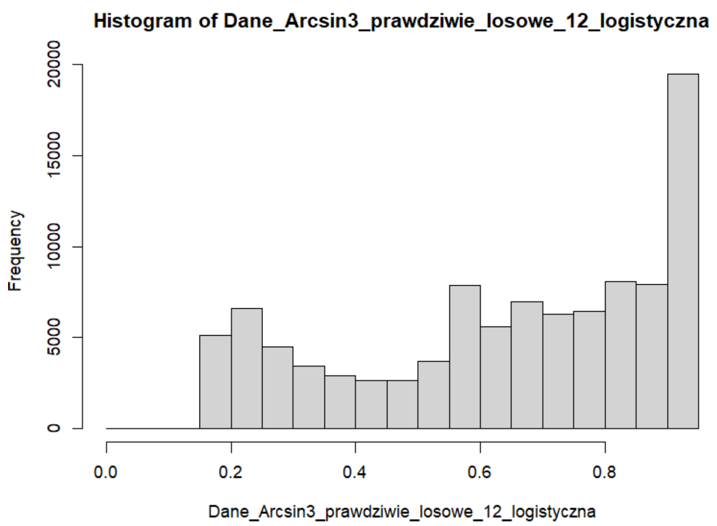
Rysunek 43: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
```



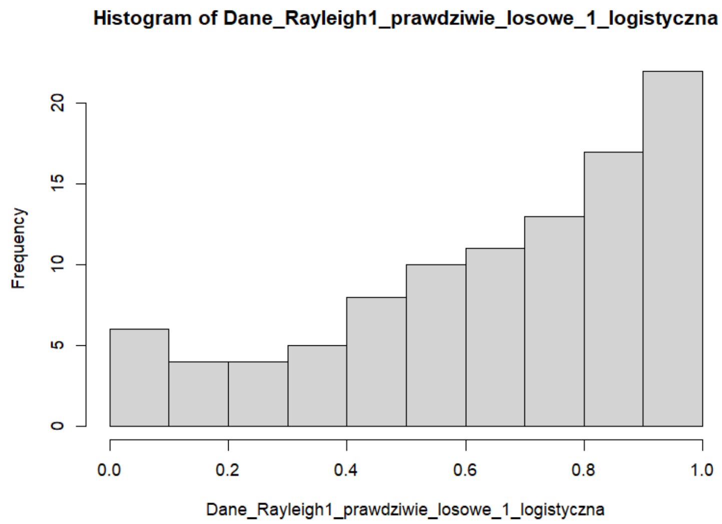
Rysunek 44: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_12_logistyczna)
```



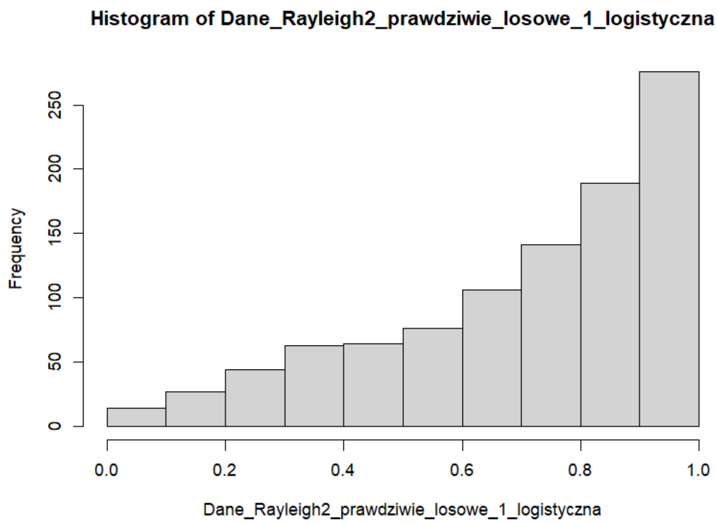
Rysunek 45: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna)
```



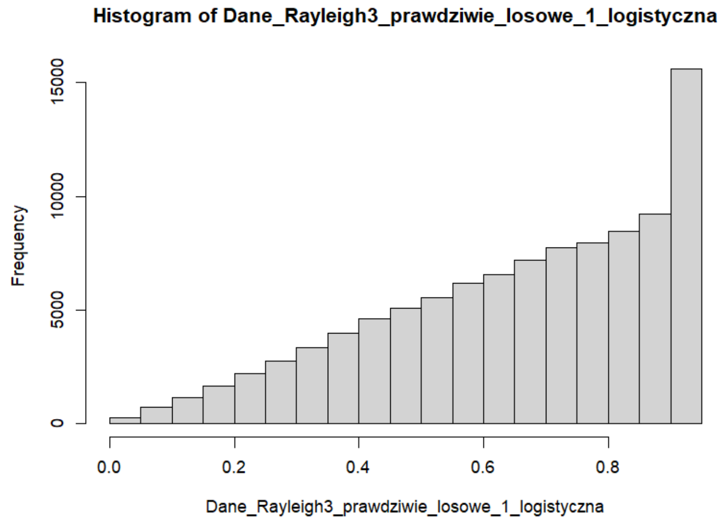
Rysunek 46: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
```



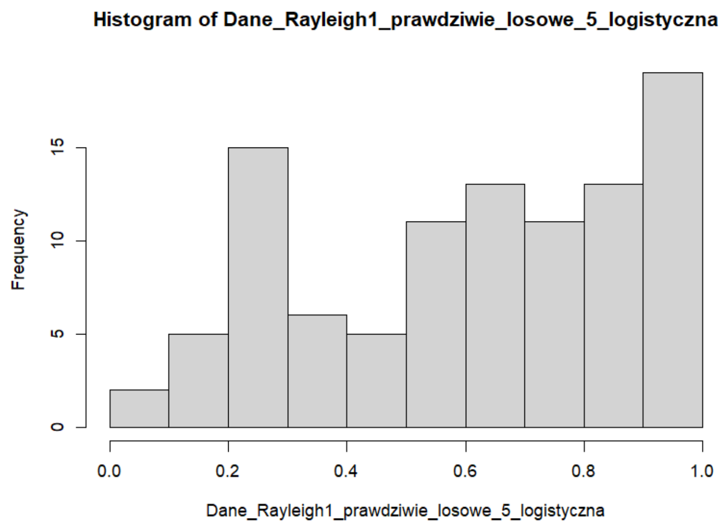
Rysunek 47: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna)
```



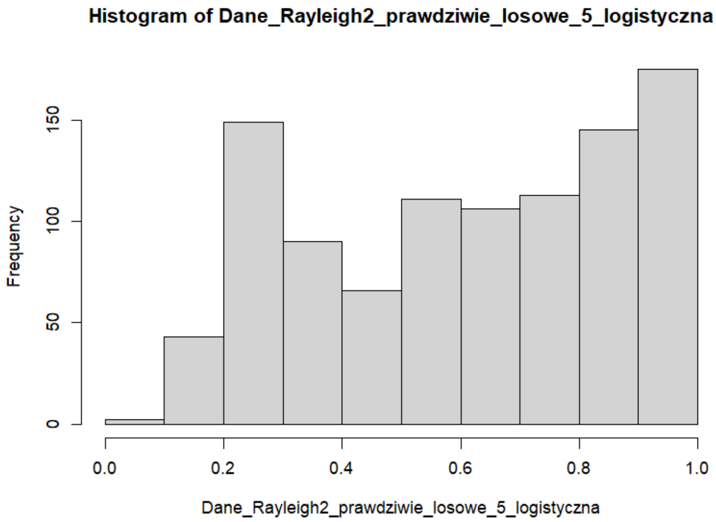
Rysunek 48: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna)
```



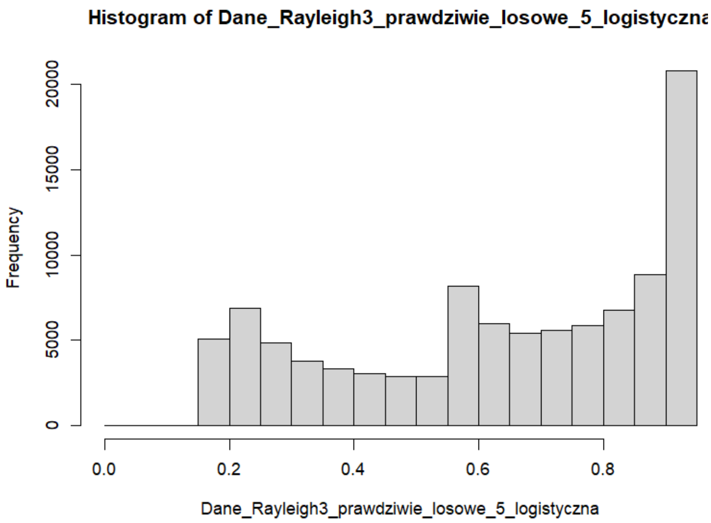
Rysunek 49: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
```



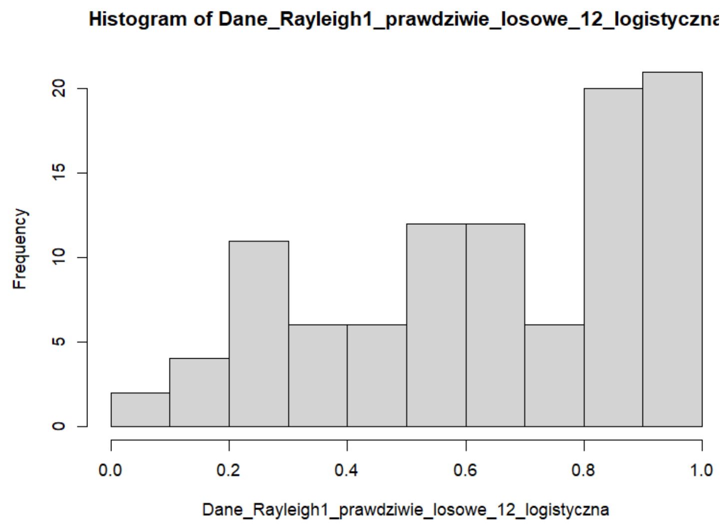
Rysunek 50: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna)
```



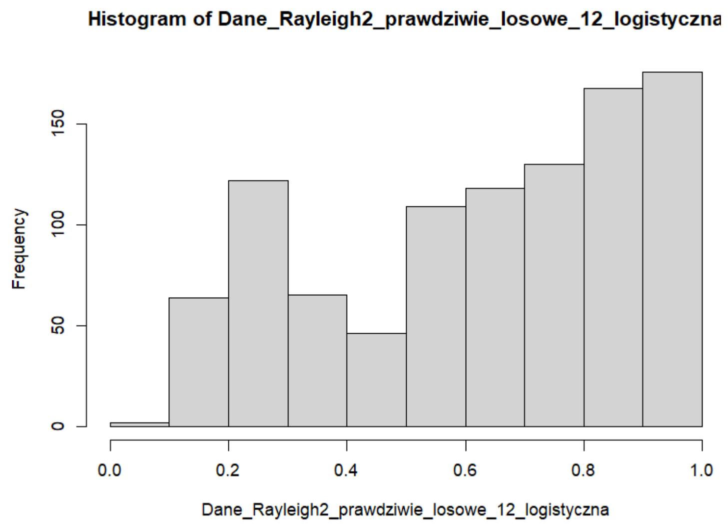
Rysunek 51: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna)
```



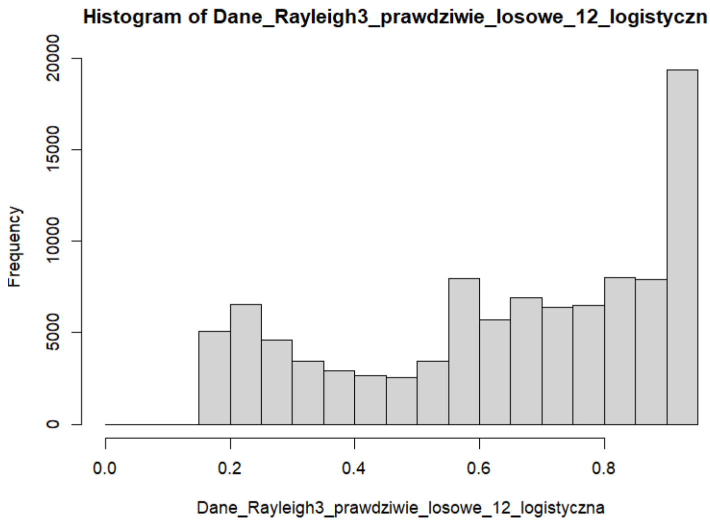
Rysunek 52: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
```



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

```
hist(Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna)
```



Rysunek 54: Próbka 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

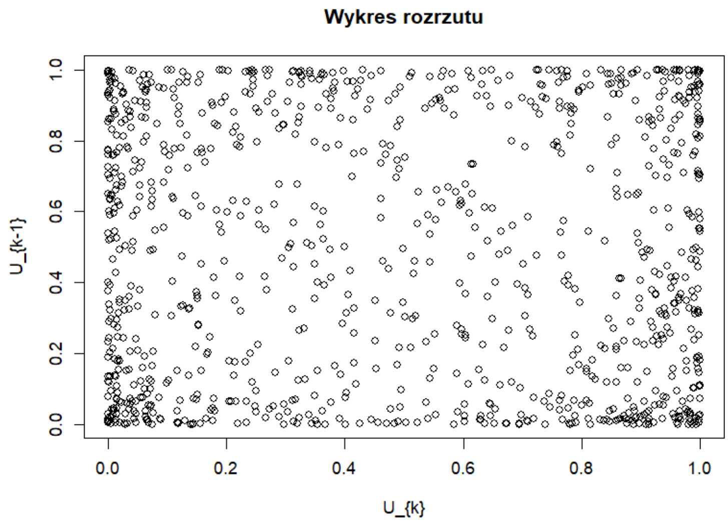
Z mapy logistycznej nie otrzymujemy rozkładu jednostajnego.

