Raport 1

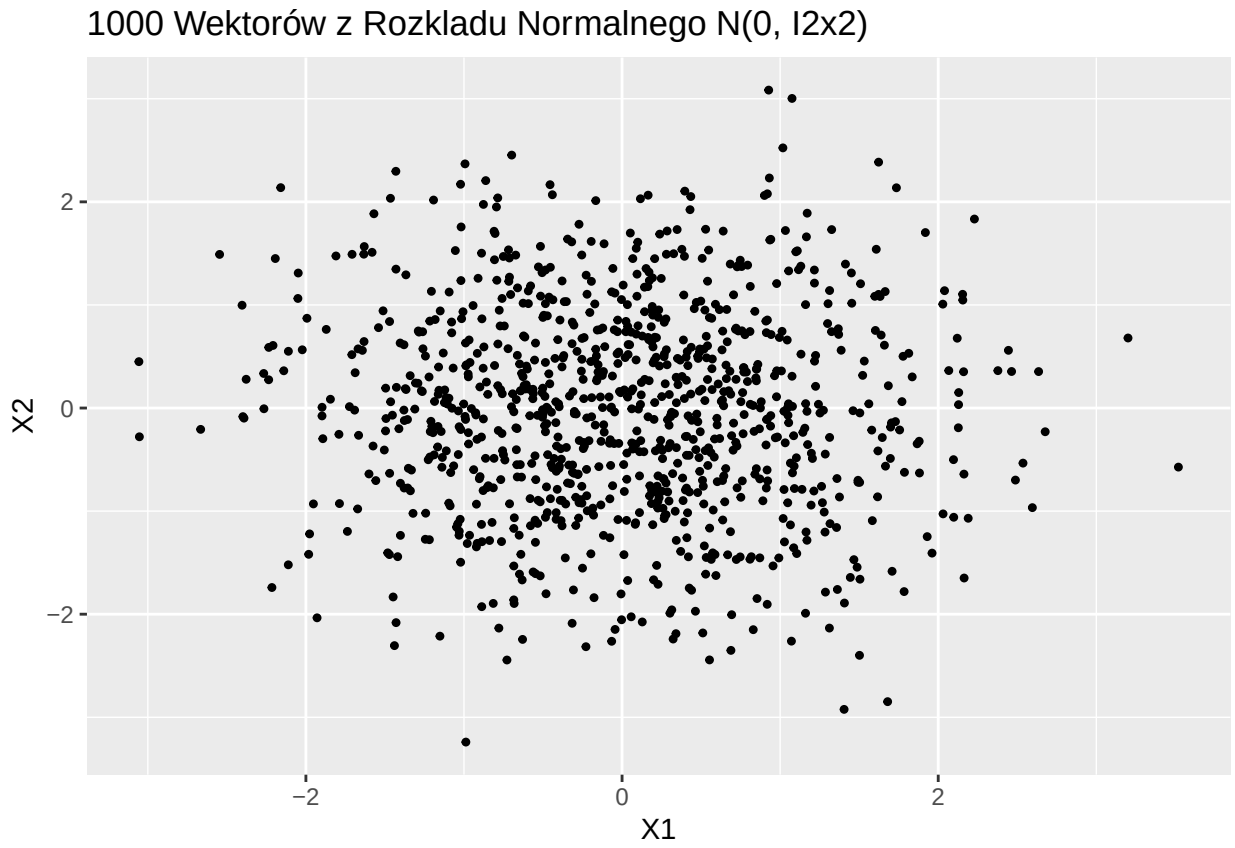
Magdalena Potok

2023-10-25

Zadanie 1

Korzystając z funkcji *rnorm* w R wygeneruję 1000 wektorów losowych z rozkładu dwuwymiarowego normalnego $N(0, I_{2x2})$ i zaznacz je na płaszczyźnie.

```
x1_sample <- rnorm(1000)
x2_sample <- rnorm(1000)
ggplot() +
  geom_point(aes(x = x1_sample, y = x2_sample), shape = 20, col = 'black') +
  labs(title = "1000 Wektorów z Rozkładu Normalnego N(0, I2x2)", x = "X1", y = "X2")
```



