# Raport 1

Magdalena Wawrzyniak

2022-10-18

## Zadanie 1

W pierwszym zadaniu należało wygenerować 100 wektorów losowych z rozkładu dwuwymiarowego normalnego $N(0, I)$, a następnie zaznaczyć je na płaszczyźnie. W pierszej kolejności musimy wygrać bibliotekę `ggplot2` i ustawić ziarno, które będzie obowiązywać dla wszystkich symulacji w raporcie. Będzie to wstęp do zadania drugiego.

```
library(ggplot2)
library(plotly)
```

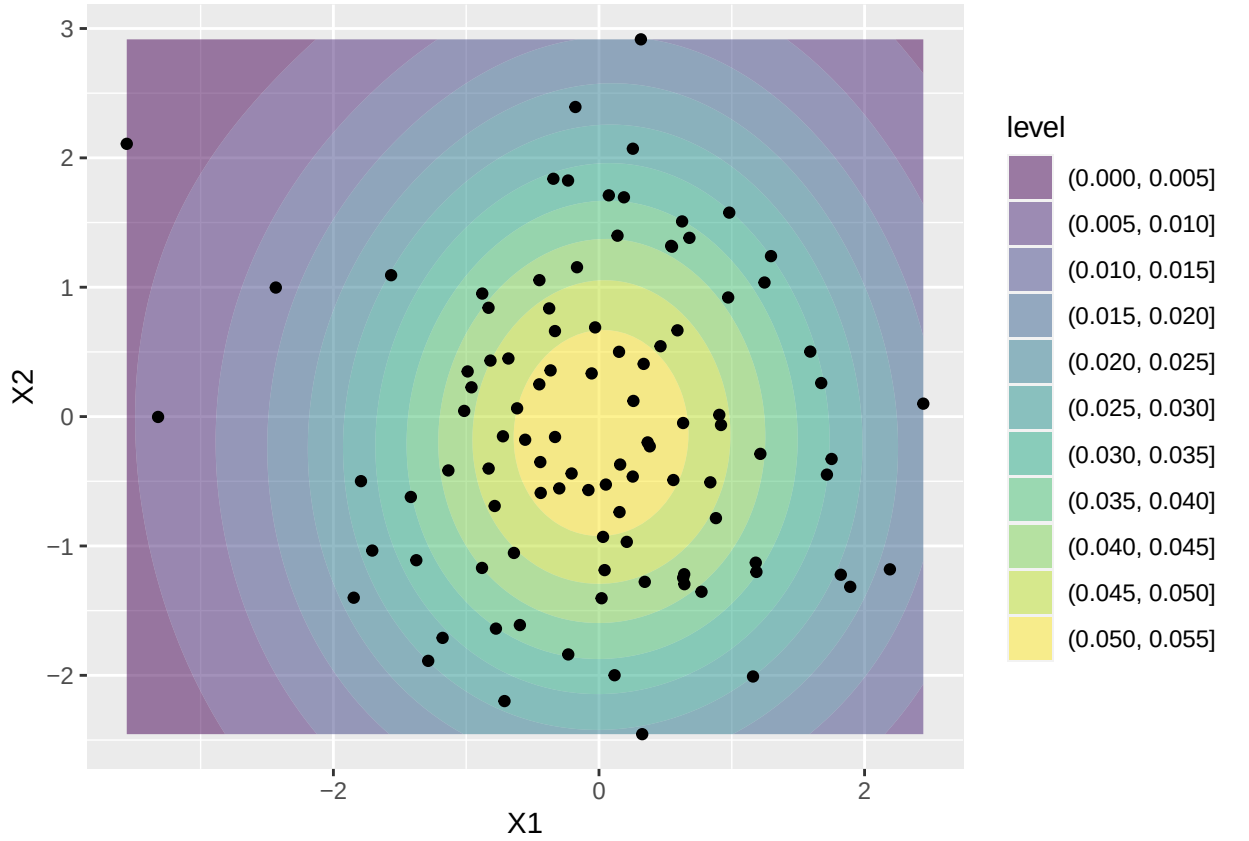
```
##
## Dołączanie pakietu: 'plotly'
```

```
## Następujący obiekt został zakryty z 'package:ggplot2':
##
##     last_plot
```

```
## Następujący obiekt został zakryty z 'package:stats':
##
##     filter
```

```
## Następujący obiekt został zakryty z 'package:graphics':
##
##     layout
```

```
set.seed(17)

num_vectors = 100
d = 2

M <- matrix(rnorm(d * num_vectors), nrow = num_vectors)
X = data.frame(X1 = M[,1], X2 = M[,2])

ggplot(X, (aes(x = X1, y = X2))) + stat_density_2d_filled(adjust = 3,
                                                          alpha = 0.5) +
                                   geom_point((aes(x = X1, y = X2)))
```




## Zadanie 2

W ty zadaniu wyznaczamy przekształcenia liniowe, które przekształca chmurę punktów z poprzedniego zadania w nową chmurę z podanego rozkładu $N(\mu, \Sigma)$. U mnie w programie $\Sigma$ oznacane jest przez E1-E3, $\mu = (4, 2)$ dla każdego przykładu. Przed omawianiem możliwych rozwiązań wgrywamy dane.

```
#Dane

mi_Y = c(4, 2)

E1 = matrix(c(1, 0.9, 0.9, 1), nrow = 2)
E2 = matrix(c(1, -0.9, -0.9, 1), nrow = 2)
E3 = matrix(c(9, 0, 0, 1), nrow = 2)
```

Zadane możemy rozwiązać czterema metodami:\

**Metoda 1: Wyliczyć przekształcenie bezpośrednio ze wzoru $\Sigma^Y = AA^T$.**

Pomijamy te metodę.

