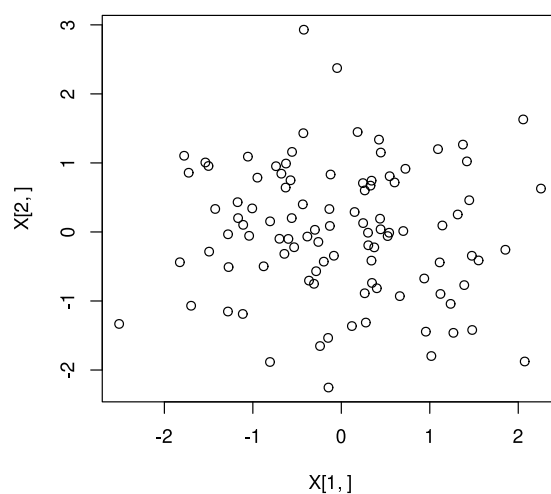


Modele Liniowe - Lista 1

Jakub Kuciński 309881

Październik 2021

1 Zadanie 1



Rysunek 1: Sto wektorów pochodzących z dwuwymiarowego rozkładu normalnego $N(0, I)$.

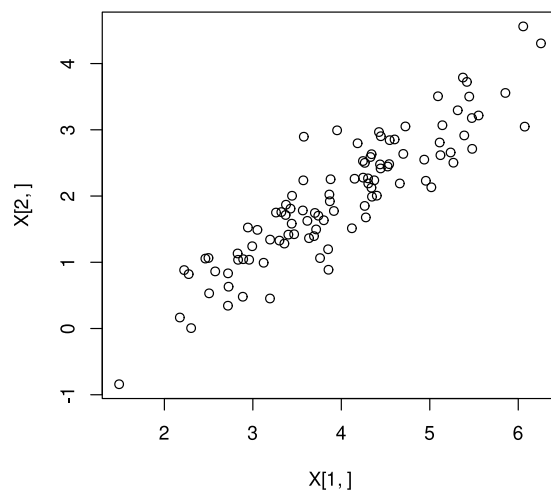
Zadanie polega na wygenerowaniu 100 losowych wektorów z dwuwymiarowego rozkładu normalnego $N(0, I)$ przy użyciu funkcji *rnorm* i przedstawieniu ich na wykresie. Funkcja *rnorm* pozwala na generowanie zmiennych wyłącznie z jednowymiarowego rozkładu normalnego. Zauważmy jednak, że skoro macierz wariancji-kowariancji jest w naszym zadaniu macierzą identycznościową, to wariancja poszczególnych zmiennych jest równa 1, a ich kowariancja 0. Korzystając z faktu, że kowariancja zmiennych losowych z rozkładów normalnych jest równa zeru wtedy i tylko wtedy, gdy zmienne są niezależne, możemy niezależnie losować

wartości dla poszczególnych zmiennych, czyli współrzędnych wektora. Dodatkowo skoro generowanie poszczególnych wektorów jest niezależne, to wystarczy wylosować 200 wartości z jednowymiarowego rozkładu normalnego $N(0, 1)$.

Wykres otrzymanych w ten sposób wektorów znajduje się na rysunku 1. Widzimy, że środek chmury punktów znajduje się w okolicy punktu $(0, 0)$, co jest zgodne ze średnią rozkładu $N(0, I)$. Widzimy też, że największe zagęszczenie punktów ma miejsce w $[-1, 1] \times [-1, 1]$, im dalej od środka tym jest mniejsze i najrzadsze w rogach wykresu, gdzie odchylenia od średniej są największe. Jest to zachowanie zgodne z rozkładem $N(0, I)$.

2 Zadanie 2

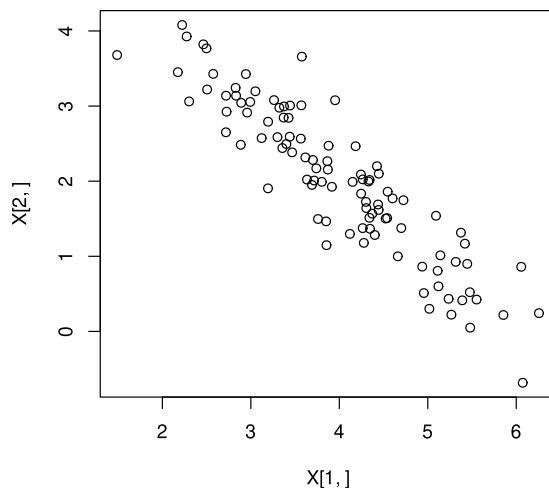
Wiemy, że wektory X z poprzedniego zadania pochodzą z rozkładu $N_x(0, I)$. Wiemy również, że dla danej macierzy A i wektora B zmienna $Y = AX + B$ pochodzi z rozkładu normalnego $N_y(\mu_Y, \Sigma_Y)$, gdzie $\mu_Y = A\mu_X + B = A \cdot 0 + B = B$ oraz $\Sigma_Y = A\Sigma_X A^T = A I A^T = A A^T$. Zatem w miejsce B wystarczy użyć μ_Y , a macierz A można wyznaczyć z rozkładu Choleskiego macierzy Σ_Y . Wektory z rozkładu $N_y(\mu_Y, \Sigma_Y)$ zostały wytworzone przez zastosowanie przekształcenia liniowego $Y = AX + B$ na wektorach wylosowanych z rozkładu $N_x(0, I)$ z poprzedniego zadania.