Wykres pokazuje wektory z rozkładu N(0,I) wygenerowane na płaszczyźnie. Gdzie 0 to wartość oczekiwana, a I to macierz kowariancji, która w tym przypadku jest macierzą jednostkową, co oznacza, że zmienne są niezależne i mają tę samą wariancję = 1. Można zaobserwować, że punkty skupiają się w otoczeniu (-2.5, 2.5)x(-2.5, 2.5), co jest zgodne z naszymi oczekiwaniami. Punkty z naszego rozkładu mają wartość oczekiwaną = 0 oraz odchylenie standardowe = 1, co oznacza, że punkty powinny być rozproszone wokół

punktu centralnego - (0,0), obszar (-2.5,2.5)x(-2.5,2.5) jest stosunkowo blisko zera i zawiera większość punktów. Łatwo również zauważyć, że im bliżej punktu (0,0) na płaszczyźnie, tym nasza chmura jest coraz gęstsza. Uzasadnieniem tego zjawiska również są parametry tego rozkładu.

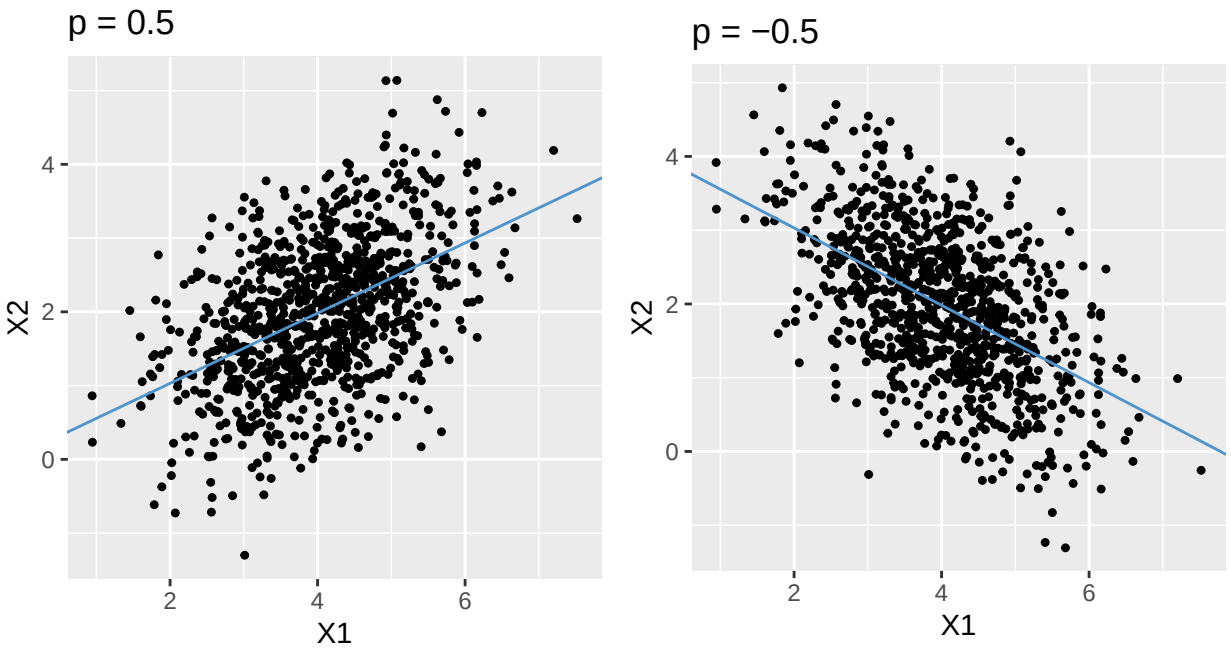
Zadanie 2

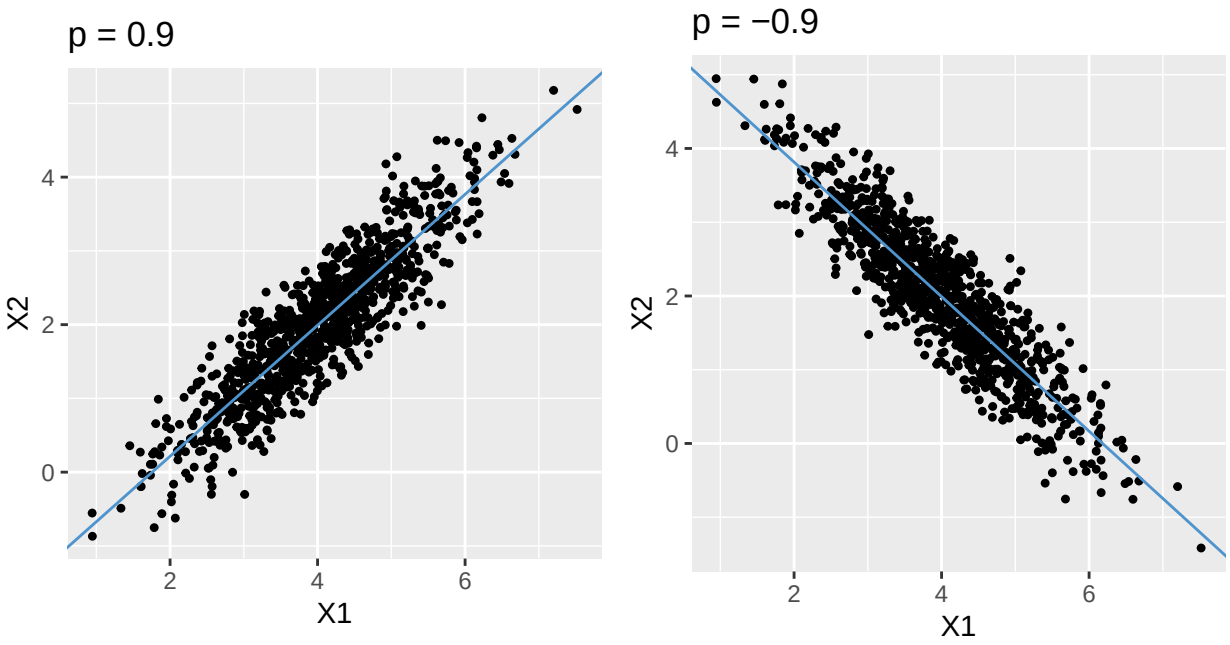
Przekształcenie liniowe, które zmienia powyższą chmurę z rozkładu $N(0, I_{2x2})$ w chmurę z rozkładu $N(\mu, \sum)$, gdzie $\mu = (4, 2)$ i $\sum = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, $\rho \in \{0.5, -0.5, 0.9, -0.9\}$.