**Metoda 2: Diagonalizacja**

Do przeprowadzenia diagonalizacji napisałam funkcję, która za argument przyjmuje macierz $\Sigma$. Wykorzystujemy tam funkcje `eigen`, która zwraca wartości i wektory własne. Są nam one potrzebne, ponieważ macierz

$A$ jest iloczynem macierzy złożonej z wektorów własnych i pierwiastka macierzy diagonalnej.

```
diag = function(E){

  H = eigen(E)
  d = H$values

  D = matrix(c(d[1], 0, 0, d[2]), nrow = 2)
  P = H$vectors

  S = sqrt(D)

  A = P %*% S

return(A)}
```

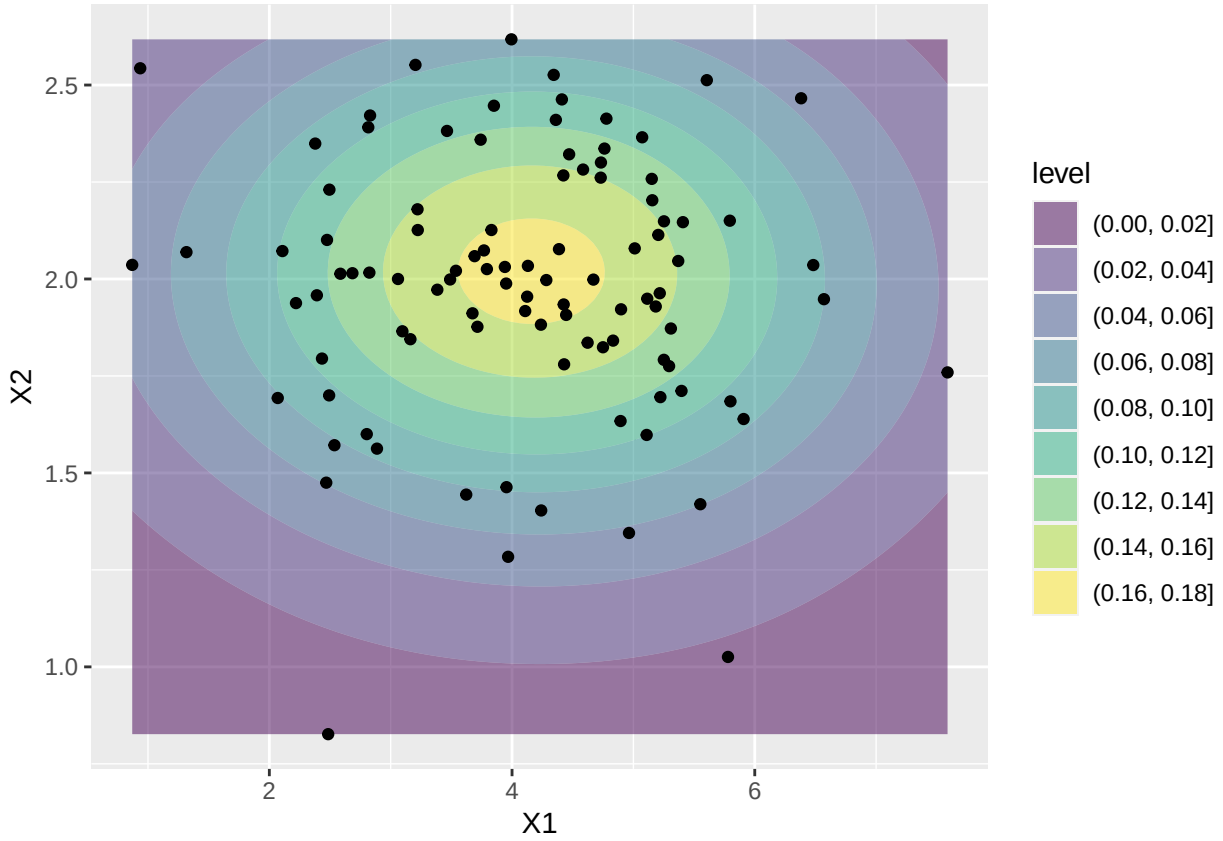
Po zastoowaniu funkcji dostajemy macierze odpowiednie macierze A:

```
A1d = diag(E1)
A2d = diag(E2)
A3d = diag(E3)

B = mi_Y
```

Chmura punktów dla metody 2 i 4 będziegenerowana przez poniższą funkcje. Przyjmuje ona cztery argumenty, kolejno: macierz $A$, wektor $B$, wymiar $d$ oraz ilość wektorów $n$.

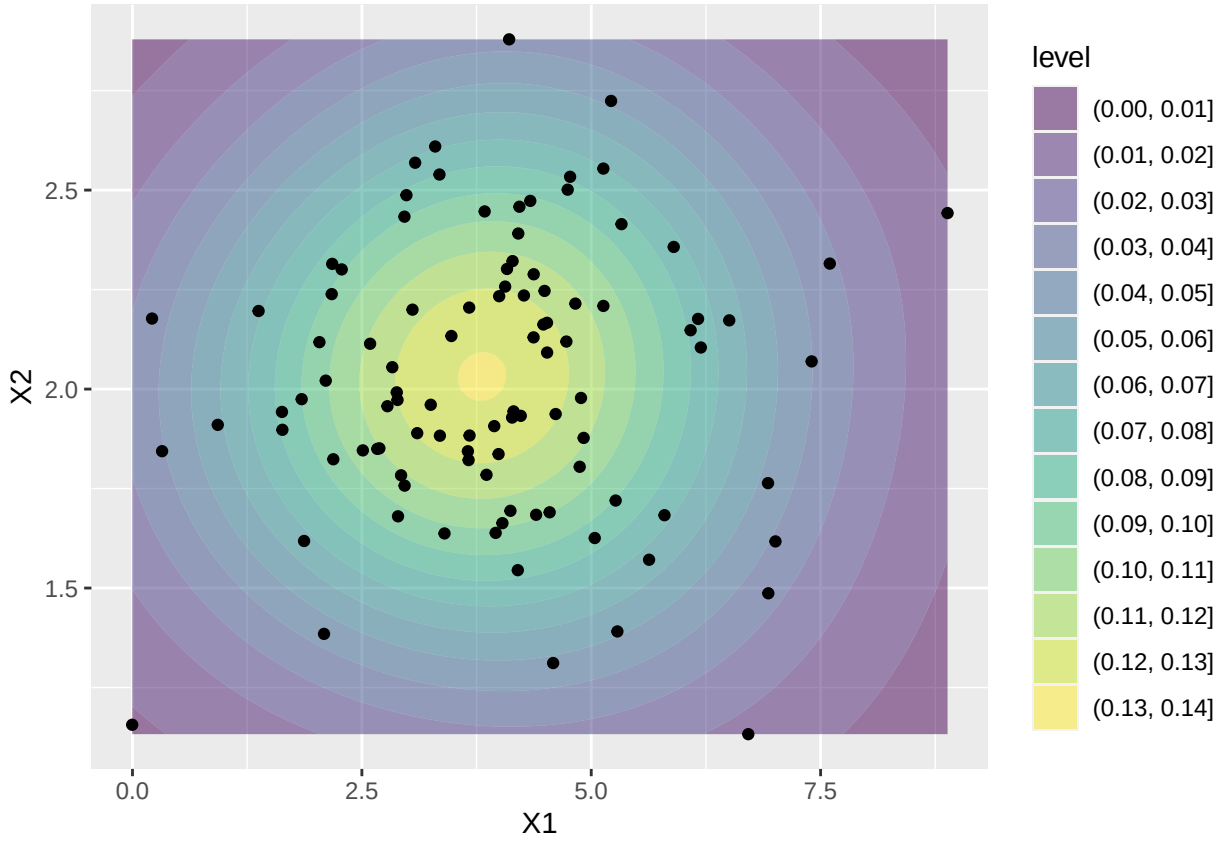
```
cloud = function(A, B, d, n){

  x1 = numeric(n)
  x2 = numeric(n)

  for (i in 1:n){
    z0 <- rnorm(d)
    z1 <- t(A) %*% z0 + B
    x1[i] <- z1[1]
    x2[i] <- z1[2]
  }
X = data.frame(X1 = x1, X2 = x2)

return(ggplot(X, aes(x = X1, y = X2)) + stat_density_2d_filled(adjust = 3,
                                                               alpha = 0.5)
                                      + geom_point()
                                      + xlab('X1')
                                      + ylab('X2'))
}
```

```
cloud(A1d, B, 2, 100)
```