Rysunek 2: $\mu = \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$

Na rysunku 2 zgodnie z oczekiwaniem średnia współrzędnych punktów znajduje się w okolicach punktu $(4, 2)$. Widzimy też, że duża wartość dla jednej

współrzędnej koreluje z dużą wartością dla drugiej, co jest zgodne z wysoką korelacją tych zmiennych (macierz Σ). Rozrzut punktów wzdłuż poszczególnych osi jest podobny co zgadza się z jednakową wariancją tych zmiennych.



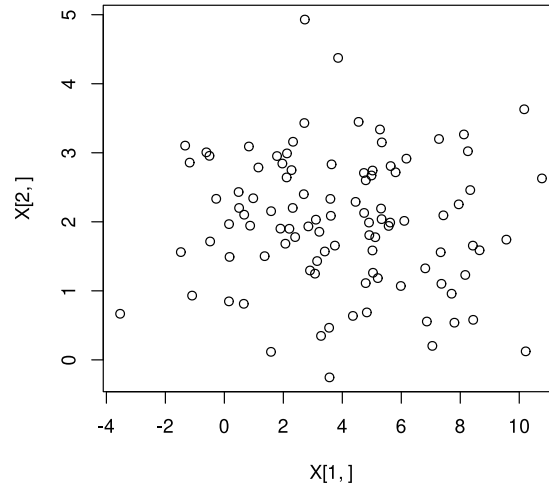
Rysunek 3: $\mu = \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$

Wnioski co do odchyień punktów i ich średniej dla rysunku 3 są analogiczne jak dla rysunku 2. Widzimy też, że wyższa wartość dla jednej zmiennej koreluje z niższą wartością drugiej zmiennej, co jest zgodne z ich wysoce negatywną korelacją (macierz Σ).

Rysunek 4 jest identyczny, co do rozmieszczenia punktów, jak rysunek 1. Wynika to z faktów, że w obu rozkładach zmienne są niezależne, centrum wykresu zostało przesunięte z $(0, 0)$ na $(4, 2)$ oraz trzykrotnie zwiększona została skala na osi OX (wariancja pierwszej zmiennej zmieniła się z 1 na 9, co oznacza, że odchylenie standardowe wzrosło trzykrotnie).

3 Zadanie 3

Macierz X z rozkładu $N(0, I)$ jest w tym zadaniu macierzą wektorów ustawionych wierszowo. Poszczególne zmienne dalej są niezależne, więc sposób losowania wartości tej macierzy nie zmienia się. Chcemy znaleźć taką macierz A , że $\tilde{X} = XA$, że wektory z \tilde{X} pochodzą z rozkładu opisanego w zadaniu. Zauważmy, że jakbyśmy wykonali obustronnie transpozycje na powyższym równaniu, to otrzymujemy $\tilde{X}^T = A^T X^T$, gdzie X^T i \tilde{X}^T zawierają wektory kolumnowe.

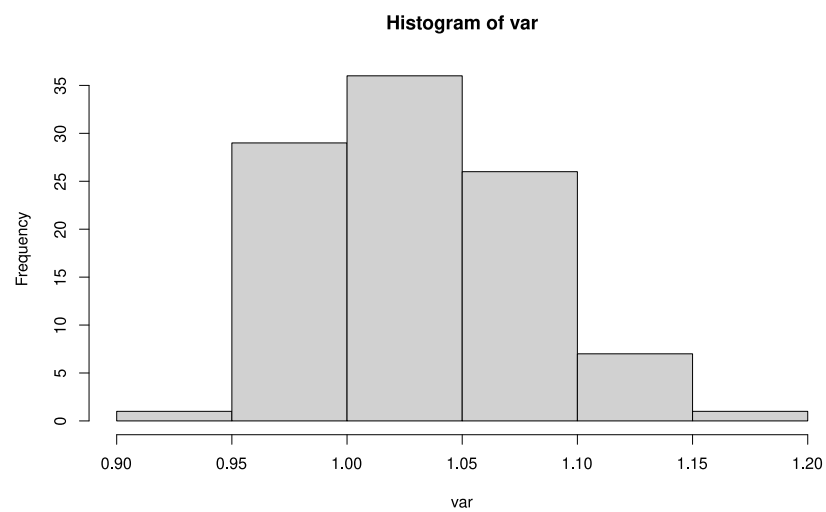


Rysunek 4: $\mu = \begin{pmatrix} 4 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$