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.

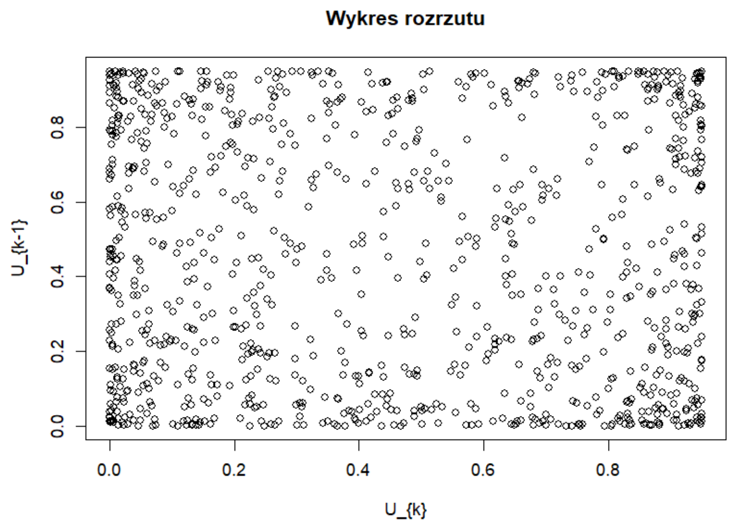
W naszych wykresach ograniczamy się tylko do próbek rozmiaru 10^3 dlatego, że dla 10^2 nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla 10^5 nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



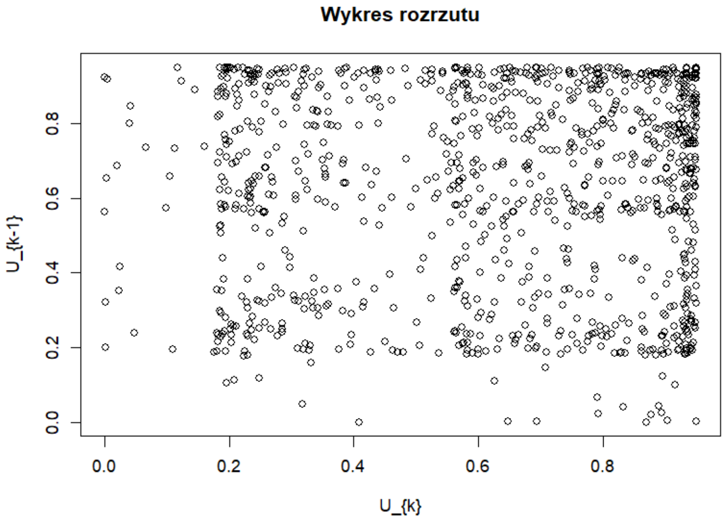
Rysunek 55: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
```



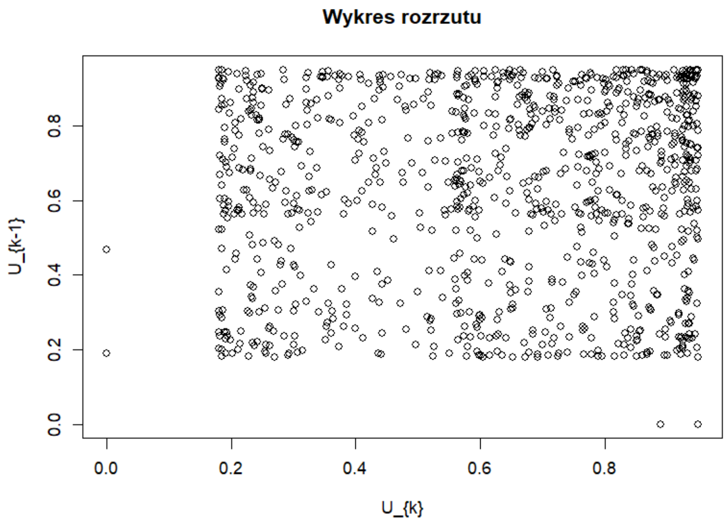
Rysunek 56: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
```



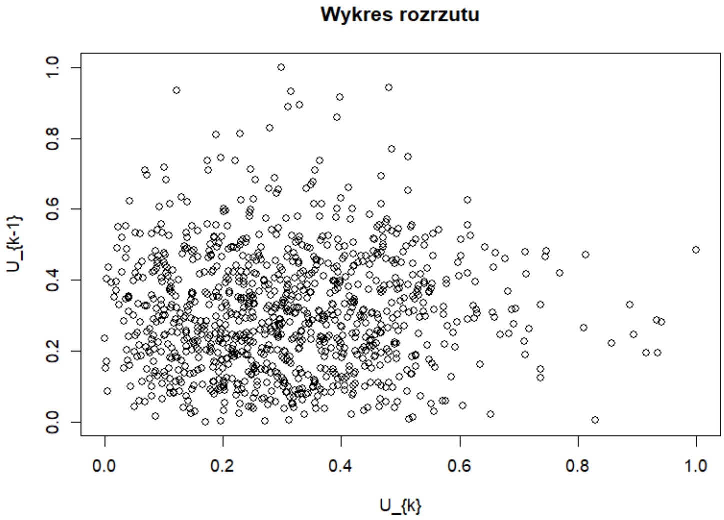
Rysunek 57: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
```



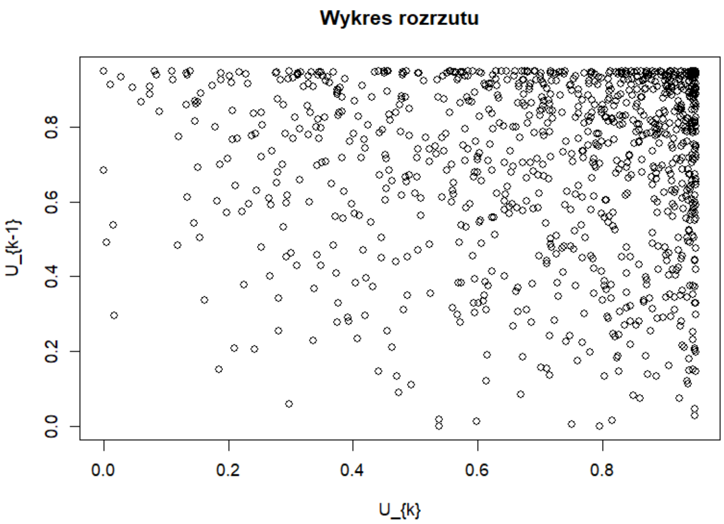
Rysunek 58: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



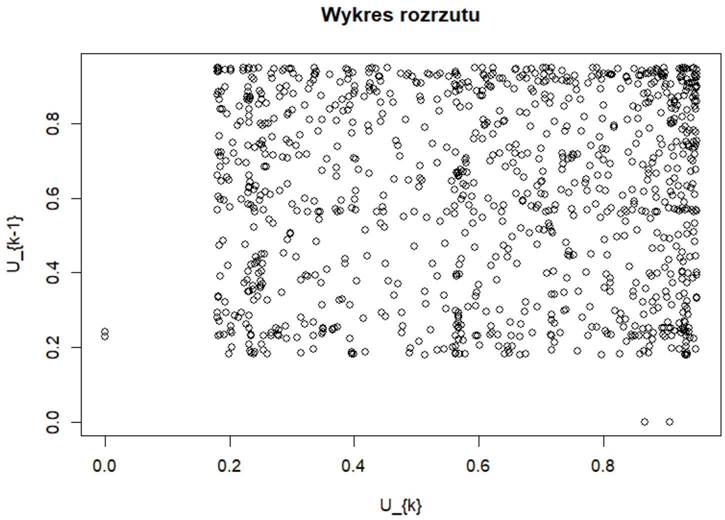
Rysunek 59: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha.

```
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
```



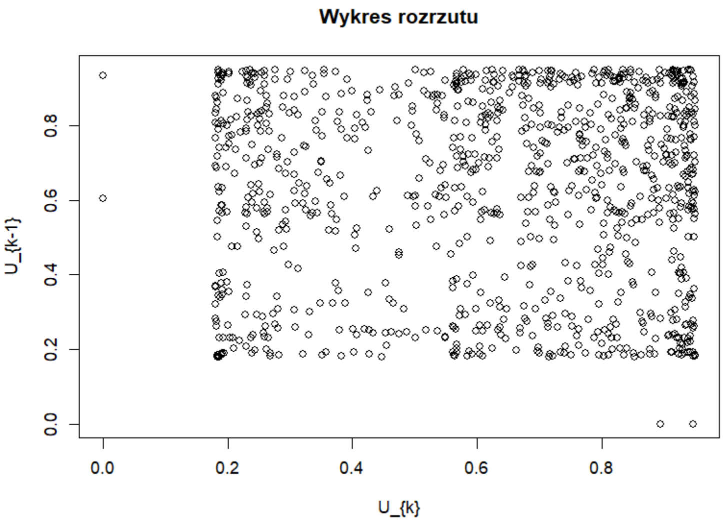
Rysunek 60: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
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.

```
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 złożeń mapy chaotycznej ($N \in \{1, 5, 12\}$) ciągi liczb pseudolosowych.
Dla rozkładu arcusa sinusa

```
Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin2_normalizowane)
 Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin3_normalizowane)
Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin1_normalizowane)
Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin2_normalizowane)
Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin3_normalizowane)
```

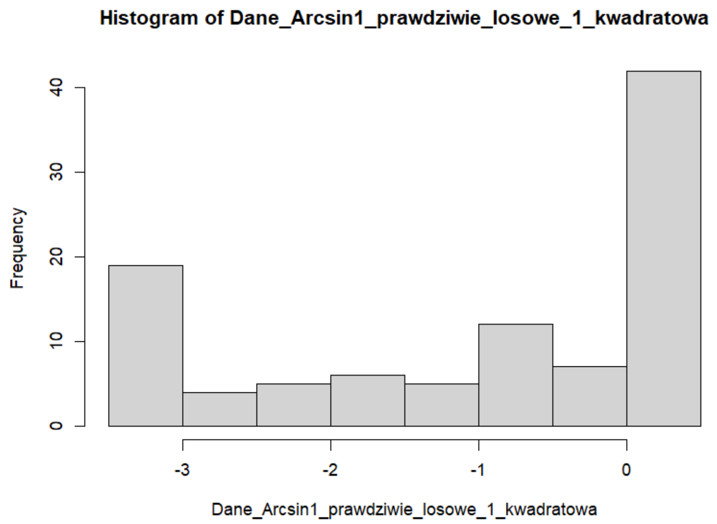
Dla rozkładu Rayleigha

```
Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
    _iteracji(1, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
    _iteracji(1, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N
    _iteracji(1, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
    _iteracji(5, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
    _iteracji(5, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N
    _iteracji(5, Dane_Rayleigh3_normalizowane)
Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
    N_iteracji(12, Dane_Rayleigh1_normalizowane)
Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
    N_iteracji(12, Dane_Rayleigh2_normalizowane)
Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_
    N_iteracji(12, Dane_Rayleigh3_normalizowane)
```

1.3.6 3.3 Histogramy wygenerowanych rozkładów

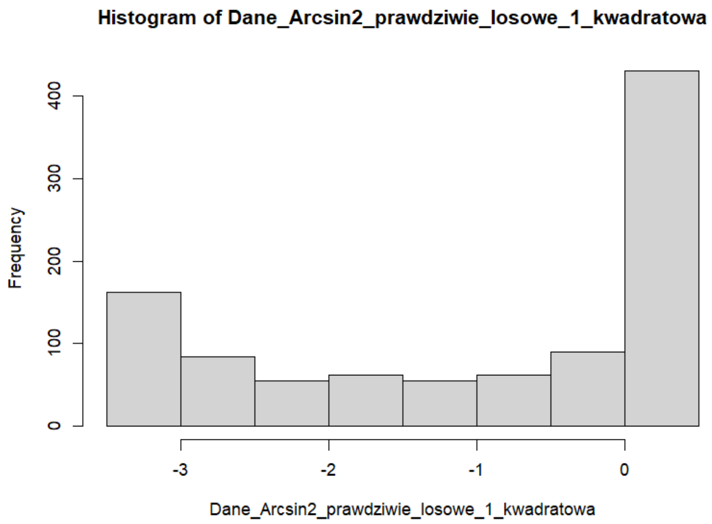
Wykreślamy histogramy wygenerowanych rozkładów.

```
hist(Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa)
```