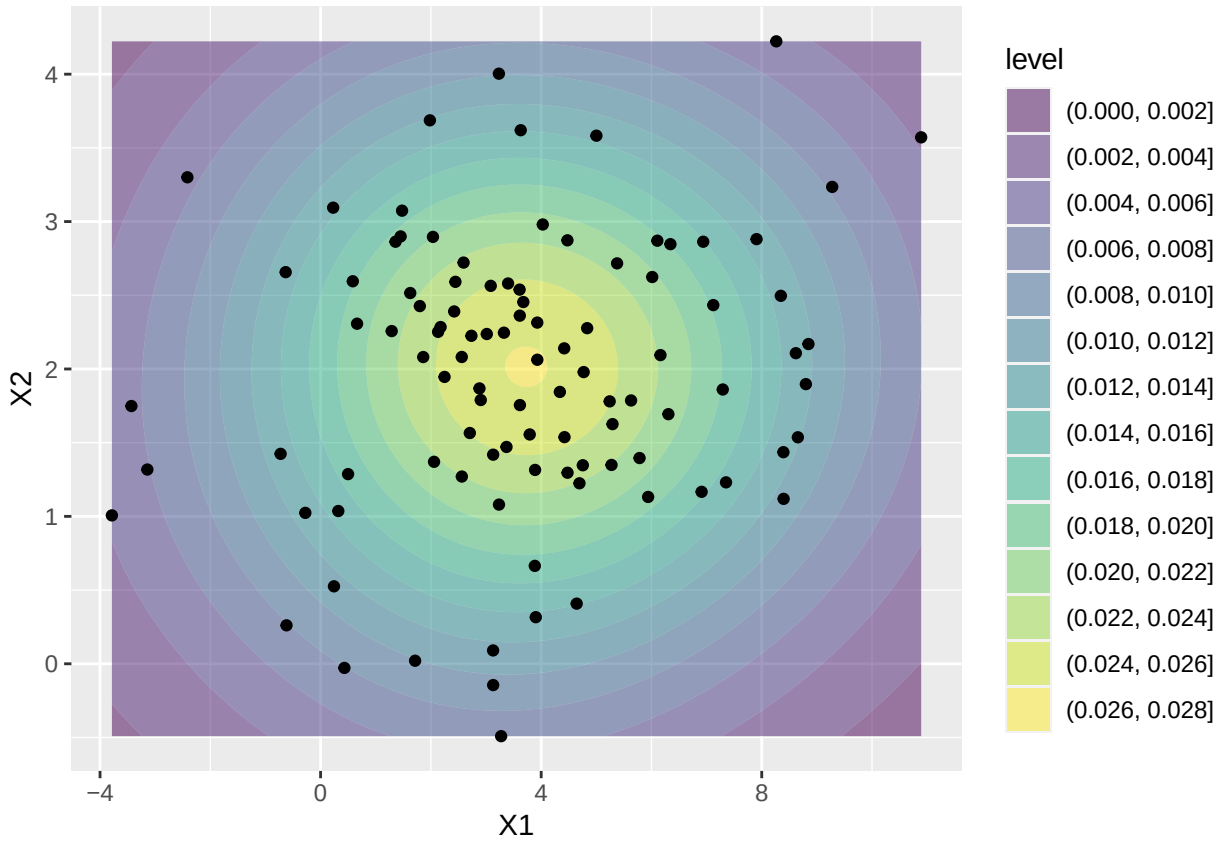

**Przykład 1:**

```
cloud(A2d, B, 2, 100)
```




**Przykład 2:**

```
cloud(A3d, B, 2, 100)
```



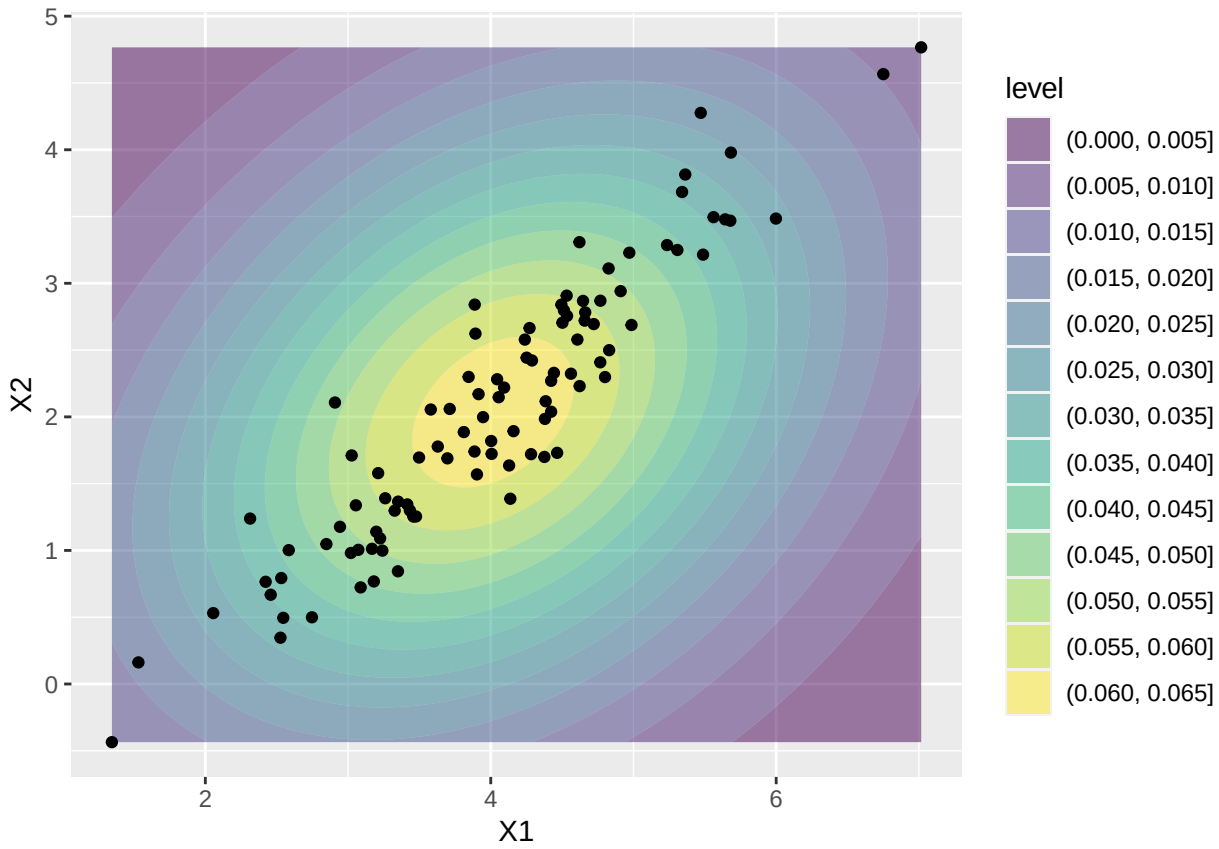

**Przykład 3:**

**Metoda 3: Wykorzystanie funkcji z pakietu MASS**

W tej metodzie wykorzystujemy wbudowaną funkcję `mvrnorm` pochodzącą z pakietu `MASS`, argumenty jakie przyjmuje to $n$, $\mu$ oraz $\Sigma$, a zwraca nam gotowe przekształcenie, które od razu możemy zaznaczyć jako chmurę punktów. Stworzyłam funkcję o nazwie `Xmass`, która zwraca mi wygenerowaną chmurę punktów.