Z wykładu wiemy, że jeśli $X \sim N(\mu', \sum')$, a $Y = AX+B$, to $Y \sim N(A\mu'+B, A\sum' A^T)$. W naszym przypadku $X \sim N(0, I_{2x2})$, więc $Y \sim N(B, AA^T)$, a więc $B = \mu = (4, 2)$ i szukamy takiej A, aby $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} = AA^T$.

Do znalezienia takiego A założymy, że jest to macierz dolno-trójkątna (w przeciwnym wypadku rozwiązanie nie byłoby jednoznaczne) i użyjemy rozkładu Choleskiego, który za pomocą komendy chol() zwraca A^T, którą następnie trzeba ztransponować, aby otrzymać macierz A i wyznaczyć $Y = AX + B$.
Poniżej przedstawię kod, który użyłam, aby otrzymać dane z rozkładu $N(\mu, \sum)$ przekształcając liniowo dane z rozkładu $N(0, I_{2x2})$.

```
przekszt_lin <- function(data, p) {
  sigma <- matrix(c(1,p,p,1),2,2)
  A <- t(chol(sigma))
  wynik <- matrix(NA, nrow = nrow(data), ncol = ncol(A))
  for (i in 1:nrow(data)) {
    wynik[i, ] <- A %*% data[i, ] + c(4,2)
  }
  return(wynik)
}
```