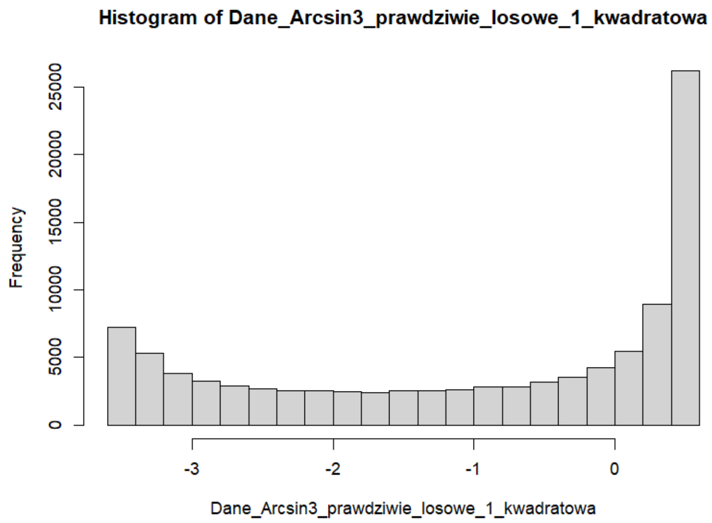
Rysunek 63: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
```



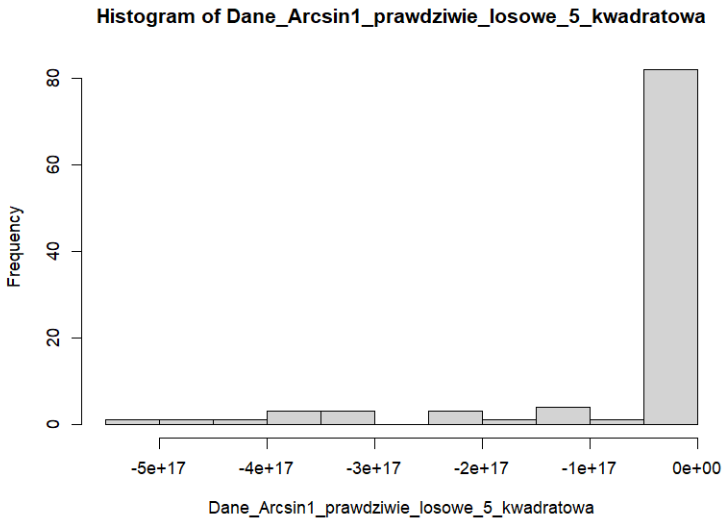
Rysunek 64: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa)
```



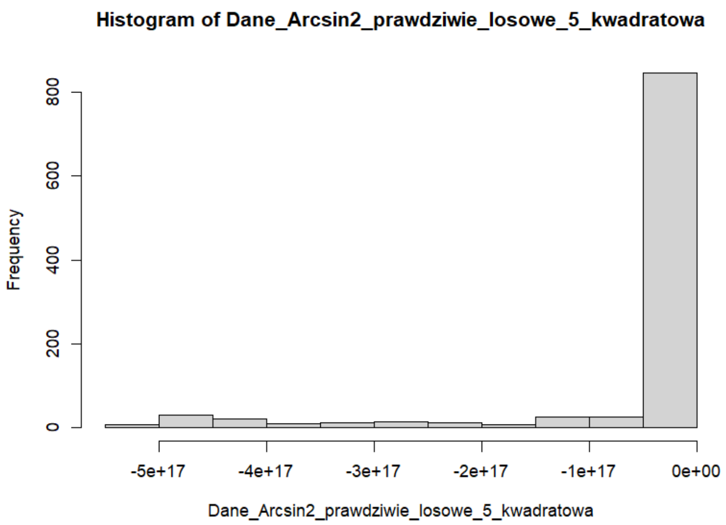
Rysunek 65: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa)
```



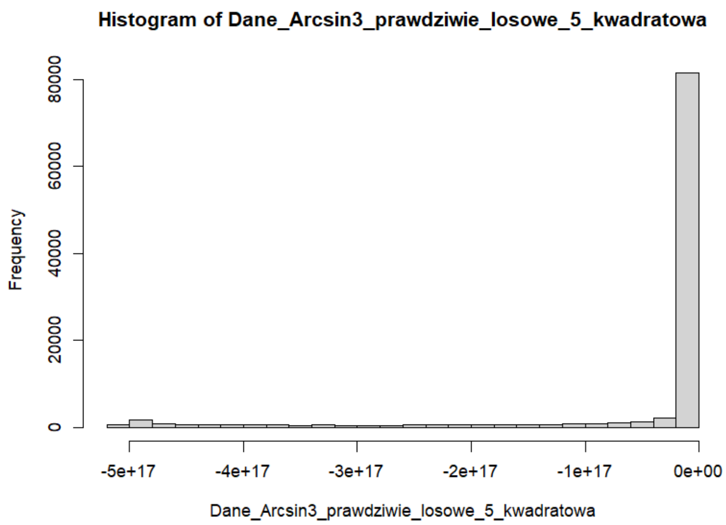
Rysunek 66: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
```



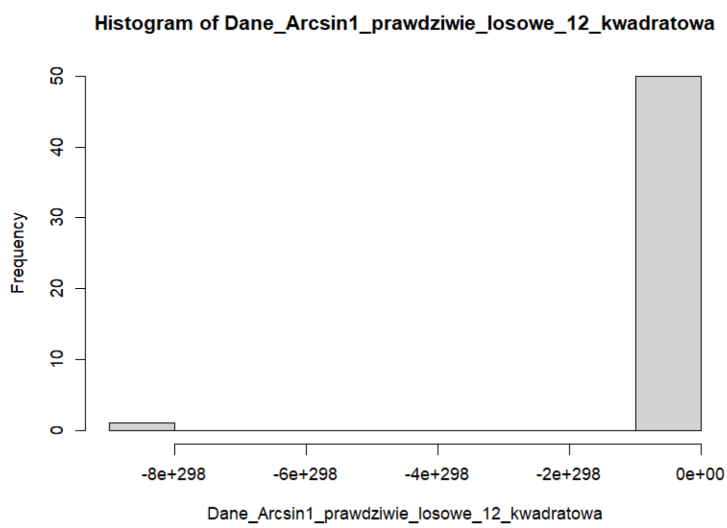
Rysunek 67: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa)
```



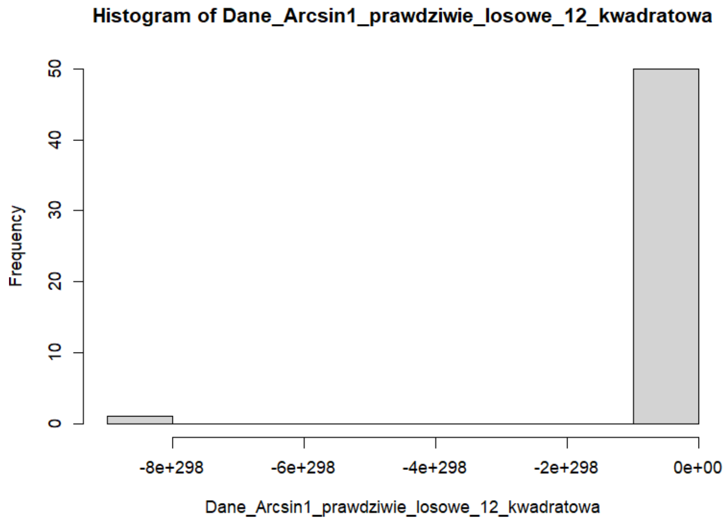
Rysunek 68: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
```



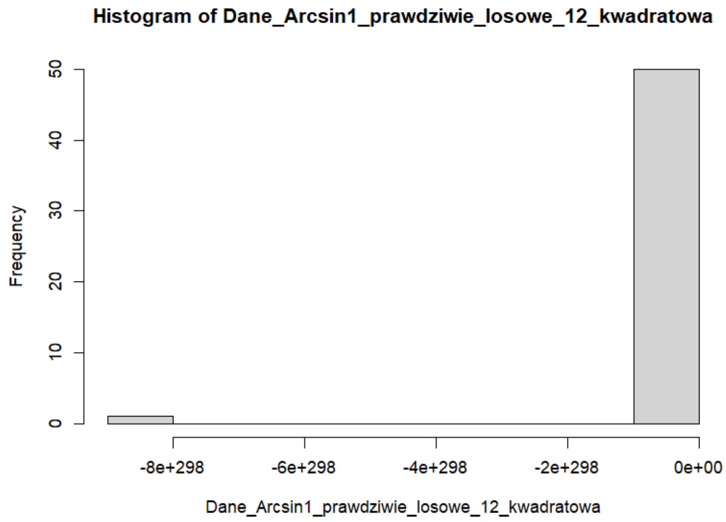
Rysunek 69: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)
```



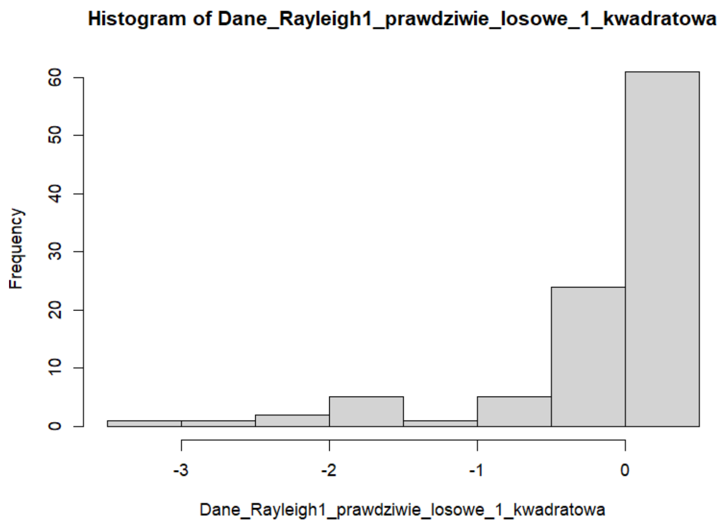
Rysunek 70: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^3 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa)
```



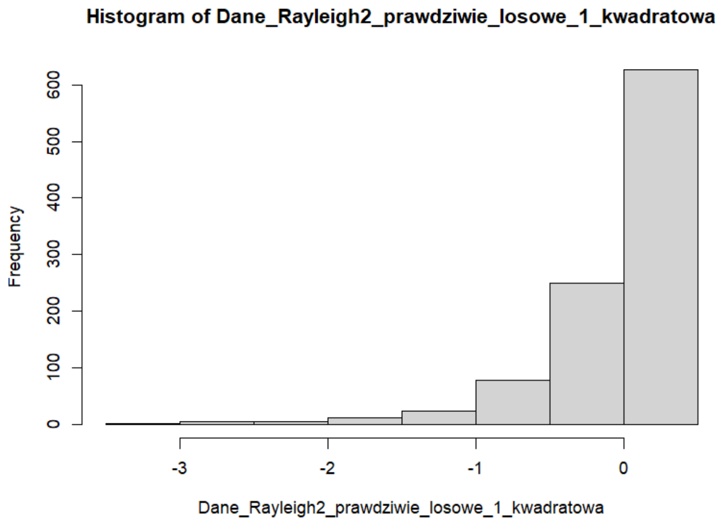
Rysunek 71: Próbka z normalizowanego rozkładu arcusa sinusa o rozmiarze 10^5 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa)
```



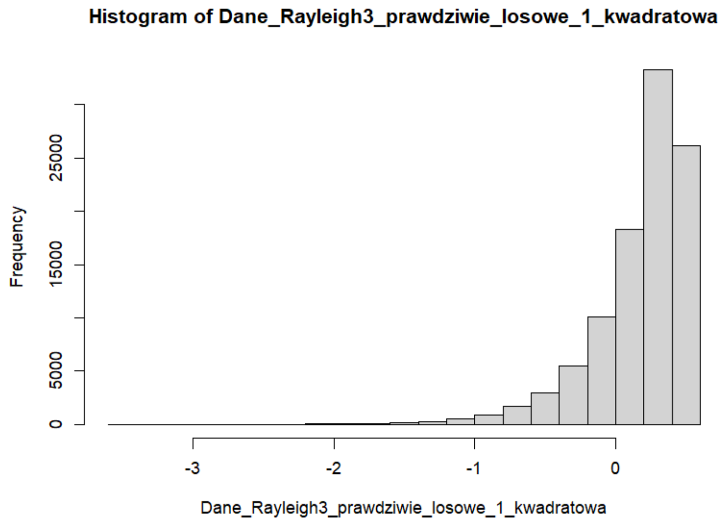
Rysunek 72: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
```