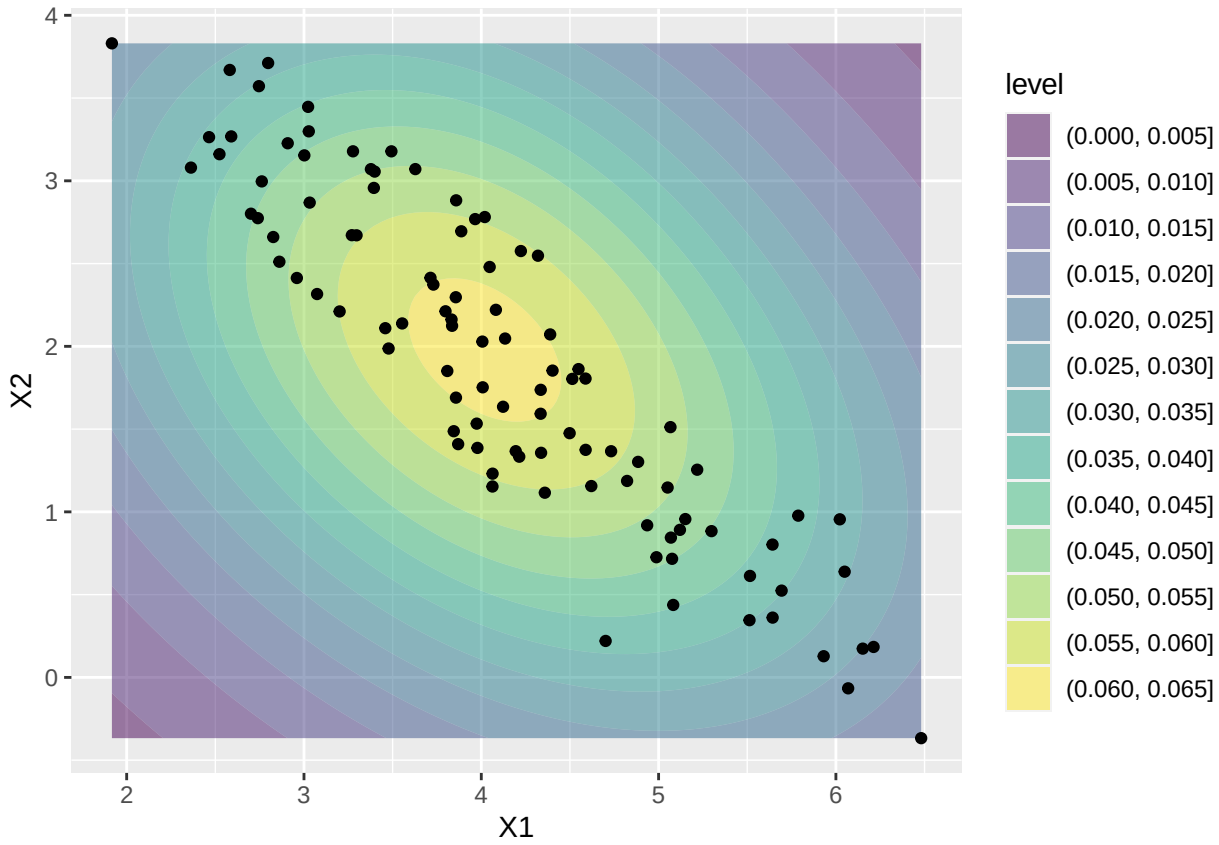
```
Xmass = function(n, mu, Sigma){

  X = MASS::mvrnorm(n, mu, Sigma)
  Y = data.frame(X1 = X[, 1],
                 X2 = X[, 2])

  return(ggplot(Y, aes(x = X1, y = X2)) + stat_density_2d_filled(adjust = 3,
                                                                 alpha = 0.5)
                                        + geom_point()
                                        + xlab('X1')
                                        + ylab('X2'))
}
```

```
Xmass(num_vectors, B, E1)
```