Wykresy przedstawiają przekształcenie liniowe chmury z 1. zadania. W każdym przypadku największe zagęszczenie znajduje się w punkcie (4,2) - co ponownie jest zgodne z naszymi oczekiwaniami, ponieważ jest to wartość oczekiwana naszego rozkładu. Dla $\rho = 0.5$ i $\rho = -0.5$ można zauważyć, że punkty ułożyły się bardziej liniowo względem pierwotnego wykresu. To co różni te dwa wykresy, to fakt, że są swoim odbiciem względem osi X2, jest to spowodowane przeciwnymi znakami w macierzy $\sum$. Natomiast, gdy $\rho = 0.9$ i $\rho = -0.9$ ułożenie punktów coraz bardziej przypomina linię prostą, co może oznaczać, że im większe ρ, tym bardziej kształt chmury będzie zbliżać się do kształtu prostej. Ponownie te dwa wykresy są swoimi odbiciami, tak, jak w przypadku wcześniejszym.
Na wykresach niebieską linią zaznaczyłam prostą, która jest modelem regresji liniowej danych rozkładów. Jest to prosta, która najbardziej przypomina osie symetrii tych chmur, jednak nie jest idealna, ponieważ chmury nie są symetryczne. Im wyższe ρ, tym lepiej ta prosta naśladuje oś symetrii, ponieważ punkty zbiegają do prostej, tak jak wyżej zauważyłam.
Poniżej znajduje się kod, który pozwolił mi na wyznaczenie parametrów prostych oznaczonych kolorem niebieskim, które znajdują się na wykresach. Kod przedstawiam dla prostej z wykresu pierwszego, dla $\rho = 0.5$, ale każdą inną prostą wyznaczałam w analogiczny sposób.

```
data <- matrix(c(x1_sample,x2_sample),ncol = 2)
wynik1 <- przekszt_lin(data,0.5)
wynik1_df <- data.frame(x1 = wynik1[,1], x2 = wynik1[,2])
reg1 <- lm(x2~x1, wynik1_df)
```

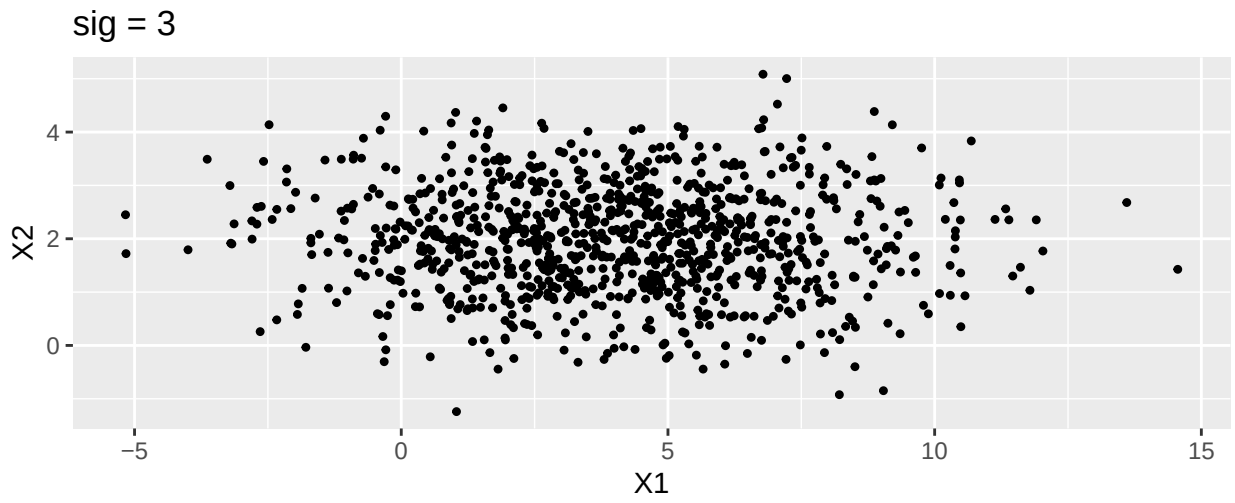
Zadanie 3

W tym zadaniu przekształcę chmurę punktów z zadania 1 na chmurę z rozkładu $N(\mu, \sum)$, gdzie

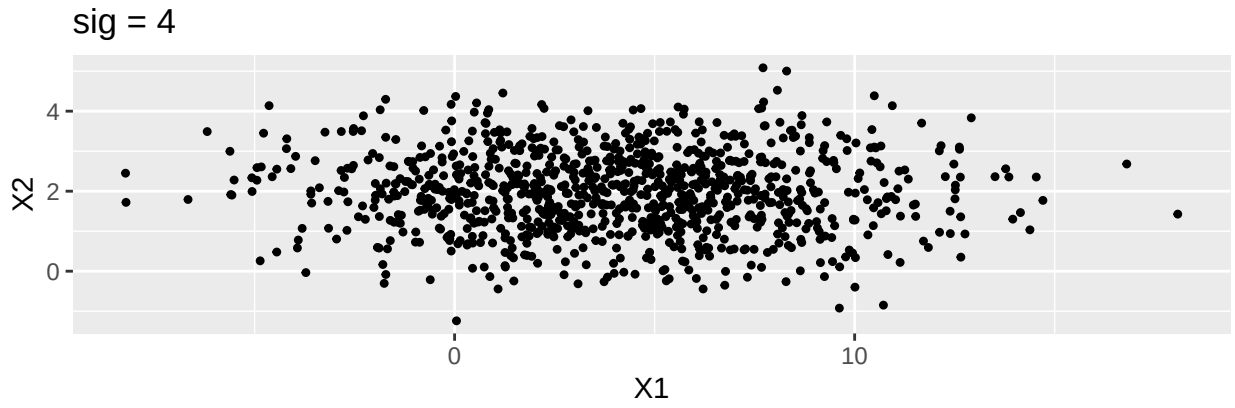
$$\mu = (4, 2),\ \sum = \begin{pmatrix} \sigma^2 & 0 \\ 0 & 1 \end{pmatrix},\ \sigma \in \{3, 4\}$$