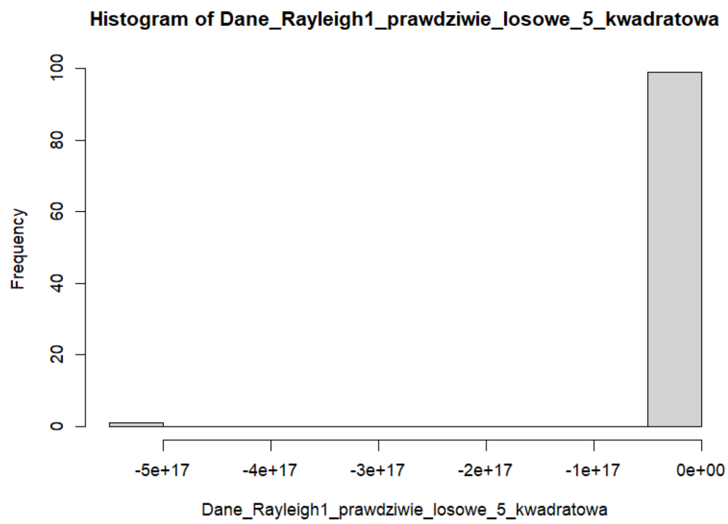
Rysunek 73: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa)
```



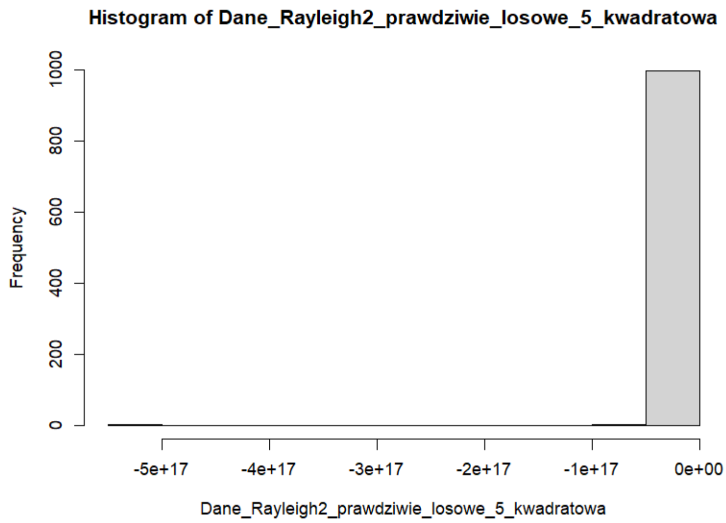
Rysunek 74: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pojedynczym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa)
```



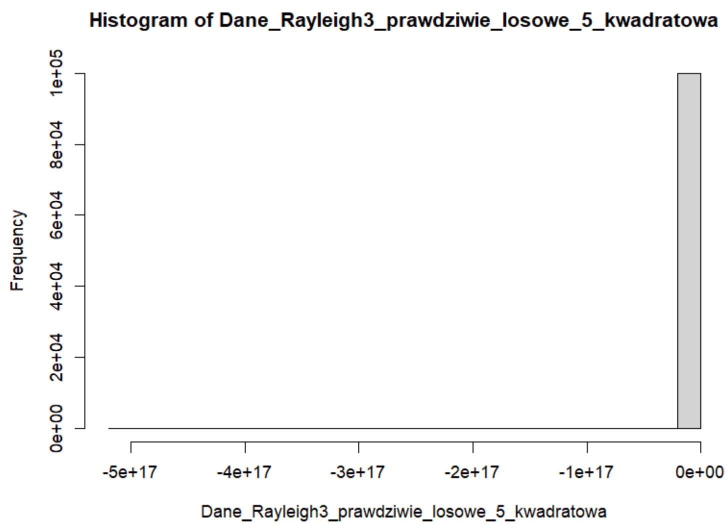
Rysunek 75: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
```



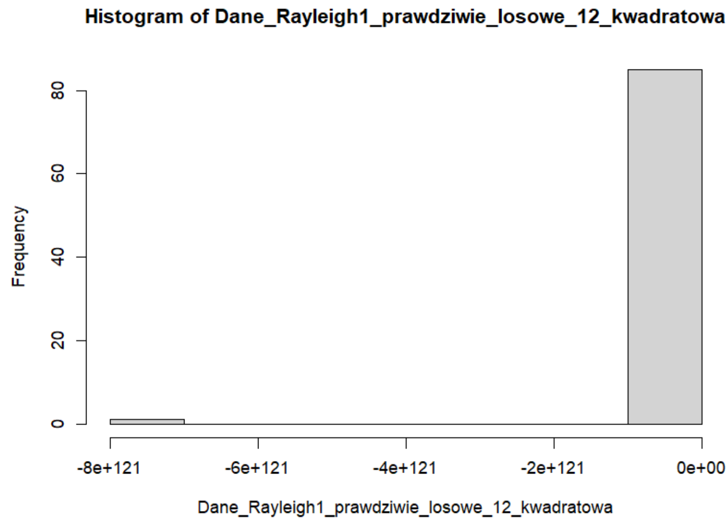
Rysunek 76: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^3 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa)
```



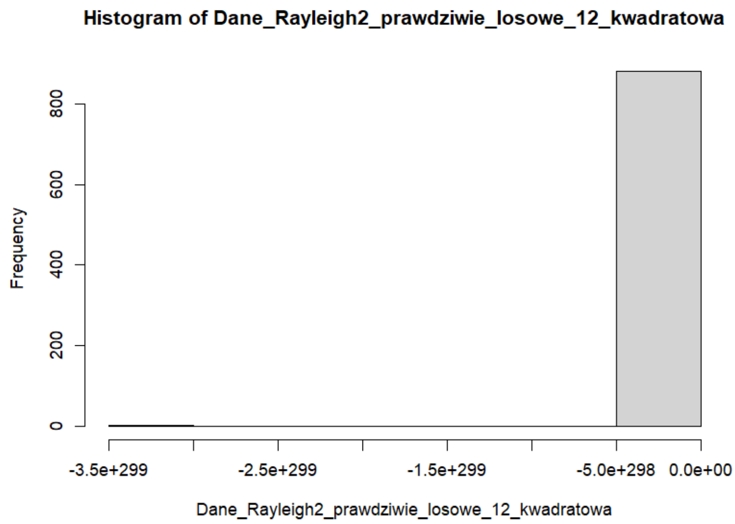
Rysunek 77: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^5 i pięciokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa)
```



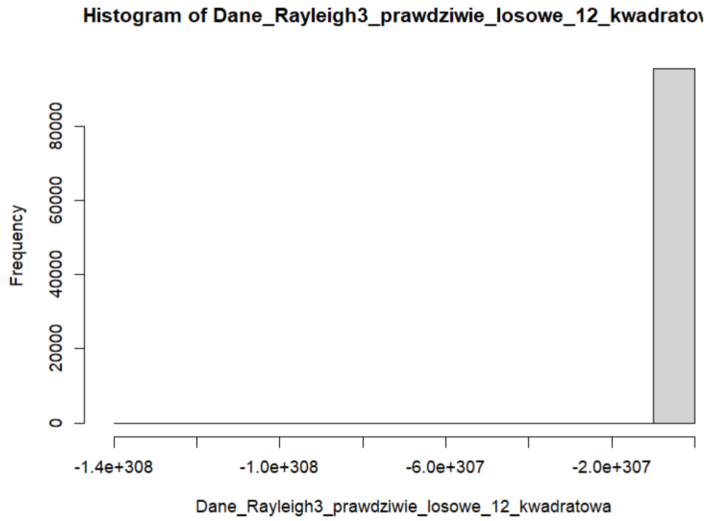
Rysunek 78: Próbka z normalizowanego rozkładu Rayleigha o rozmiarze 10^2 i dwunastokrotnym złożeniu mapy chaotycznej.

```
hist(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
```



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

```
hist(Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa)
```



Rysunek 80: Próbka 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

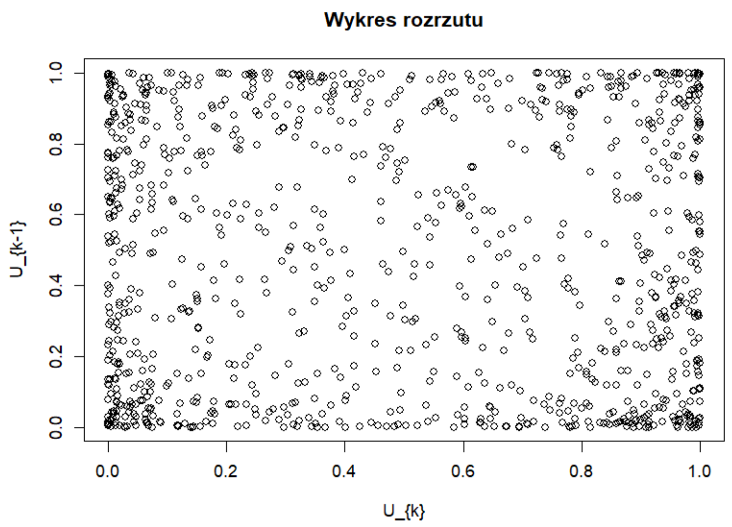
Z mapy kwadratowej nie uzyskujemy rozkładu jednostajnego, wszystkie punkty zbiegają do jednej wartości.

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.

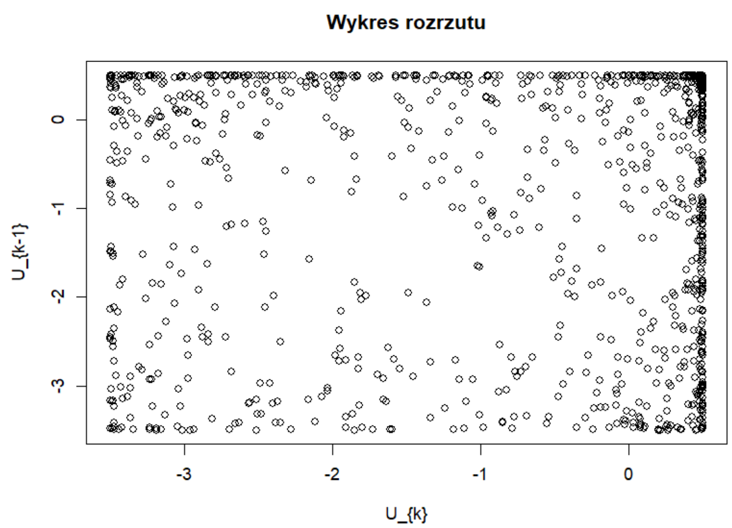
W naszych wykresach ograniczamy się tylko do próbek rozmiaru 10^3 dlatego, że dla 10^2 nie da się zaobserwować jednorodności wykresu z uwagi na zbyt małą liczbę obserwacji, a dla 10^5 nie da się z uwagi na zbyt dużą liczbę obserwacji.

```
wykres_rozrzutu(Dane_Arcsin2_normalizowane)
```



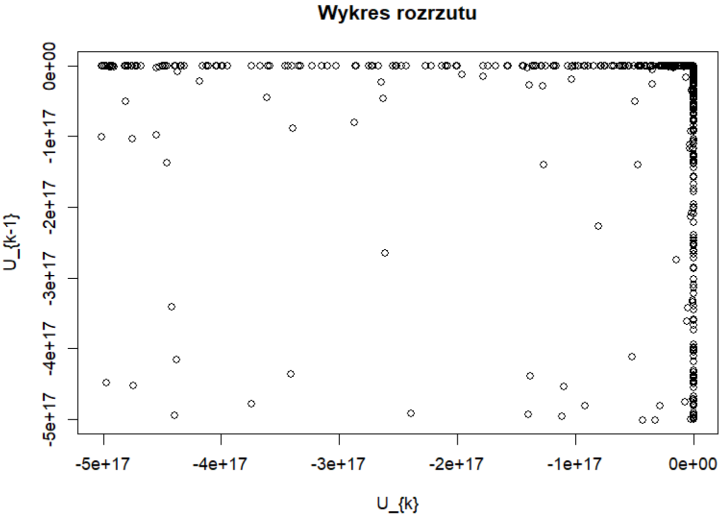
Rysunek 81: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
```



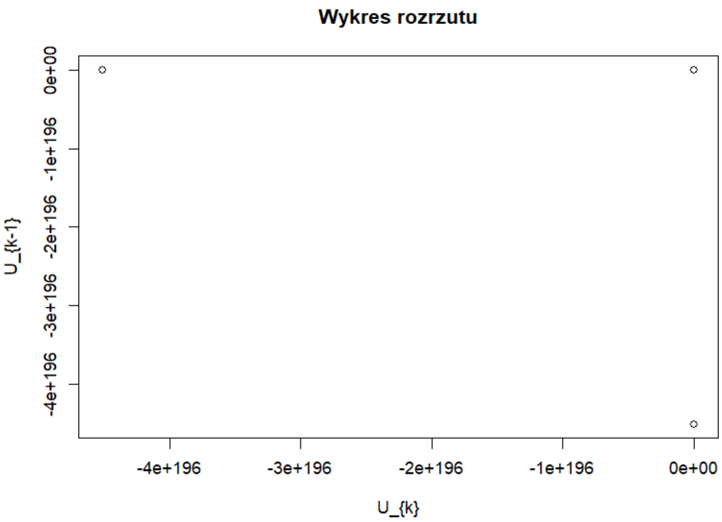
Rysunek 82: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
```



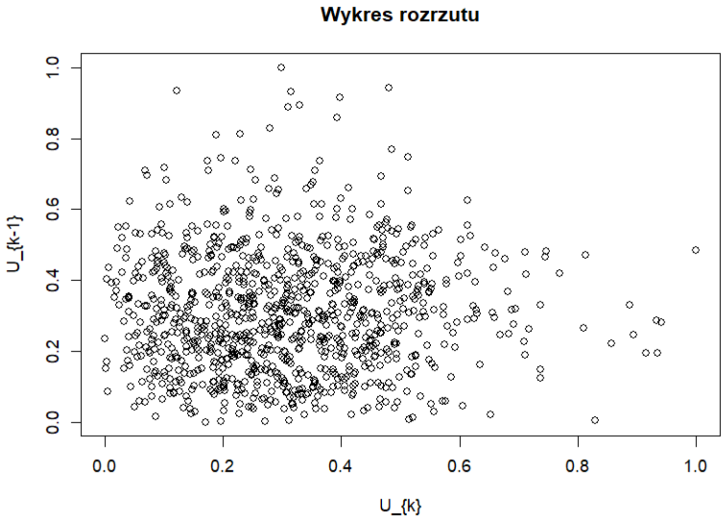
Rysunek 83: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu pięciokrotnego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)
```