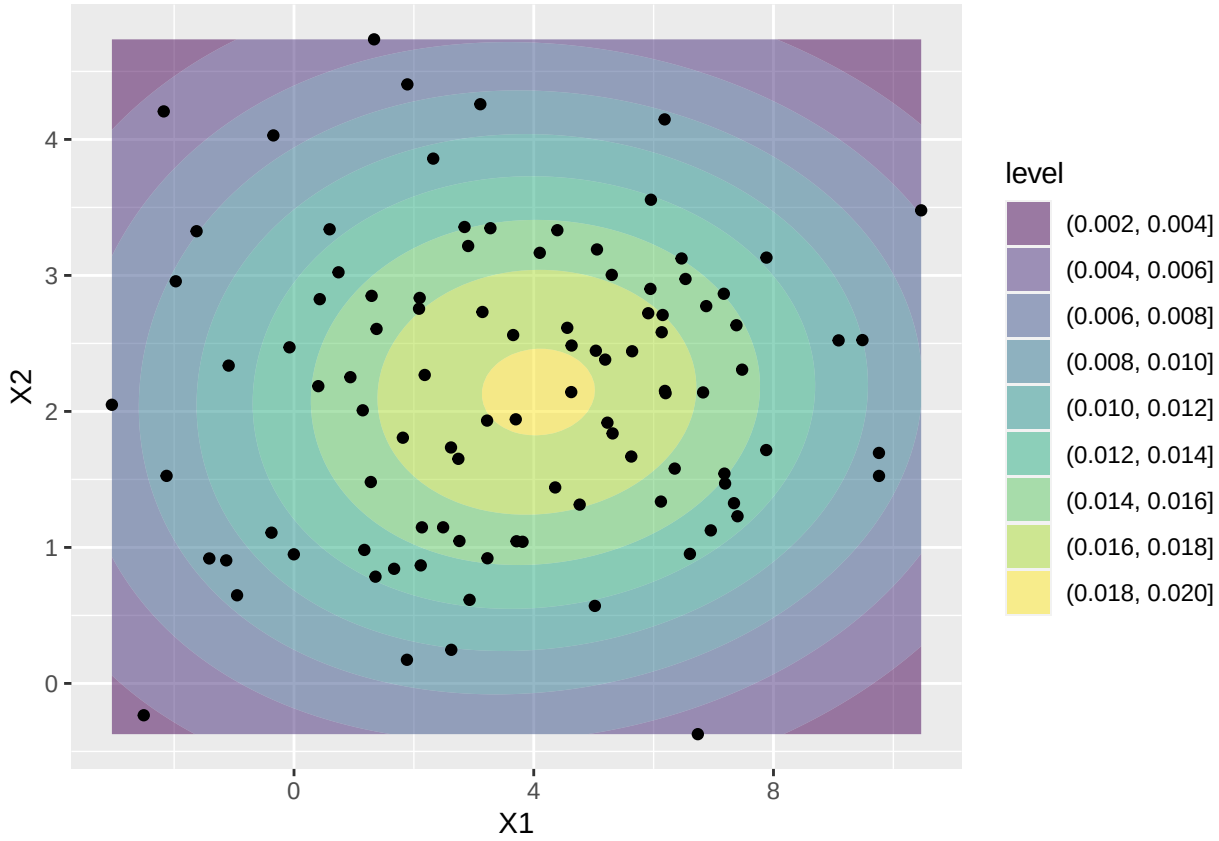

**Przykład 1:**

```
Xmass(num_vectors, B, E2)
```




**Przykład 2:**

```
Xmass(num_vectors, B, E3)
```




**Przykład 3:**

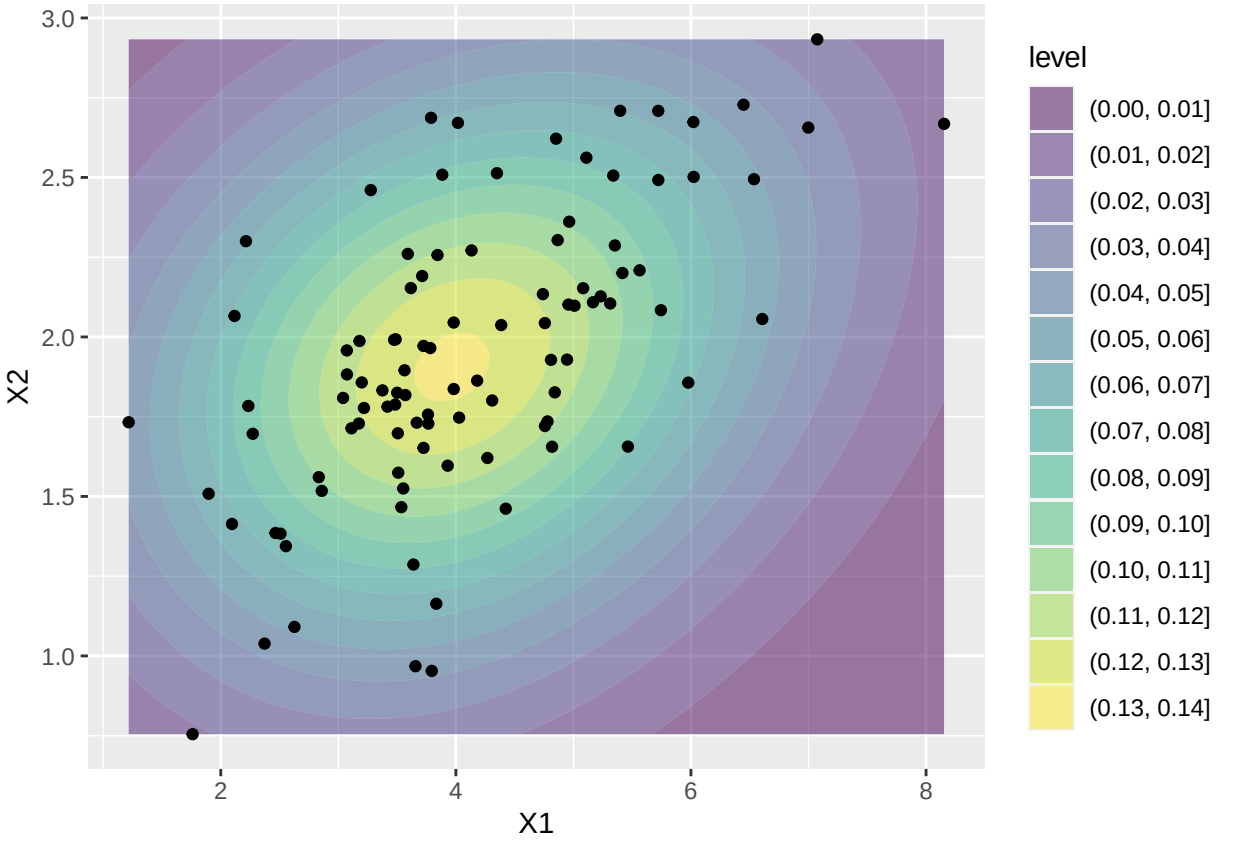
**Metoda 4: Algorytm Cholewskiego**

W tej metodzie chcemy zapisać naszą macierz w postaci$\Sigma = AA^T$, gdzie $A$ jest macierzą dolnotrójkątną. Funkcja `chol()` zwraca macierz górnotrójkątną, stąd do wyznaczenia szukanego przekształcenia musimy dokonać transpozycji po zaimplementowaniu funkcji.

```
A1c = t(chol(E1))
A2c = t(chol(E2))
A3c = t(chol(E3))
```