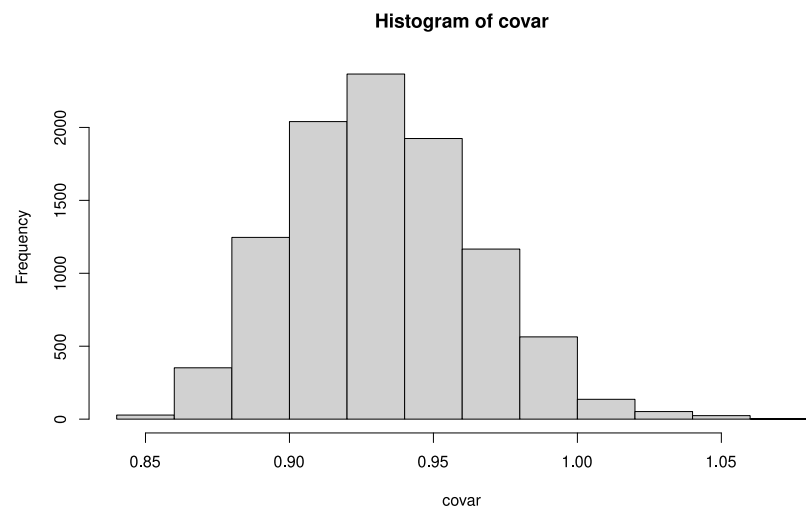
Możemy zatem uzyskać macierz A^T w taki sam sposób, jak w poprzednim zadaniu, a następnie wykonać na niej transpozycję i pomnożyć lewostronnie przez X otrzymując finalnie macierz \bar{X} o szukanym rozkładzie.

Widzimy, że zgodnie z oczekiwaniami populacyjne wariancje zmiennych są w okolicach wartości 1 (rysunek 5), czyli właściwej wariancji rozkładu, z którego pochodzą próbki. Dokładna średnia tych wariancji wynosi 1.032629.

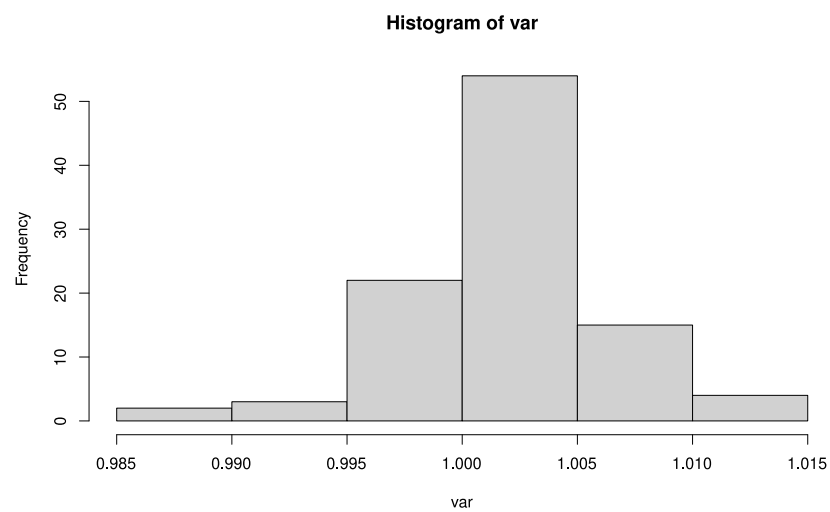
Podobnie populacyjne kowariancje zmiennych koncentrują się w okolicy 0.9 (rysunek 6), czyli prawdziwych kowariancji rozkładu. Dokładna średnia wyniosła 0.9324107. Aby z większą pewnością stwierdzić, że zmienne rzeczywiście pochodzą z wybranego rozkładu, można wylosować ich większą liczbę np. 20000. Dzięki większej próbie, wylosowane wartości lepiej aproksymują rozkład, z którego pochodzą. W granicy średnie wariancji i kowariancji wynosiłyby wartości 1 i 0.9. Faktycznie z wykresów 7 i 8 widzimy, że wariancje i kowariancje zmiennych znajdują się bliżej rzeczywistym wartościom. Ich średnie wyniosły tym razem odpowiednio 1.001805 i 0.9016942.