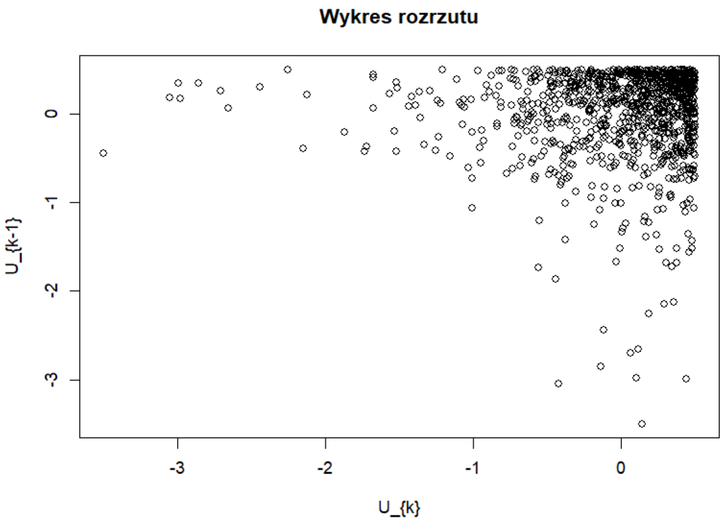
Rysunek 84: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu arcusa sinsua i zastosowaniu dwunastokrotnego złożenia mapy chaotycznej.

```
wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
```



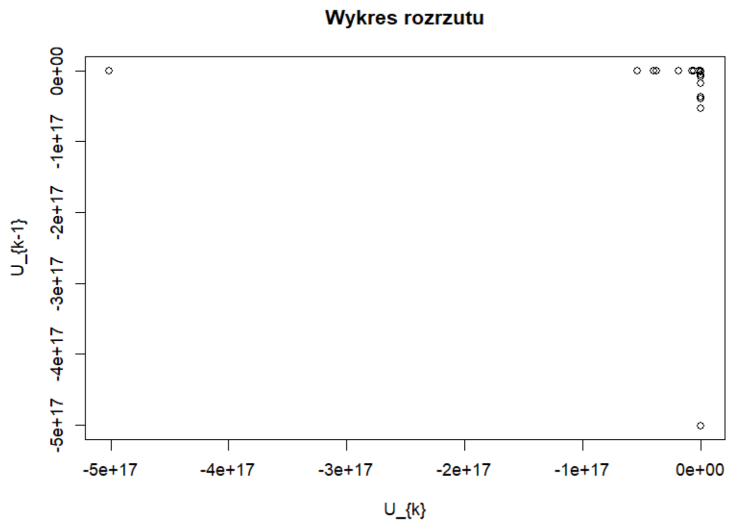
Rysunek 85: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha.

```
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
```



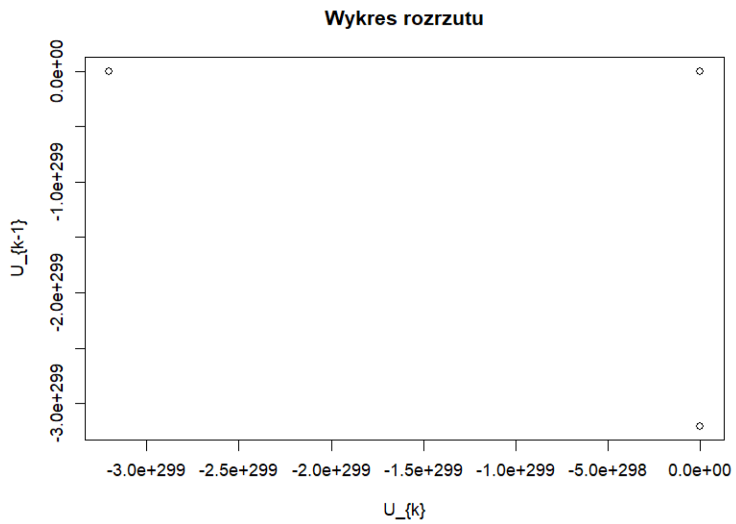
Rysunek 86: Wykres rozrzutu znormalizowanej próbki o rozmiarze 10^3 z rozkładu Rayleigha i zastosowaniu pojedynczego złożenia mapy chaotycznej.

```r
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.

```r
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:

```
img <- readImage("papugi_res.png")
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·M, co daje $\{R_w, G_w, B_w\}$

Wyodrębniamy trzy macierze R, G,B

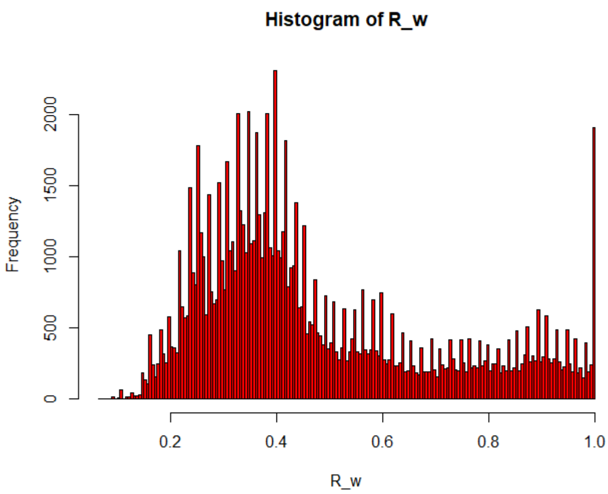
```
R <- img[,,1]
G <- img[,,2]
B <- img[,,3]

R_w <- c(R)
G_w <- c(G)
```

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

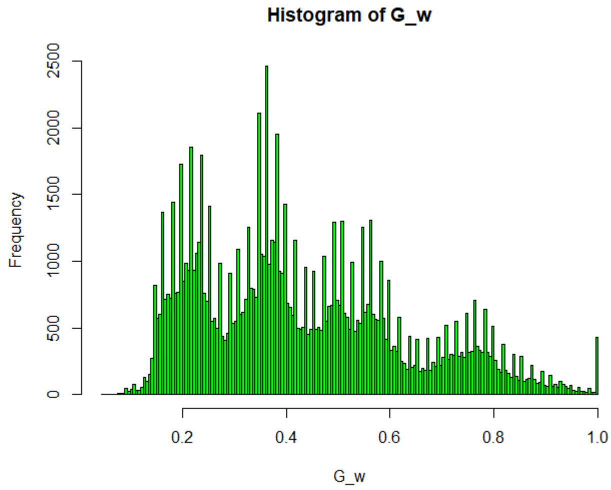
2.3 Zadanie 3

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

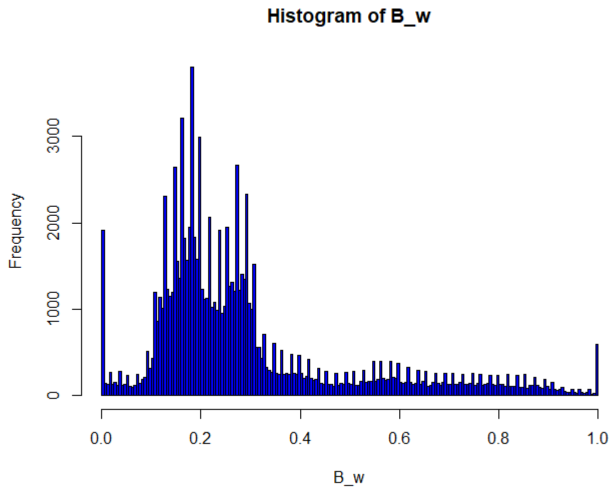
```
hist(R_w, col = 'red', breaks = 150)
hist(G_w, col = 'green', breaks = 150)
hist(B_w, col = 'blue', breaks = 150)
```



Rysunek 90: Histogram dla koloru R_w

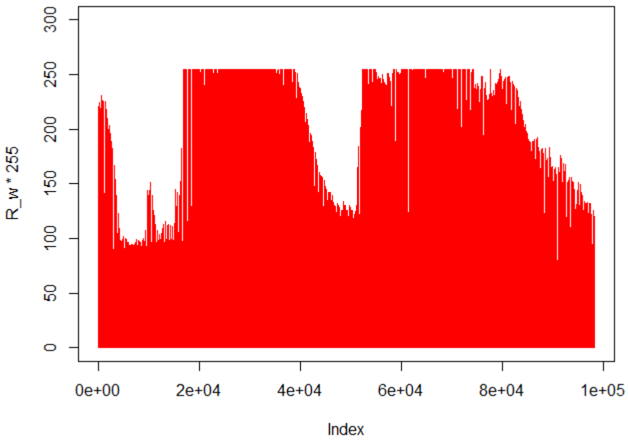


Rysunek 91: Histogram dla koloru G_w

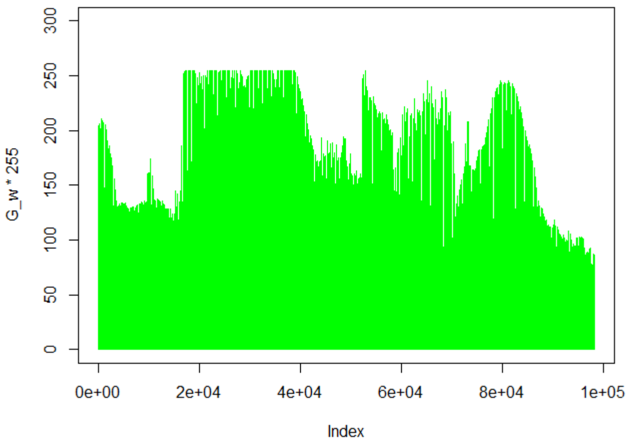


Rysunek 92: Histogram dla koloru B_w

```
plot(R_w*255, col = 'red', ylim = c(0,300), type = 'h')
plot(G_w*255, col = 'green', ylim = c(0,300), type = 'h')
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

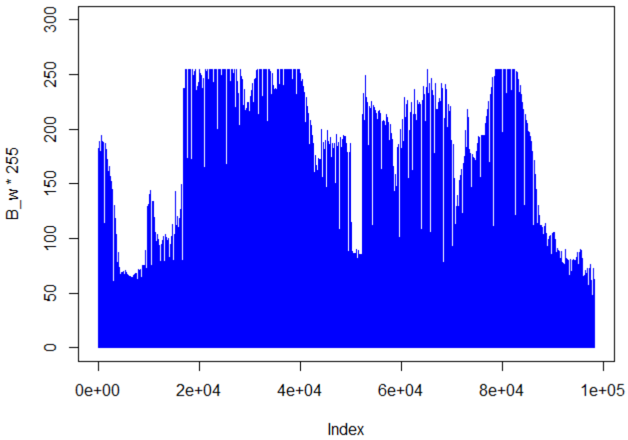
2.4 Zadanie 4

Wykorzystując mapę logistyczną

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

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

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



Rysunek 95: Wykres intensywności koloru B_w

```
  tmp[1] <- x0
for (i in 2:length(x)) {
    tmp[i] <- mapa_logistyczna(tmp[i-1], r)
  }
  return(tmp)}

x_R <- mapa_logistyczna_wektor(R_w, x0 = 0.5, r =3.8)
x_G <- mapa_logistyczna_wektor(G_w, x0 = 0.5, r =3.8)
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:

```
x_R_s <- sort(x_R)
x_G_s <- sort(x_G)
x_B_s <- sort(x_B)
```

Wektory pozycji:

```
P_R <- match(x_R_s, x_R)
P_G <- match(x_G_s, x_G)
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 itej pozycji wektora R_w, czyli $R_w[i]$, szukamy na jakiej pozycji stoi i w wektorze P_R (załóżmy, że naj), wówczas na pozycji j wektora T_R, czyli $T_R[j]$ wstawiamy $R_w[i]$.