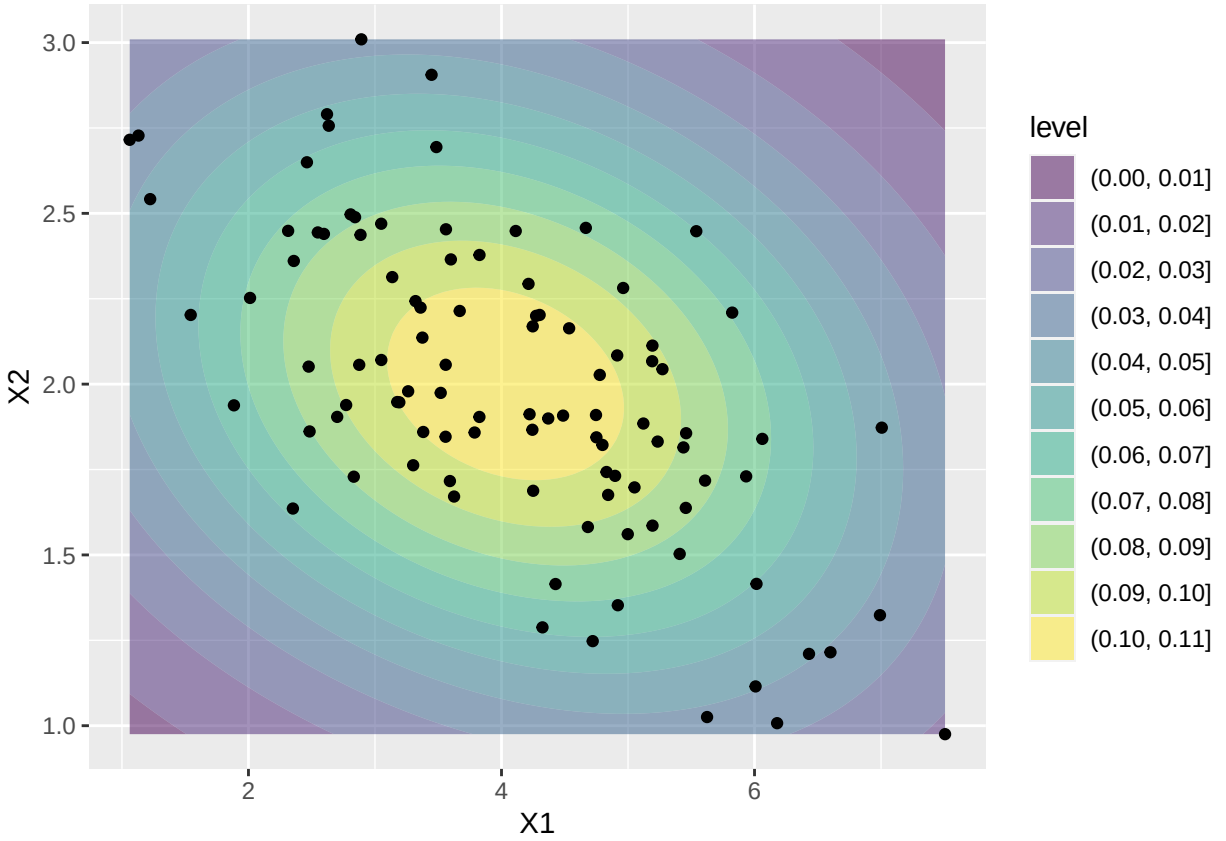
**Przykład 1**

```
cloud(A1c, B, 2, 100)
```