. Poniżej znajduje się kod, którego użyłam do wyznaczenia danych dla rozkładu z $N(\mu, \sum)$.

```
przekszt_lin_3 <- function(data, sig) {
  sigma <- matrix(c(sig^2,0,0,1),2,2)
  A <- t(chol(sigma))
  wynik <- matrix(NA, nrow = nrow(data), ncol = ncol(A))
  for (i in 1:nrow(data)) {
    wynik[i, ] <- A %*% data[i, ] + c(4,2)
  }
  return(wynik)
}
```



Wykres ilustruje przekształćenie danych z rozkładu normalnego $N(0, I_{2x2})$ w nowy rozkład normalny $N(\mu, \sum)$. Można zaobserwować jak zmienia się rozmieszczenie danych (porównując z zadaniem 1.) w wyniku tego przekształćenia liniowego. Większość punktów z tego rozkładu znajduje się w obszarze (-5,13)x(-1,5), to rozmieszczenie jest uzasadnione parametrami naszego rozkładu. Macierz kowarianski $\sum$ to macierz diagonalna, która ma różne wariancje dla obu zmiennych X1 i X2. Wariancja dla X1 wynosi σ^2, czyli w tym przypadku 9, stąd takie szerokie rozmieszczenie punktów na osi X1. Gdzie dla X2 wariancja wynosi jedynie 1, dlatego oś X2 jest o wiele mniejsza. Wartość oczekiwana rozkładu $\mu = (4, 2)$, co widać na wykresie, że właśnie dla punktu (4,2) nasza chmura jest najbardziej zagęszczona.



Wykres przedstawia drugi przypadek przekształcenia chmury punktów z rozkładu $N(0, I_{2x2})$ w rozkład $N(\mu, \sum)$, gdzie $\mu = 4$. Różnicę, którą można zaobserwować jest to, że punkty na osi X1 są jeszcze bardziej rozproszone. Jest to spowodowane tym, że ich wariancja jest jeszcze większa, wcześniej wynosiła 9, teraz 16. Punkty chmury znajdują się głównie na obszarze (-12,20)x(-1,5), jak widać zmienił się przedział jedynie pierwszej współrzędnej, bo właśnie tej współrzędnej wariancja się zmieniła. Ponownie największe zagęszczenie jest w okolicach punktu (4,2), ponieważ wartość oczekiwana nie została zmieniona.

Zadanie 4

1. Korzystając z funkcji **rnorm** w R wygeneruję 1000 wektorów losowych z rozkładu wielowymiarowego normalnego $N(0, I_{100x100})$. Wyniki zostały zapisane w macierzy $X_{1000x100}$, której wiersze zawierają kolejne wygenerowane wektory losowe.

```
X <- matrix(rnorm(100000), 1000, 100)
```

2. Wyznaczę macierz A, tak aby $\hat{X} = XA$ zawierała 1000 wektorów z rozkładu wielowymiarowego normalnego $N(0, \sum_{100x100})$, gdzie $\sum(i,i) = 1$ i $\sum(i,j) = 0.9$ dla $i \neq j$.
   Aby wyznaczyć taką macierz A ponownie użyję rozkładu choleskiego, rozwiązując równanie:

$$\hat{A^T}\hat{A} = \sum,$$

   gdzie

$$\hat{A} = A^T$$

   W następujący sposób:

```
cov_matrix <- matrix(0.9, 100, 100)
diag(cov_matrix) <- 1
A <- t(chol(cov_matrix))
```

Następnie nasz $\hat{X}$ powstaje w ten sposób:

```
h_X <- matrix(0, 1000, 100)
for(i in 1:1000){
  h_X[i,] <-  A %*% X[i,]
}
```