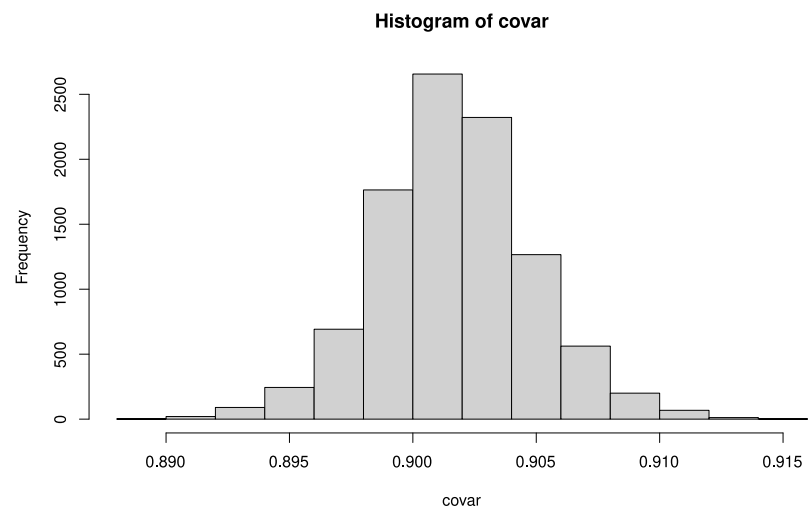
Rysunek 5: Histogram wariancji \bar{X} .



Rysunek 6: Histogram kowariancji \bar{X} .



Rysunek 7: Histogram wariancji \bar{X} .



Rysunek 8: Histogram kowariancji \bar{X} .

4 Kod w R

```
##### 1)
X = matrix(rnorm(200), 2, 100)
plot(X[1, ], X[2, ])

##### 2)
mu = matrix(c(4,2), 2, 1)
Sigma1 = matrix(c(1, 0.9, 0.9, 1), 2, 2)
Sigma2 = matrix(c(1, -0.9, -0.9, 1), 2, 2)
Sigma3 = matrix(c(9, 0, 0, 1), 2, 2)

CholeskyDecompositionTwoOnTwo <- function(A)
{
  x = sqrt(A[1,1])
  y = A[2,1]/x
  z = sqrt(A[2,2] - y*y)
  matrix(c(x, y, 0, z), 2, 2)
}

MakeSubtaskOfSecondExercise <- function(X, mu, Sigma)
{
  A = CholeskyDecompositionTwoOnTwo(Sigma)
  X = A %*% X + c(mu[1,1], mu[2,1])
  print(X)
  plot(X[1,], X[2,])
}

MakeSubtaskOfSecondExercise(X, mu, Sigma1)
MakeSubtaskOfSecondExercise(X, mu, Sigma2)
MakeSubtaskOfSecondExercise(X, mu, Sigma3)

##### 3)
CholeskyDecomposition <- function(A)
{
  dimensionality = dim(A)[1]
  L = matrix(0, dimensionality, dimensionality)
  for (i in 1:dimensionality){
    for (j in 1:i){
      sum = sum(L[i, 1:j] * L[j, 1:j])
      if (i == j){
        L[i,j] = sqrt(A[i,i] - sum)
      }
      else{
        L[i,j] = (1.0 / L[j,j] * (A[i,j] - sum))
      }
    }
  }
}
```

```

    }
  }
}
L
}

GetVariances <- function(M)
{
  n = dim(M)[1]
  variances = matrix(0, 1, n)
  for(i in 1:n){
    variances[1, i] = M[i, i]
  }
  variances
}

GetCovariances <- function(M)
{
  n = dim(M)[1]
  covariances = matrix(0, 1, n*n - n)
  k = 1
  for(i in 1:n){
    for(j in 1:n){
      if (i != j){
        covariances[1, k] = M[i, j]
        k = k + 1
      }
    }
  }
  covariances
}

ThirdTask <- function(num_of_vectors, d)
{
  X = matrix(rnorm(num_of_vectors*d), num_of_vectors, d)
  Sigma = matrix(0.9, d, d)
  for(i in 1:d)
  {
    Sigma[i, i] = 1
  }

  A = CholeskyDecomposition(Sigma)
  XDashed = X %*% t(A)
  C = cov(XDashed)

  var = GetVariances(C)
}

```



```
    covar = GetCovariances(C)
    print(mean(var))
    print(mean(covar))
    hist(var)
    hist(covar)
}
ThirdTask(200, 100)
ThirdTask(20000, 100)
```