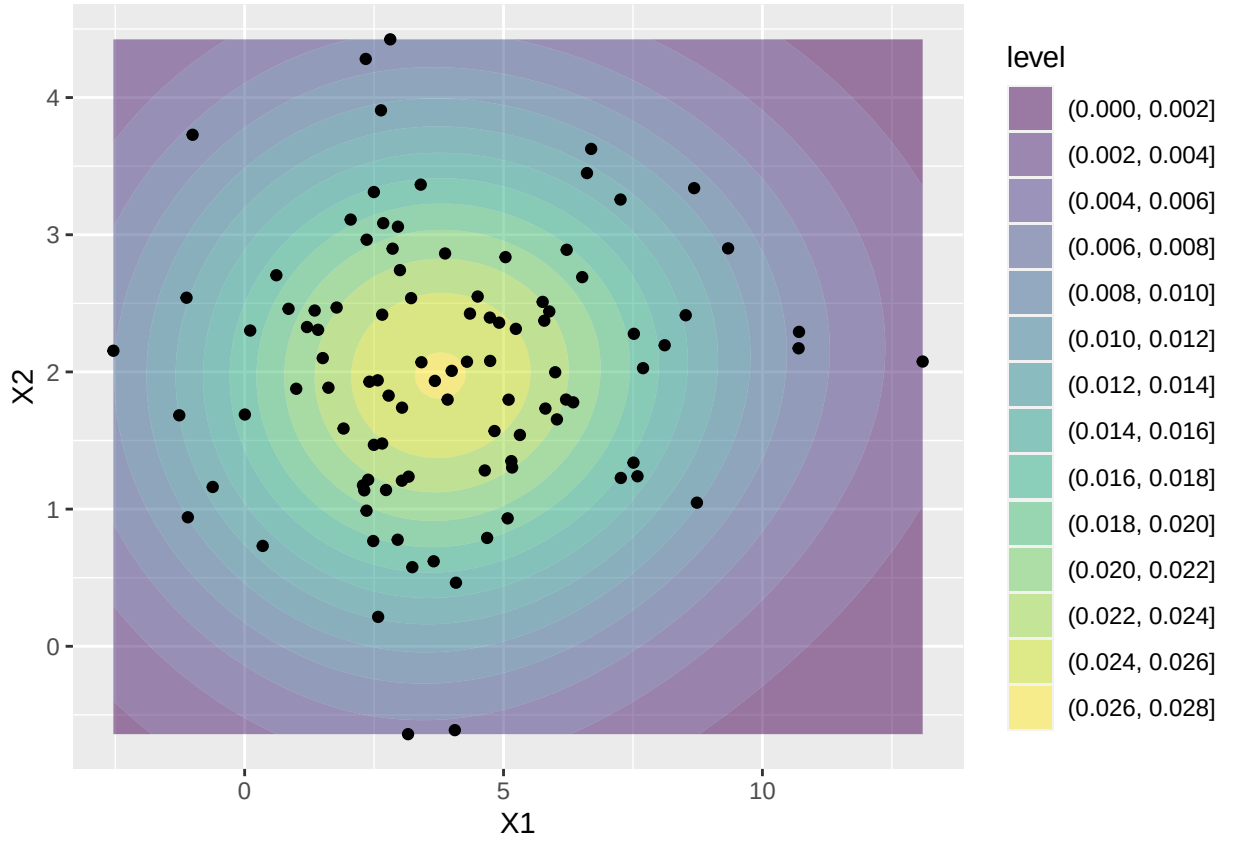

**Przykład 2**

```
cloud(A2c, B, 2, 100)
```




**Przykład 3**

```
cloud(A3c, B, 2, 100)
```



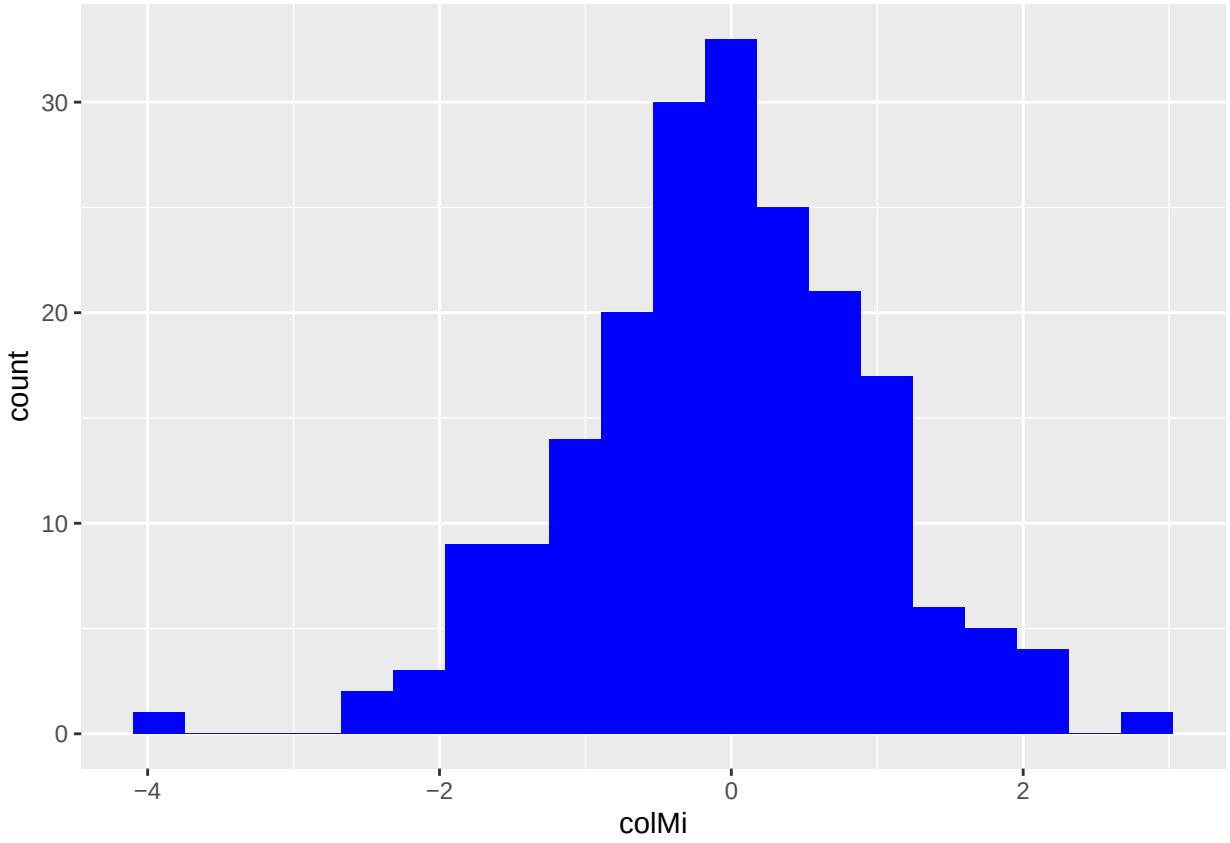

**Obserwacje i wnioski**

## Zadanie 3

To zadanie jest podobne do zadania poprzedniego, z tą różnicą, że mamy teraż większy wymiar macierzy $\Sigma$ i dwa razy więcej wektorów do wygenerowania. Do znalezienia naszego przekształcenia użyłam funkcji z pakietu MASS, ale każda z omówionych powyżej metod także by się sprawdziła. Na koniec należało zweryfikować wyniki na histogramach.

```
#Dane
num_vectors = 200
d = 100
m = c(1:100)
X <- matrix(rnorm(d * num_vectors), nrow = num_vectors)

E = matrix(0.9, nrow = 100, ncol = 100, byrow = FALSE, dimnames = NULL)

for (i in 1:d) {
  E[i,i] = 1
}

A = MASS::mvrnorm(num_vectors, mu = m*0, E)

Y = X %*% t(A)
```