3. Zweryfikuję otrzymane wyniki...

- wyliczając średnią współrzędnych

```
mean(colMeans(h_X))
```

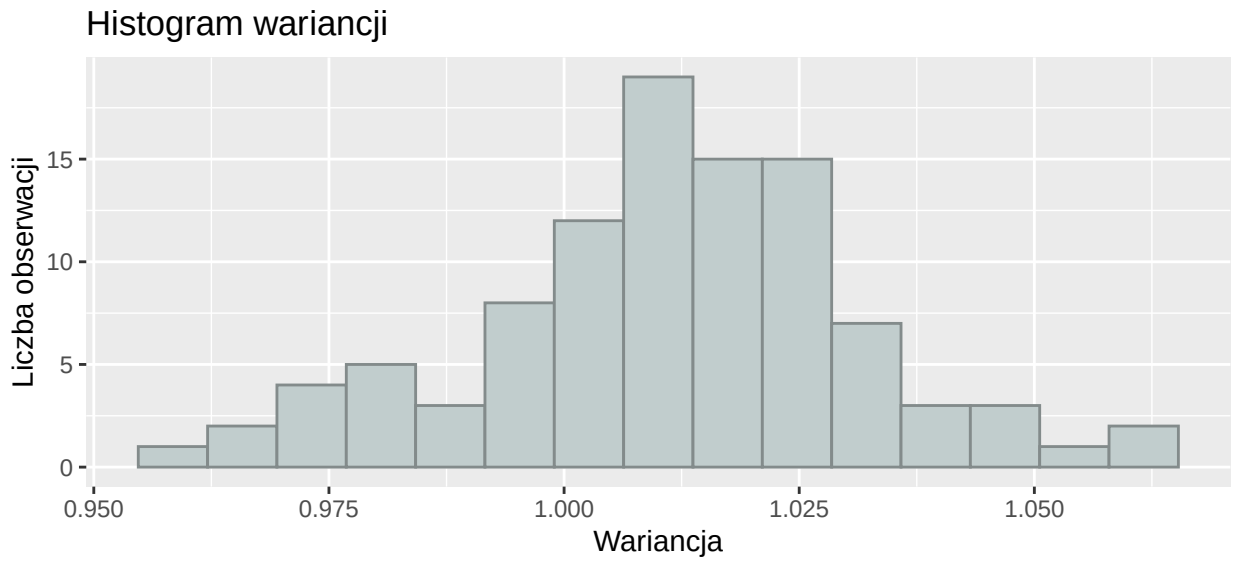
```
## [1] -0.02271522
```

Wartość średnia jest bardzo bliska 0, co jest spodziewanym dla nas wynikiem, ponieważ taka jest wartość oczekiwana naszego rozkładu.

- rysując histogram próbkowych wariancji współrzędnych

```
var <- apply(h_X, 2, var)

ggplot(data.frame(var), aes(x = var)) +
  geom_histogram(bins = 15, fill = "azure3", color = "azure4") +
  labs(x = "Wariancja", y = "Liczba obserwacji") +
  ggtitle("Histogram wariancji")
```