Kodowanie:

```
kodowanie <- function(x, P){
  tmp <- numeric(length(x))
  for (i in 1:length(x)){
    x1 <- x[i]
    x2 <- P[i]
    tmp[x2] <- x1
  }
  return(tmp)}
```

Przygotowanie wektorów tasowanych:

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

2.7 Zadanie 7

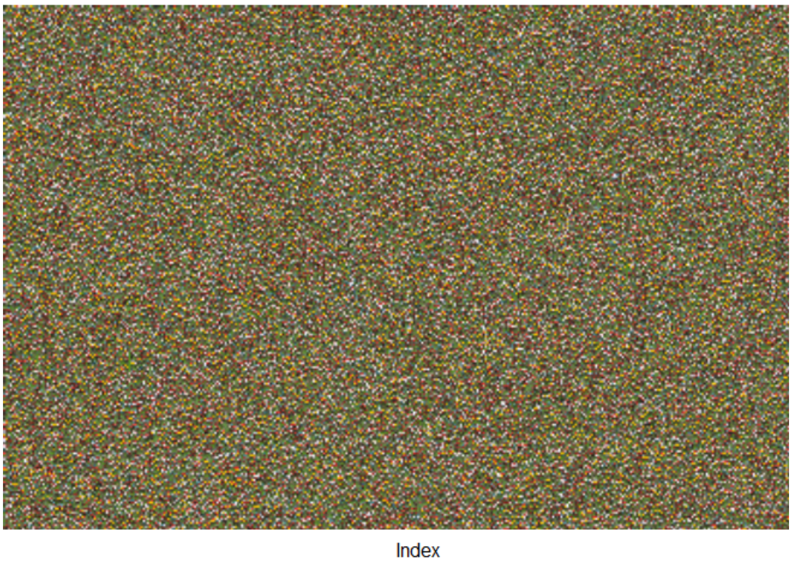
Łączymy wektory kolorów $\{T_R, T_G, T_B\}$ w tablicę (funkcja array()) o wymiarze N×M×3.

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

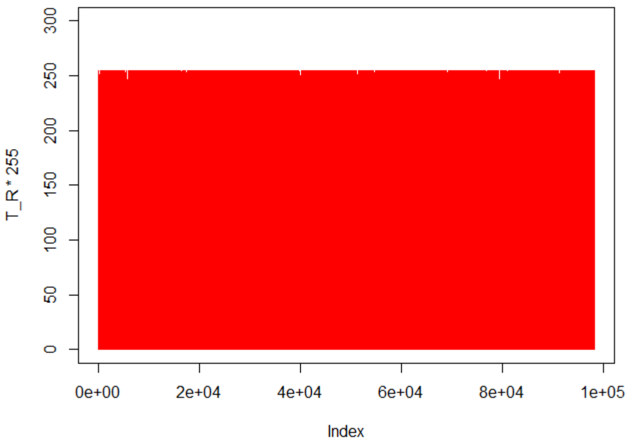
2.8 Zadanie 8

Wykreślamy intensywność każdego koloru zakodowanego obrazka

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



Rysunek 96: Zakodowany obraz papug

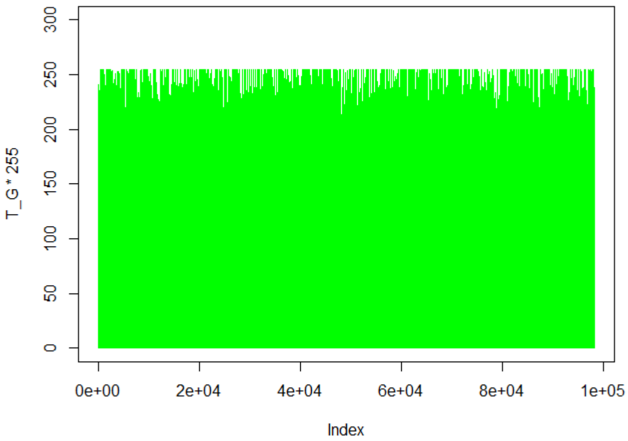


Rysunek 97: Wykres koloru R obrazu zakodowanego

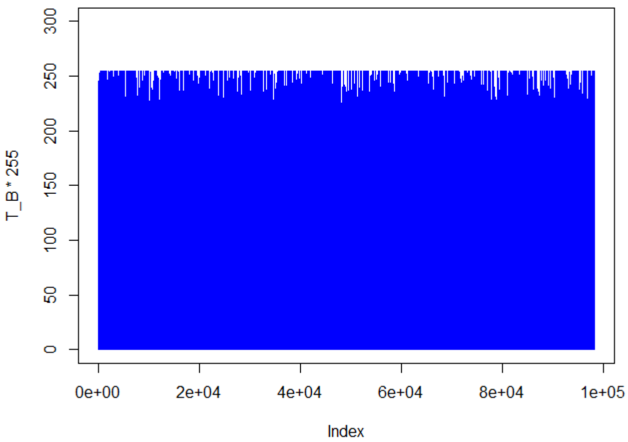
2.9 Zadanie 9

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

```
dekodowanie <- function(x, x0, r){
   x_mapowane <- mapa_logistyczna_wektor(x, x0, r)
   x_mapowane_sort <- sort(x_mapowane)
   x_mapowane_sort_pozycja <- match(x_mapowane_sort, x_mapowane)
  tmp <- numeric(length(x))
  for(i in (1:length(x))){
    pozycja <- x_mapowane_sort_pozycja[i]
    wartosc <- x[pozycja]
    tmp[i] <- wartosc
  }
  return(tmp)}

D_R <- dekodowanie(T_R, 0.5, 3.8)
D_G <- dekodowanie(T_G, 0.5, 3.8)
D_B <- dekodowanie(T_B, 0.5, 3.8)
```

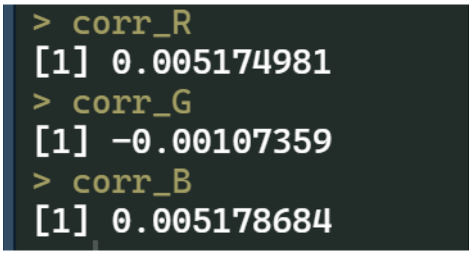
```
D <- array(c(D_R, D_G, D_B), dim=dim(img))
```

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

```
wspolczynnik_korelacj <- function(C_w, T_C) {

  licznik <- sum((C_w - mean(C_w))*(T_C - mean(T_C)))
  mianownik <- sqrt(sum(((C_w - mean(C_w)))^2))*sqrt(sum(((T_C - mean(T_C)))^2))
  return(licznik/mianownik)
}

corr_R <- wspolczynnik_korelacj(R_w, T_R)
corr_G <- wspolczynnik_korelacj(G_w, T_G)
corr_B <- wspolczynnik_korelacj(B_w, T_B)
```



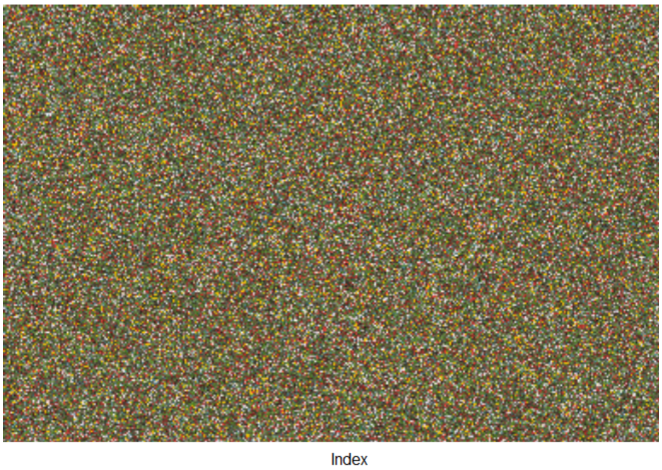
Rysunek 101: Wyniki korelacji dla kolorów R, G i B

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** $(x_0 = 0.1, r = 3.800000001)$

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

```
D_R_zmienione <- dekodowanie(T_R, x0 = 0.49, r = 3.79)
D_G_zmienione <- dekodowanie(T_G, x0 = 0.49, r = 3.79)
D_B_zmienione <- dekodowanie(T_B, x0 = 0.49, r = 3.79)

D_zmienione <- array(c(D_R_zmienione, D_G_zmienione, D_B_zmienione
    ), dim=dim(img))

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

```
# Projekt 1
# Aleksandra Burczyk
# Monika Gmaj
# Kinga Jankowska
# Kamil Langowski


# Cz. I ------------------------------------------------------------------




# Zadanie 1. ------------------------------------------------------------


# Ciagi liczb pseudolosowych generowane metoda dystrybuanty odwrotnej dla
# zadanej gestosci (rozklad jednostajny U(0,1))


pacman::p_load(moments, pracma)
# Rozklad jednostajny
set.seed(1)
sim_unif <- runif(10^3,0,1)



# 1.1 wyznaczyc w sposob analityczny dystrybuante i dystrybuante odwrotna

################ ARCUS SINUS ####################

# rozklad Arcus Sinus F^(x) : U(0,1) -> R
Arcsin <- function(x){ 2/pi*asin(sqrt(x))}


# rozklad Arcus Sinus F^-1(x) : U(0,1) -> R
inv_Arcsin <- function(x){ 1/2 - 1/2*cos(x*pi) }

Dane_Arcsin1 <- inv_Arcsin(runif(10^2,0,1))
Dane_Arcsin2 <- inv_Arcsin(runif(10^3,0,1))
Dane_Arcsin3 <- inv_Arcsin(runif(10^5,0,1))


# 1.2 przedstawienie dystrybuanty na wykresie
dystrybuanty <- function(x1,x2,d1,d2, ylimits) {
  plot(x1,d1(x1),type = "l",ylim = ylimits,col="green",ylab = "", lwd=2,
       main = "Funkcja gestosci i gestosci odwrotnej")
  lines(x2,d2(x2), lwd=2)
  legend("topleft",inset=.02, legend=c("F(x)", "F^-1(x)"),
         col=c("green", "black"),lty=3, cex=0.75,
         box.lty=1)
```

```
}

par(mfrow=c(1,1))
n = 1000
dystrybuanty(seq(0,1,length=n), seq(0,1,length=n), Arcsin, inv_Arcsin, c(0,1))

# 1.3 Miary statystyczne
Miary <- function(Dane, t_params){
  Miary_Statystyczne <- c("Mediana", "Srednia", "Dominanta", "Kurtoza",
                           "Wariancja", "Odchylenie standardowe", "Rozstep")
  Wyniki_Z_Generatora <- c(median(Dane), mean(Dane),density(Dane)$x[which.max(density(Dane)$y)],
                            kurtosis(Dane), var(Dane), std(Dane), max(Dane)-min(Dane))
  Wyniki_Teoretyczne <- t_params
  tabela <- data.frame(Miary_Statystyczne,Wyniki_Z_Generatora,Wyniki_Teoretyczne)
  print(summary(Dane))
  print(tabela)
}


Miary(Dane_Arcsin1, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
Miary(Dane_Arcsin2, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))
Miary(Dane_Arcsin3, c(1/2, 1/2, '{0,1}', 3/2, 1/8, sqrt(1/8), 1))



# 1.4 Histogram i gestosc
hist_PDF <- function(dane){
  par(mfrow=c(1,2))
  hist(dane, prob=TRUE, breaks=40,xlab = "x", main='Histogram', ylab = "Czestotliwosc")
  plot(density(dane), type = "l", pch = 19,
       col = "red", xlab = "x", ylab = "Gestosc", main='Gesto Ż ')
}

hist_PDF(Dane_Arcsin1)
hist_PDF(Dane_Arcsin2)
hist_PDF(Dane_Arcsin3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna
dyst_emp_teor <- function(d, d_inv, n, xlims, umin=0, umax=1, tmin=0, tmax=1) {
  plot(ecdf(d_inv(runif(n,umin,umax))),do.points=FALSE,
       main=paste("Dystrybuanta empiryczna i zgodna z rozkladem"),
       ylab = "", xlim = xlims)

  x <- seq(tmin,tmax,0.01)
  lines(x,d(x),type = "l",col="red")
  legend("bottomright",inset=.02, legend=c("F(x)-Dyst. emp", "F(x)-Dyst rozkl"),
         col=c("black", "red"),lty=3, cex=0.75,
         box.lty=1)
```

```
}

par(mfrow=c(1,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^2, xlims = c(0,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^3, xlims = c(0,1))
dyst_emp_teor(Arcsin,inv_Arcsin, n=10^5, xlims = c(0,1))

################# CAUCHY #####################

# rozklad Cauchyego F^(x) : U(0,1) -> R

Cauchy <- function(x,x0 = 0 ,alfa = 1) 1/2 + (1/pi)*atan((x-x0)/
    alfa)

# rozklad Cauchyego F^-1(x) : U(0,1) -> R

inv_Cauchy <- function(x,x0 = 0,alfa = 1/2){ x0 - alfa*cot(pi*x)}


Dane_inv_Cauchy <- inv_Cauchy(sim_unif)
Dane_inv_Cauchy <- Dane_inv_Cauchy[Dane_inv_Cauchy > -10 & Dane_
    inv_Cauchy < 10] # usunac mocno odstajace obs

Dane_Cauchy1 <- inv_Cauchy(runif(10^2,0,1))
Dane_Cauchy2 <- inv_Cauchy(runif(10^3,0,1))
Dane_Cauchy3 <- inv_Cauchy(runif(10^5,0,1))



# 1.2 przedstawienie dystrybuanty na wykresie
par(mfrow = c(1,1))
dystrybuanty(seq(-5,5,length=n),seq(0,1,length=n), Cauchy, inv_
    Cauchy, c(-3,3))

# 1.3 Miary statystyczne


Miary(Dane_Cauchy1, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infty'))
Miary(Dane_Cauchy2, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infty'))
Miary(Dane_Cauchy3, c(0,0,0,'nieoznaczony','nieoznaczony','
    nieoznaczony','infty'))


# 1.4 Histogram i gestosc

hist_PDF(Dane_Cauchy1)
hist_PDF(Dane_Cauchy2)
hist_PDF(Dane_Cauchy3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Cauchy,inv_Cauchy, n=1000, xlims= c(-10,10), umin
    =-10, umax=10, tmin=-10, tmax=10)


```