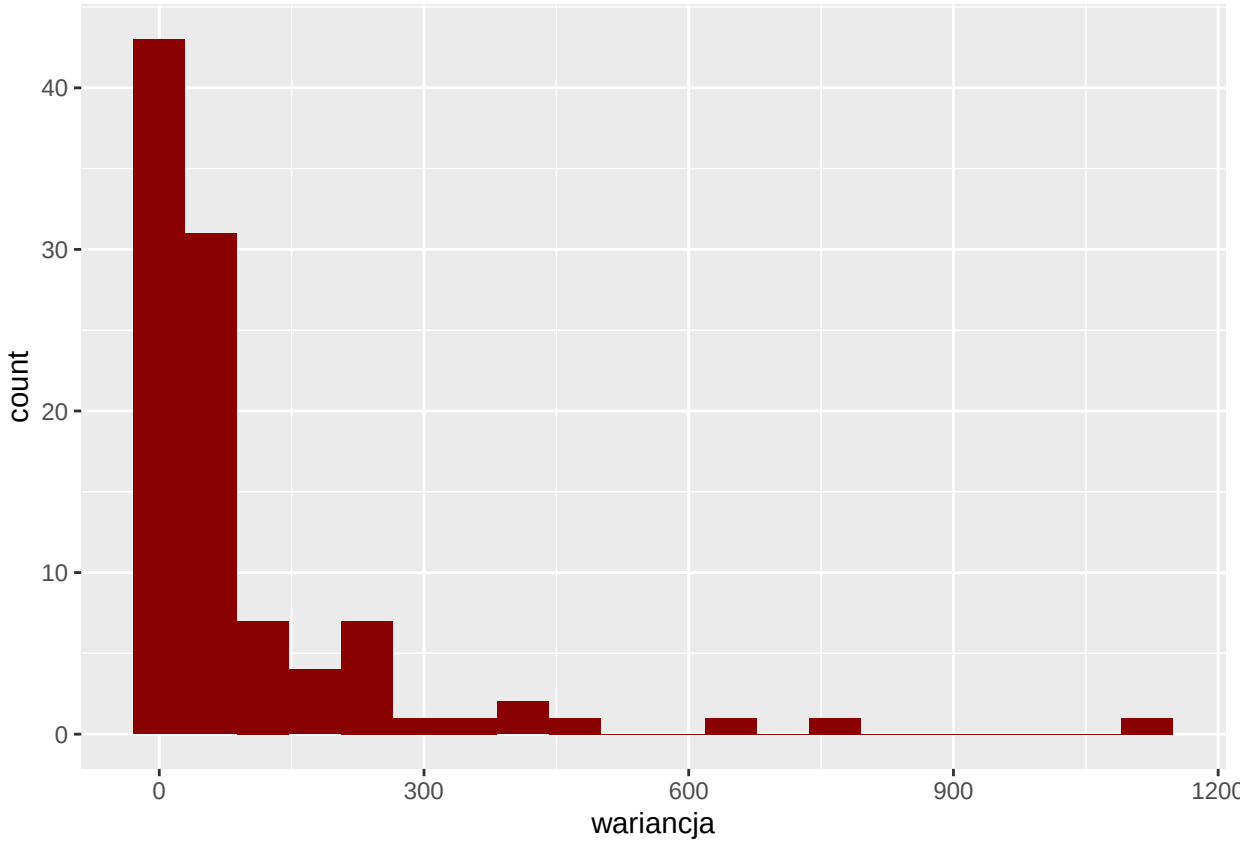
```
colMi = colMeans(Y)
ggplot() + geom_histogram(aes(colMi), bins = 20, fill = "blue")
```




```
wariancja <- c()

for (i in 1:d){
  wariancja[i] <- var(Y[,i])
}

mean(wariancja)
```

```
## [1] 99.63954
```

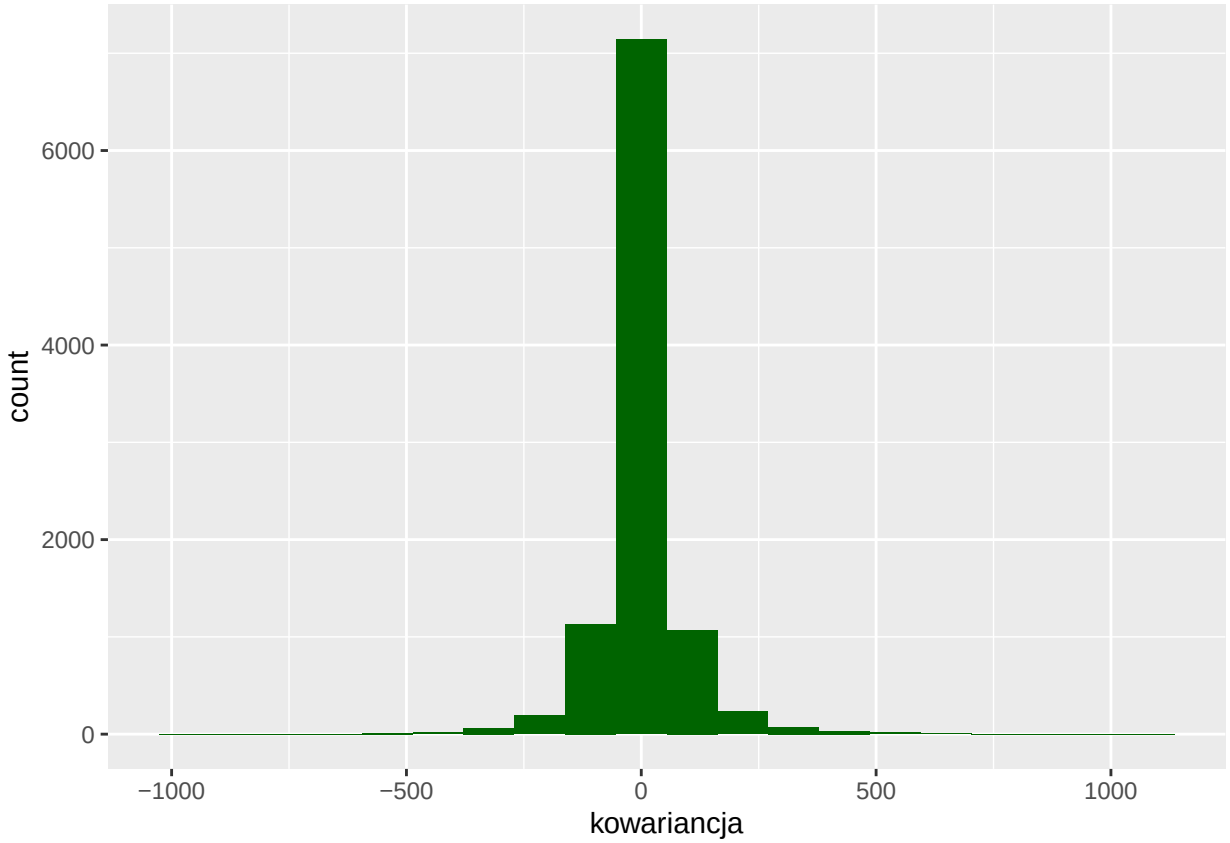
```
ggplot() + geom_histogram(aes(wariancja), bins = 20, fill = "darkred")
```




```
kowariancja <- c()

for (i in 1:100){
  for (j in 1:100){

      kowariancja <- c(kowariancja, cov(Y[,i], Y[, j]))
  }
}

mean(kowariancja)
```

```
## [1] 2.685964
```

```
ggplot() + geom_histogram(aes(kowariancja), bins = 20, fill = "darkgreen")
```




**Obserwacje i wnioski**

Wartość wariancji wyszła ok. 1, a kowariancji około 0.9, tak jak mogliśmy się spodziewać patrząc na macierz $\Sigma$, podobnie z wykresami. Średnie próbkowe również oscylują w pobliżu 0. Oznacza to, że model dość dobrze odzwierciedla nasze przekształcenie.