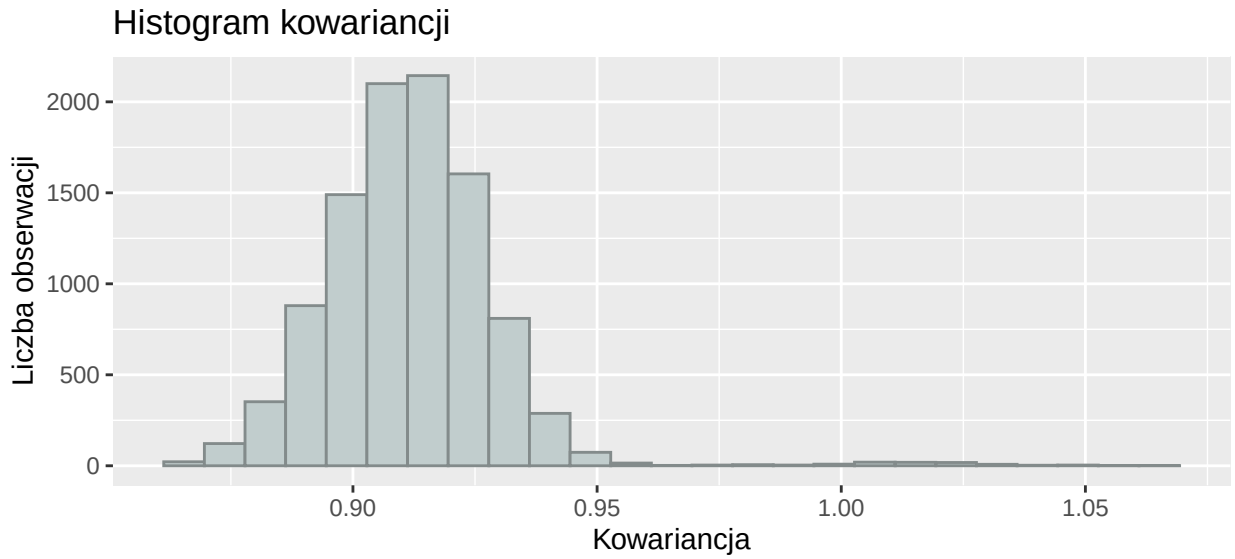
Histogram wyglądem przypomina rozkład normalny. Widać, że jego największy słupek znajduje się niedaleko wartości 1, co ponownie jest tym, czego się spodziewamy, ponieważ wariancje współrzędnych (przekątna macierzy kowariancji) powinny być bliskie 1. Warto zauważyć, że istnieją odchylenia od tej wartości, ale jest to naturalne dla próbek losowych.

- rysując histogram próbkowych kowariancji między różnymi współrzędnymi

```
kow <- cov(h_X)
kow_v <- as.vector(kow)

ggplot(data.frame(Kowariancja = kow_v), aes(x = Kowariancja)) +
  geom_histogram(bins = 25, fill = "azure3", color = "azure4") +
```

```
  labs(x = "Kowariancja", y = "Liczba obserwacji") +
  ggtitle("Histogram kowariancji")
```



Kształt histogramu przypomina rozkład normalny lub rozkład Gaussa. Najwyższy słupek znajduje się w okolicy 0.87, co jest ponownie spodziewaną wartością, ponieważ wartość kowariancji między różnymi wektorami macierzy $\hat{X}$ to liczby z macierzy $\sum$, które nie znajdują się na przekątnej - czyli dokładnie 0.9. Widać, że wartości kowariancji są dodatnie, znaczy to, że obie zmienne rosną razem.

Zadanie dodatkowe

Napisałam własną funckję implementującą rozkład Choleskiego, czyli funkcję która po podaniu jej symetrycznej macierzy dodatnio określonej (A) zwraca jej rozkład Choleskiego w postaci macierzy dolno-trójkątnej (L). Jest to procedura rozkładu macierzy A na iloczyn postaci $A = LL^T$. Tak wygląda ten proces:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & \dots & l_{n1} \\ 0 & l_{22} & \dots & l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_{nn} \end{pmatrix}$$

Zatem algorytm szukania współczynników macierz L wygląda następująco:

$$a_{11} = l_{11}^2 \to l_{11} = \sqrt{a_{11}}$$

$$a_{21} = l_{21} l_{11} \to l_{21} = \frac{a_{21}}{l_{11}}$$

$$a_{22} = l_{21}^2 + l_{22}^2 \to l_{22} = \sqrt{a_{22} - l_{21}^2}$$

$$a_{32} = l_{31} l_{21} + l_{32} l_{22} \to l_{32} = \frac{a_{32} - l_{31} l_{21}}{l_{22}} \dots$$

W ogólności współczynniki znajdujemy w ten sposób:

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$$

$$l_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}}{l_{ii}}$$

Zgodnie z tymi wyliczeniami napisałam następujący kod:

```
rozkl_chol <- function(A) {
  n <- nrow(A)
  L <- matrix(0, n, n)

  for (j in 1:n) {
    for (i in j:n) {
      if (i == j) {
        L[i, j] <- sqrt(A[i, i] - sum(L[i, 1:(i-1)]^2))
      } else {
        L[i, j] <- (A[i, j] - sum(L[i, 1:(i-1)] * L[j, 1:(i-1)])) / L[j, j]
      }
    }
  }

  return(L)
}
```

Przykładowe użycie funkcji:

```
A <- matrix(c(4, 12, 12, 41), 2, 2)
rozkl_chol(A)
```

```
##      [,1]     [,2]
## [1,]    2 0.000000
## [2,]    6 2.236068
```