```
################# WARTOSCI EKSTREMALNYCH #####################

# rozklad wartosci ekstremalnych F^(x) : U(0,1) -> R
extr.val <- function(x,alfa=1,beta=1/2) 1-exp(-exp(x-alfa)/beta)

# rozklad wartosci ekstremalnych F^-1(x) : U(0,1) -> R
inv_extr.val <- function(x,alfa=1,beta=1/2) {(log(-alfa*log((1-x)))/beta)}

hist(inv_extr.val(sim_unif,alfa=1,beta=1/2), prob=TRUE, breaks=40)
lines(density(inv_extr.val(sim_unif,alfa=1,beta=1/2)))


Dane_extr.val1 <- inv_extr.val(runif(10^2,0,1))
Dane_extr.val2 <- inv_extr.val(runif(10^3,0,1))
Dane_extr.val3 <- inv_extr.val(runif(10^5,0,1))

# 1.2 przedstawienie dystrybuanty na wykresie
dystrybuanty(seq(-5,5,length=n),seq(0,1,length=n), extr.val, inv_extr.val, c(-2,3))

# 1.3 Miary statystyczne


Miary(Dane_extr.val1, c(0,1-1/2*-digamma(1),1,0,1/2^2*pi^2/6,0,0))
Miary(Dane_extr.val2, c(0,1-1/2*-digamma(1),1,0,1/2^2*pi^2/6,0,0))
Miary(Dane_extr.val3, c(0,1-1/2*-digamma(1),1,0,1/2^2*pi^2/6,0,0))

# 1.4 Histogram i gestosc
hist_PDF(Dane_extr.val1)
hist_PDF(Dane_extr.val2)
hist_PDF(Dane_extr.val3)

# 1.5 Dystrybuanta i Dystrybuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(extr.val,inv_extr.val, n=10000, xlims= c(-5,5), umin=-5, umax=5, tmin=-5, tmax=5)



################# Rayleigh #####################
# rozklad Reyleigha F^(x) : U(0,1) -> R
Rayleigh <- function(x,alfa = 0,beta=1)  1-exp(-((x-alfa)/beta)^2)


# rozklad Reyleigha F^-1(x) : U(0,1) -> R
inv_Rayleigh <- function(x,alfa = 0,beta=1)  alfa+beta*(sqrt(log(1/(1-x))))

hist(inv_Rayleigh(sim_unif), prob=TRUE, breaks=40)
lines(density(inv_Rayleigh(sim_unif)))

Dane_Rayleigh1 <- inv_Rayleigh(runif(10^2,0,1))
Dane_Rayleigh2 <- inv_Rayleigh(runif(10^3,0,1))
Dane_Rayleigh3 <- inv_Rayleigh(runif(10^5,0,1))

# 1.2 przedstawienie dystrybuanty na wykresie
```

```
dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Rayleigh, inv_Rayleigh, c(-2,3))

# 1.3 Miary statystyczne



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'))
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'))
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'))

# 1.4 Histogram i gestosc

hist_PDF(Dane_Rayleigh1)
hist_PDF(Dane_Rayleigh2)
hist_PDF(Dane_Rayleigh3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Rayleigh,inv_Rayleigh, n=1000, xlims= c(0,3), umin=0, umax=1 , tmax=3)


################# Weibull #####################

# rozklad Weibulla F^(x) : U(0,1) -> R
Weibull <- function(x,alfa=1,beta=2,c=1)  1-exp(-((x-alfa)/beta)^c) # x > alfa

# rozklad Weibulla F^-1(x) : U(0,1) -> R
inv_Weibull <- function(x,alfa=1,beta=2,c=1) alfa + beta*((-log(1-x))^(1/c))


Dane_Weibull1 <- inv_Weibull(runif(10^2,0,1))
Dane_Weibull2 <- inv_Weibull(runif(10^3,0,1))
Dane_Weibull3 <- inv_Weibull(runif(10^5,0,1))

# 1.2 przedstawienie dystrybuanty na wykresie
dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Weibull, inv_Weibull, c(-3,10))

# 1.3 Miary statystyczne


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'))
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'))
```

```
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'))

# 1.4 Histogram i gestosc


hist_PDF(Dane_Weibull1)
hist_PDF(Dane_Weibull2)
hist_PDF(Dane_Weibull3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Weibull,inv_Weibull, n=100, xlims= c(0,10), umin=0, umax=1, tmin=1, tmax=10)



################# LOGISTYCZNY #####################

# rozklad Logistyczny F^(x) : U(0,1) -> R
Logist <- function(x,alfa = 0 ,beta = 1) 1/(1+exp(-(x-alfa)/beta))

# rozklad Logistyczny F^-1(x) : U(0,1) -> R

inv_Logist <- function(x,alfa = 0,beta = 1) alfa + beta*log(x/(1-x))

hist(inv_Logist(sim_unif), prob=TRUE, breaks=40)
lines(density(inv_Logist(sim_unif)))

Dane_Logist1 <- inv_Logist(runif(10^2,0,1))
Dane_Logist2 <- inv_Logist(runif(10^3,0,1))
Dane_Logist3 <- inv_Logist(runif(10^5,0,1))

# 1.2 przedstawienie dystrybuanty na wykresie
dystrybuanty(seq(-5,5,length=n),seq(0,1,length=n), Logist, inv_Logist, c(-3,3))

# 1.3 Miary statystyczne



Miary(Dane_Logist1, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2),'infty'))
Miary(Dane_Logist2, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2),'infty'))
Miary(Dane_Logist3, c(0,0,0,pi^2*1/3,pi^2/3*1^2,sqrt(pi^2/3*1^2),'infty'))

# 1.4 Histogram i gestosc


hist_PDF(Dane_Logist1)
hist_PDF(Dane_Logist2)
hist_PDF(Dane_Logist3)
```

```
# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Logist,inv_Logist, n=100, xlims= c(-5,7), umin=0,
    umax=1, tmin=-5, tmax=7)


################# PARETO #####################

# rozklad Pareto F^(x) : U(0,1) -> R
Pareto <- function(x,c = 3)  1-x^(-c)

# rozklad Pareto F^-1(x) : U(0,1) -> R
inv_Pareto <- function(x,c = 3) (1-x)^(-1/c)

hist(inv_Pareto(sim_unif), prob=TRUE, breaks=40)
lines(density(inv_Pareto(sim_unif)))

Dane_Pareto1 <- inv_Pareto(runif(10^2,0,1))
Dane_Pareto2 <- inv_Pareto(runif(10^3,0,1))
Dane_Pareto3 <- inv_Pareto(runif(10^5,0,1))

# 1.2 przedstawienie dystrybuanty na wykresie
par(mfrow=c(1,1))
dystrybuanty(seq(0,1,length=n),seq(0,1,length=n),Pareto, inv_
    Pareto, c(-3,3))

# 1.3 Miary statystyczne


Miary(Dane_Pareto1, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infty'))
Miary(Dane_Pareto2, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infty'))
Miary(Dane_Pareto3, c(2^(1/3),3/2,1,0,3-(3/2)^2,sqrt(3-(3/2)^2),'
    infty'))

# 1.4 Histogram i gestosc

hist_PDF(Dane_Pareto1)
hist_PDF(Dane_Pareto2)
hist_PDF(Dane_Pareto3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Pareto,inv_Pareto, n=1000, xlims= c(0,50), umin=0,
    umax=1, tmin=-5, tmax=50)



################ TROJKATNY ####################

# rozklad Pareto F^(x) : U(0,1) -> R

Triangular_base <- function(x,a = 0,c = 1/2, b= 1) {
  if (x < a) value <- 0
  else if (x <= c) value <-(x-a)^2/((b-a)*(c-a))
```

```
  else if (x < b) value <- 1 - (b-x)^2/((b-a)*(b-c))
  else if (b <= x) value <- 1

  return(value)
}


Triangular <- function(x, ...) {
  sapply(x, Triangular_base, ...)
}


# rozklad Trojkatny F^-1(x) : U(0,1) -> R

inv_Triangular_base <- function(x,a = 0,c = 1/2, b= 1) {
  if (x < (c - a) / (b - a)) value <- a + sqrt(max(0, (c - a) * (b
    - a) * x))
  else if (x >= (c - a) / (b - a)) value <- b - sqrt(max(0, (b - a
    ) * (b - c) * (1 - x)))
  return(value)
}
inv_Triangular <- function(x, ...) {
  sapply(x, inv_Triangular_base, ...)
}

hist(inv_Triangular(sim_unif), prob=TRUE, breaks=40)
lines(density(inv_Triangular(sim_unif)))

Dane_Triangular1 <- inv_Triangular(runif(10^2,0,1))
Dane_Triangular2 <- inv_Triangular(runif(10^3,0,1))
Dane_Triangular3 <- inv_Triangular(runif(10^5,0,1))
# 1.2 przedstawienie dystrybuanty na wykresie
dystrybuanty(seq(0,1,length=n),seq(0,1,length=n), Triangular, inv_
    Triangular, c(0,1))

# 1.3 Miary statystyczne


Miary(Dane_Triangular1, c(min(Dane_Triangular3)+sqrt((max(Dane_
    Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
    ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
    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)))
Miary(Dane_Triangular2, c(min(Dane_Triangular3)+sqrt((max(Dane_
    Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
    ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
    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)))
Miary(Dane_Triangular3, c(min(Dane_Triangular3)+sqrt((max(Dane_
    Triangular3)-min(Dane_Triangular3))*(1/2 - min(Dane_Triangular3
    ))/2),(min(Dane_Triangular3)+max(Dane_Triangular3)+1/2)/3,1/
    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)))

# 1.4 Histogram i gestosc

hist_PDF(Dane_Triangular1)
hist_PDF(Dane_Triangular2)
hist_PDF(Dane_Triangular3)

# 1.5 Dystrybuanta i Dystrubuanta empiryczna

par(mfrow=c(1,1))
dyst_emp_teor(Triangular,inv_Triangular, n=1000, xlims= c(0,1), umin=0, umax=1, tmin=0, tmax=1)



# Zadanie 2 -------------------------------------------------------------

# Dla wygenerowanych pr bek z Zadania 1 c i g w liczb pseudolosowych o wybranym rozkĆadzie
# 2.1 Znormalizowa pr bki

normalizacja <- function(x) (x - min(x))/(max(x) - min(x))

{
  Dane_Arcsin1_normalizowane <- normalizacja(Dane_Arcsin1)
  Dane_Arcsin2_normalizowane <- normalizacja(Dane_Arcsin2)
  Dane_Arcsin3_normalizowane <- normalizacja(Dane_Arcsin3)
}

{
  Dane_Rayleigh1_normalizowane <- normalizacja(Dane_Rayleigh1)
  Dane_Rayleigh2_normalizowane <- normalizacja(Dane_Rayleigh2)
  Dane_Rayleigh3_normalizowane <- normalizacja(Dane_Rayleigh3)
}




# 2.2
# Dla ka dej znormalizowanej pr bki wygenerowa c i g liczb pseudolosowych U wykorzystuj c (2),
# gdzie fN oznacza N-t iteracjŹ mapy chaotycznej (1) dla p = 0.45.

# definiujemy funkcje mapy chaotycznej z polecenia
ukosna_mapa_namiotu <- function(zk, p=0.45) {
  if(zk >= 0 & zk < p) {
    return(zk/p)
  }
  if(zk >= p & zk <= 1) {
    return((1-zk)/(1-p))
  }
  else{
```

```
    print('Nieprawid Ćowa warto Ż ')
  }
}


# definiujemy N-te zlozenie mapy chaotycznej z polecelnia
ukosna_mapa_namiotu_N_iteracji <- function(N, x0, p = 0.45){
  x = c()
  for (j in 1:length(x0)) {
    x[j] = ukosna_mapa_namiotu(x0[j], p)

    if(N > 1) {
      for(i in 2:N){
        x[j] = ukosna_mapa_namiotu(x[j], p)
      }
    }
  }
  return(x)
}



{
  Dane_Arcsin1_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Arcsin3_normalizowane)
}


{
  Dane_Rayleigh1_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_1 <- ukosna_mapa_namiotu_N_iteracji(1, Dane_Rayleigh3_normalizowane)
}
```

```
{
  Dane_Rayleigh1_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_5 <- ukosna_mapa_namiotu_N_iteracji(5, Dane_Rayleigh3_normalizowane)
}


{
  Dane_Rayleigh1_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_12 <- ukosna_mapa_namiotu_N_iteracji(12, Dane_Rayleigh3_normalizowane)
}

# 2.3 Wygenerowa  histogramy rozkĆadu U dla ka dej z pr bek, dla trzech wartoŻci N = {1, 5, 12}

hist(Dane_Arcsin1_prawdziwie_losowe_1)
hist(Dane_Arcsin2_prawdziwie_losowe_1)
hist(Dane_Arcsin3_prawdziwie_losowe_1)

hist(Dane_Arcsin1_prawdziwie_losowe_5)
hist(Dane_Arcsin2_prawdziwie_losowe_5)
hist(Dane_Arcsin3_prawdziwie_losowe_5)

hist(Dane_Arcsin1_prawdziwie_losowe_12)
hist(Dane_Arcsin2_prawdziwie_losowe_12)
hist(Dane_Arcsin3_prawdziwie_losowe_12)




hist(Dane_Rayleigh1_prawdziwie_losowe_1)
hist(Dane_Rayleigh2_prawdziwie_losowe_1)
hist(Dane_Rayleigh3_prawdziwie_losowe_1)

hist(Dane_Rayleigh1_prawdziwie_losowe_5)
hist(Dane_Rayleigh2_prawdziwie_losowe_5)
hist(Dane_Rayleigh3_prawdziwie_losowe_5)

hist(Dane_Rayleigh1_prawdziwie_losowe_12)
hist(Dane_Rayleigh2_prawdziwie_losowe_12)
hist(Dane_Rayleigh3_prawdziwie_losowe_12)


# 2.4 W ka dym przypadku wygenerowa  wykresy zale noŻci par (uk,uk+1)
# (rozrzut punkt w na pĆaszczy nie (uk,uk+1)).

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

```
                                         ylab = 'U_{k-1}',
                                         main = 'Wykres rozrzutu')

wykres_rozrzutu(Dane_Arcsin2_normalizowane)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12)



wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12)





# Zadanie 3
    ---------------------------------------------------------------
# Dla wygenerowanych pr bek z Zadania 1 c i g w liczb
    pseudolosowych o wybranym rozk Ćadzie
# w y k o n a  podpunkty 2.1-2.4 dla d w c h dowolnie wybranych z [1]
    map chaotycznych.

#
# definiujemy funkcje mapy logistycznej
mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))


# definiujemy N-te zlozenie mapy chaotycznej
mapa_logistyczna_N_iteracji <- function(N, x0, r=3.8){
  x = c()
  for (j in 1:length(x0)) {

    x[j] = mapa_logistyczna(x0[j], r)
    if(N > 1){
      for(i in 2:N){
        x[j] = mapa_logistyczna(x[j], r)
      }
    }
  }
  return(x)
}


# definiujemy funkcje mapy kwadratowej
mapa_kwadratowa <- function(xk, a=4, b=0.5) return(b - a*xk^2)


# definiujemy N-te zlozenie mapy chaotycznej
mapa_kwadratowa_N_iteracji <- function(N, x0, a=4, b=0.5){
  x = c()
  for (j in 1:length(x0)) {

    x[j] = mapa_kwadratowa(x0[j], a, b)
    if(N>1){
```

```
      for(i in 2:N){
        x[j] = mapa_kwadratowa(x[j], a, b)
      }
    }
  }
  return(x)
}

################# MAPA LOGISTYCZNA #####################


# 3.2
# Dla ka dej znormalizowanej pr bki wygenerowa  ci g liczb pseudolosowych U wykorzystuj c (2),
# gdzie fN oznacza N-t  iteracjŹ mapy chaotycznej (1) dla p = 0.45.

{
  Dane_Arcsin1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Arcsin3_normalizowane)
}


{
  Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna <- mapa_logistyczna_N_iteracji(1, Dane_Rayleigh3_normalizowane)
}

{
  Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh1_normalizowane)
```

```
  Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna <- mapa_logistyczna_N_iteracji(5, Dane_Rayleigh3_normalizowane)
}

{
  Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna <- mapa_logistyczna_N_iteracji(12, Dane_Rayleigh3_normalizowane)
}

# 3.3 Wygenerowa histogramy rozkĆadu U dla ka dej z pr bek, dla trzech wartoŻci N = {1, 5, 12}

hist(Dane_Arcsin1_prawdziwie_losowe_1_logistyczna)
hist(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
hist(Dane_Arcsin3_prawdziwie_losowe_1_logistyczna)

hist(Dane_Arcsin1_prawdziwie_losowe_5_logistyczna)
hist(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
hist(Dane_Arcsin3_prawdziwie_losowe_5_logistyczna)

hist(Dane_Arcsin1_prawdziwie_losowe_12_logistyczna)
hist(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
hist(Dane_Arcsin3_prawdziwie_losowe_12_logistyczna)

hist(Dane_Rayleigh1_prawdziwie_losowe_1_logistyczna)
hist(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
hist(Dane_Rayleigh3_prawdziwie_losowe_1_logistyczna)

hist(Dane_Rayleigh1_prawdziwie_losowe_5_logistyczna)
hist(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
hist(Dane_Rayleigh3_prawdziwie_losowe_5_logistyczna)

hist(Dane_Rayleigh1_prawdziwie_losowe_12_logistyczna)
hist(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)
hist(Dane_Rayleigh3_prawdziwie_losowe_12_logistyczna)

# 3.4 W ka dym przypadku wygenerowa wykresy zale noŻci par (uk,uk+1)
# (rozrzut punkt w na pĆaszczy nie (uk,uk+1)).

wykres_rozrzutu(Dane_Arcsin2_normalizowane)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_logistyczna)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_logistyczna)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_logistyczna)
```

```
wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_logistyczna)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_logistyczna)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_logistyczna)



################# MAPA KWADRATOWA #####################

# 3.2
# Dla ka dej znormalizowanej pr bki wygenerowa  ci g liczb pseudolosowych U wykorzystuj c (2),
# gdzie fN oznacza N-t  iteracj Ź mapy chaotycznej (1) dla p = 0.45.

{
  Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Arcsin3_normalizowane)
}

{
  Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin1_normalizowane)
  Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin2_normalizowane)
  Dane_Arcsin3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Arcsin3_normalizowane)
}


{
  Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa <- mapa_kwadratowa_N_iteracji(1, Dane_Rayleigh3_normalizowane)
}

{
  Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Rayleigh2_normalizowane)
```

```
  Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa <- mapa_kwadratowa_N_iteracji(5, Dane_Rayleigh3_normalizowane)
}

{
  Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Rayleigh1_normalizowane)
  Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Rayleigh2_normalizowane)
  Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa <- mapa_kwadratowa_N_iteracji(12, Dane_Rayleigh3_normalizowane)
}


# 3.3 Wygenerowa  histogramy rozkĆadu U dla ka dej z pr bek, dla trzech wartoŻci N = {1, 5, 12}

hist(Dane_Arcsin1_prawdziwie_losowe_1_kwadratowa)
hist(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
hist(Dane_Arcsin3_prawdziwie_losowe_1_kwadratowa)

hist(Dane_Arcsin1_prawdziwie_losowe_5_kwadratowa)
hist(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
hist(Dane_Arcsin3_prawdziwie_losowe_5_kwadratowa)

hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)
hist(Dane_Arcsin1_prawdziwie_losowe_12_kwadratowa)



hist(Dane_Rayleigh1_prawdziwie_losowe_1_kwadratowa)
hist(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
hist(Dane_Rayleigh3_prawdziwie_losowe_1_kwadratowa)

hist(Dane_Rayleigh1_prawdziwie_losowe_5_kwadratowa)
hist(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
hist(Dane_Rayleigh3_prawdziwie_losowe_5_kwadratowa)

hist(Dane_Rayleigh1_prawdziwie_losowe_12_kwadratowa)
hist(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)
hist(Dane_Rayleigh3_prawdziwie_losowe_12_kwadratowa)

# 3.4 W ka dym przypadku wygenerowa  wykresy zale noŻci par (uk,uk+1)
# (rozrzut punkt w na pĆaszczy nie (uk,uk+1)).


wykres_rozrzutu(Dane_Arcsin2_normalizowane)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_1_kwadratowa)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_5_kwadratowa)
wykres_rozrzutu(Dane_Arcsin2_prawdziwie_losowe_12_kwadratowa)



wykres_rozrzutu(Dane_Rayleigh2_normalizowane)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_1_kwadratowa)
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_5_kwadratowa)
```

```r
wykres_rozrzutu(Dane_Rayleigh2_prawdziwie_losowe_12_kwadratowa)

# Cz. II --------------------------------------------------------------

library(OpenImageR)

# 1. Zapisujemy i Ćadujemy obrazek
img <- readImage("papugi_res.png")
imageShow(img)
# 2. WyodrŹbniamy trzy macierze {R, G,B}

R <- img[,,1]
G <- img[,,2]
B <- img[,,3]
R_w <- c(R)
G_w <- c(G)
B_w <- c(B)

# 3. Dla ka dego koloru oryginalnego obrazka mo emy zobrazowa intensywnoŻ
# oraz przedstawi histogramy

hist(R_w, col = 'red', breaks = 150)
hist(G_w, col = 'green', breaks = 150)
hist(B_w, col = 'blue', breaks = 150)

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

# 4. Wykorzystuj c mapŹ logistyczn dla wybranego r  [3.6, 4] i x0  (0, 1)
# konstruujemy wektor wartoŻci o dĆugoŻci K = 3 N M .
# Otrzymany wektor dzielimy na trzy podwektory {xR, xG, xB}.
# Klucz kodowania skĆada siŹ z dw ch informacji (x0, r).

# przyjmujemy x0 = 0.5 i r = 3.8

x0 <- 0.5
r <- 3.8
mapa_logistyczna <- function(x0, r=3.8) return(r*x0*(1 - x0))
mapa_logistyczna_wektor <- function(x, x0, r) {

  tmp <- numeric(length(x))

  tmp[1] <- x0

  for (i in 2:length(x)) {

    tmp[i] <- mapa_logistyczna(tmp[i-1], r)

  }

  return(tmp)
}

```

```
x_R <- mapa_logistyczna_wektor(R_w, x0 = 0.5, r =3.8)
x_G <- mapa_logistyczna_wektor(G_w, x0 = 0.5, r =3.8)
x_B <- mapa_logistyczna_wektor(B_w, x0 = 0.5, r =3.8)


# 5. Sortujemy (malej co lub rosn co) osobno wektory {xR, xG, xB } otrzymuj c {xsR, xsG, xsB}.
# Nast Źpnie tworzymy trzy wektory pozycji {PR, PG, PB}, kt re przechowuj  numery pozycji
# element w z {xsR, xsG, xsB} w wektorach {xR, xG, xB}.




# sortowanie
x_R_s <- sort(x_R)
x_G_s <- sort(x_G)
x_B_s <- sort(x_B)




# wektory pozycji
P_R <- match(x_R_s, x_R)
P_G <- match(x_G_s, x_G)
P_B <- match(x_B_s, x_B)

# 6. Kodowanie
kodowanie <- function(x, P){

  tmp <- numeric(length(x))

  for (i in 1:length(x)){
    x1 <- x[i]
    x2 <- P[i]
    tmp[x2] <- x1
  }

  return(tmp)
}
# przygotowanie wektor w tasowanych

T_R <- kodowanie(R_w, P_R)
T_G <- kodowanie(G_w, P_G)
T_B <- kodowanie(B_w, P_B)

# 7.  Ączenie  wektor w

T <- array(data=c(T_R, T_G, T_B), dim = dim(img))

# wy Żwietlenie zakodowanego obrazka
imageShow(T)

# 8. Wykre Żlamy intensywno Ż  ka dego koloru zakodowanego obrazka

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

```
plot(T_B*255, col = 'blue', ylim = c(0,300), type = 'h')


# 9. Dekodowanie

# klucz: x0 = 0.5 i r = 3.8


dekodowanie <- function(x, x0, r){
  x_mapowane <- mapa_logistyczna_wektor(x, x0, r)
  x_mapowane_sort <- sort(x_mapowane)
  x_mapowane_sort_pozycja <- match(x_mapowane_sort, x_mapowane)
  tmp <- numeric(length(x))
  for(i in (1:length(x))){
    pozycja <- x_mapowane_sort_pozycja[i]
    wartosc <- x[pozycja]
    tmp[i] <- wartosc
  }
  return(tmp)
}


D_R <- dekodowanie(T_R, 0.5, 3.8)
D_G <- dekodowanie(T_G, 0.5, 3.8)
D_B <- dekodowanie(T_B, 0.5, 3.8)

D <- array(c(D_R, D_G, D_B), dim=dim(img))


imageShow(D)


# 10  Wyznaczamy wsp Ćczynnik korelacji mi Ździ poszczeg lnymi
    kolorami oryginalnego obrazu
# a kolorami zakodowanego obrazu


wspolczynnik_korelacj <- function(C_w, T_C) {

  licznik <- sum((C_w - mean(C_w))*(T_C - mean(T_C)))
  mianownik <- sqrt(sum(((C_w - mean(C_w)))^2))*sqrt(sum(((T_C -
    mean(T_C)))^2))
  return(licznik/mianownik)
}

corr_R <- wspolczynnik_korelacj(R_w, T_R)
corr_G <- wspolczynnik_korelacj(G_w, T_G)
corr_B <- wspolczynnik_korelacj(B_w, T_B)

# 12. Sprawdzamy wra liwo Ż  kodowania wzgl Źdem doboru klucza.
# przyjmujemy x0 = 0.49, r = 3.79

D_R_zmienione <- dekodowanie(T_R, x0 = 0.49, r = 3.79)
D_G_zmienione <- dekodowanie(T_G, x0 = 0.49, r = 3.79)
D_B_zmienione <- dekodowanie(T_B, x0 = 0.49, r = 3.79)

D_zmienione <- array(c(D_R_zmienione, D_G_zmienione, D_B_zmienione
    ), dim=dim(img))
```

```

imageShow(D_zmienione)
```