Zaawansowane modele liniowe - lista 4

Klaudia Weigel

1 Zadanie 1

Przyjmujemy oznaczenia:

- n - liczba obiektów,
- k – liczba pomiarów na każdym obiekcie,
- p – liczba kolumn w macierzy planu,
- $N = n * k$ - liczba zmiennych objaśnianych y_{ij}.

1.1 Podpunkt a

Dla $n = 20, k = 3, p = 4$ wygenerujemy macierz $X \in \mathbb{M}_{N\times p-1}$ taką, że jej elementy są niezależnymi realizacjami z rozkładu $N(0, 1/\sqrt{N})$. Do macierzy dodamy również kolumnę jedynek odpowiadającą interceptowi. Następnie podzielimy macierz na $n = N/k$ podmacierzy $X_1, \dots, X_n \in \mathbb{M}_{k\times p-1}$.

```
n = 20
k = 3
p = 4
N = n*k

X = cbind(1, matrix(rnorm(N*(p-1), sd = 1/sqrt(N)), nrow = N, ncol = p-1))
Xsplit = lapply(split(X, rep(c(1:n),each=k)), matrix, nrow=k)

beta = c(0,3,3,0)
rho = 0.3
gamma = 2
sigma = matrix(rho, nrow = k, ncol = k)
diag(sigma) = 1
sigma = gamma^2*sigma
```

Przyjmujemy $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)' = (0, 3, 3, 0)'$ oraz

$$\Sigma = \gamma^2 \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{pmatrix} \in \mathbb{M}_{k\times k},$$

gdzie $\gamma = 2$ oraz $\rho = 0.3$.

1.2 Podpunkt b

Wygenerujemy n niezależnych wektorów losowych

$$y_i = (y_{i1}, \dots, y_{ik})' \sim N(X_i\beta, \Sigma) \in \mathbb{R}^k, \quad i = 1, 2, \dots, n.$$

Zapiszemy dane w jednowymiarowej reprezentacji.

```
Y = lapply(Xsplit, function(X) rmvnorm(1, mean = X%*%beta, sigma = sigma))

data_uni = lapply(1:n, function(i) cbind(t(Y[[i]]), rep(i, k), 1:k, Xsplit[[i]]))
data_uni = do.call(rbind, data_uni)
data_uni = data.frame(data_uni)
colnames(data_uni) = c('y', 'id', 'T','X0', 'X1', 'X2', 'X3')
head(data_uni)
```

```
##            y id T X0          X1          X2          X3
## 1 -1.3663981  1 1  1 -0.11414299 -0.13855508  0.08415341
## 2  2.5906389  1 2  1 -0.11926290 -0.05390070 -0.23656952
## 3 -1.4295557  1 3  1 -0.11970749 -0.06436812 -0.24339448
## 4  1.3175352  2 1  1  0.18548876 -0.04251278 -0.07955364
## 5 -0.2434386  2 2  1  0.05461402  0.11361755 -0.04397249
## 6 -0.6612170  2 3  1 -0.25755047  0.06615645 -0.02359897
```

Za pomocą funkcji `gls` zbudujemy model liniowy.

```
m1 = gls(y~X1+X2+X3, correlation = corCompSymm(form = ~1|id),
         weights = varIdent(form = ~1), method = "REML", data = data_uni)
```

1.3 Podpunkt c

1.3.1 Wektor β

Estymator wektora β, otrzymujemy z

$$\hat{\beta} = \left(\sum_{i=1}^{n} X_i'\Sigma^{-1}X_i\right)^{-1}\left(\sum_{i=1}^{n} X_i'\Sigma^{-1}y_i\right).$$

W tym punkcie za Σ podstawimy estymator otrzymany metodą REML ($\hat{\Sigma}_{REML}$).

Estymator wektora β ma asymptotycznie rozkład

$$\hat{\beta} \to^d N\left(\beta, \left(\sum_{i=1}^{n} X_i'\hat{\Sigma}^{-1}X_i\right)^{-1}\right).$$

W zadaniu będziemy korzystać z normy supremum, która dla wektora x jest zdefiniowana jako:

$$\|x\|_{sup} = \max_i |x_i|.$$

Natomiast dla macierzy X:

$$\|X\|_{sup} = \max_{i,j} |x_{ij}|.$$

```
sigma_reml = getVarCov(m1)

a1 = lapply(1:n, function(i) t(Xsplit[[i]])%*%solve(sigma_reml)%*%Xsplit[[i]])
a1 = Reduce('+', a1) # zsumuj n macierzy
a1 = solve(a1)

b1 = lapply(1:n, function(i) t(Xsplit[[i]])%*%solve(sigma_reml)%*%t(Y[[i]]))
b1 = Reduce('+', b1)

beta_est_reml = a1%*%b1
```

```
t(beta_est_reml)
```

```
##            [,1]     [,2]    [,3]     [,4]
## [1,] 0.04925239 3.698328 3.46146 2.185057
```

Porównamy wynik z tym zwracanym przez funkcję `gls`

```
coef(m1)
```

```
## (Intercept)          X1          X2          X3
##  0.04925239  3.69832823  3.46146001  2.18505666
```

Estymatory są takie same.

Norma supremum dla różnicy $\hat{\beta}$ oraz prawdziwych wartości:

```
max(abs(beta_est_reml - beta))
```

```
## [1] 2.185057
```

Spójrzmy teraz na macierz kowariancji wektora β

```
a1
```

```
##              [,1]        [,2]         [,3]        [,4]
## [1,]  0.091559097 -0.06035025 -0.002099436  0.05391394
## [2,] -0.060350245  2.51067175  0.117472239 -0.20896850
## [3,] -0.002099436  0.11747224  1.839375248 -0.28335792
## [4,]  0.053913941 -0.20896850 -0.283357922  2.62764304
```

Porównajmy wynik z funkcją `vcov`

```
vcov(m1)
```

```
##               (Intercept)          X1           X2          X3
## (Intercept)  0.091559097 -0.06035025 -0.002099436  0.05391394
## X1          -0.060350245  2.51067175  0.117472239 -0.20896850
## X2          -0.002099436  0.11747224  1.839375248 -0.28335792
## X3           0.053913941 -0.20896850 -0.283357922  2.62764304
```

Otrzymane macierze są takie same.

Prawdziwą wartość macierzy kowariancji wektora β obliczmy podstawiając pod Σ macierz zadaną w poleceniu.

```
cov_beta = lapply(1:n, function(i) t(Xsplit[[i]])%*%solve(sigma)%*%Xsplit[[i]])
cov_beta = Reduce('+', cov_beta)
cov_beta = solve(cov_beta)
cov_beta
```

```
##             [,1]        [,2]        [,3]       [,4]
## [1,]  0.10963015 -0.07494681 -0.00273796  0.0670384
```

```
## [2,] -0.07494681  3.11660939  0.14870924 -0.2615358
## [3,] -0.00273796  0.14870924  2.28283766 -0.3551597
## [4,]  0.06703840 -0.26153584 -0.35515967  3.2657673
```

Norma supremum różnicy między prawdziwa macierzą kowariancji a jej estymatorem REML to:

```
norm(cov_beta - a1, type = "M")
```

```
## [1] 0.6381243
```

1.3.2 Macierz Σ

Przyjrzyjmy się teraz estymatorom parametrów ρ oraz γ. Ich wartości to

```
cov2cor(sigma_reml)[1,2]; sqrt(sigma_reml[1,1])
```

```
## [1] 0.3156537
```

```
## [1] 1.811032
```

Table 1: Własności parametrów.

	Wartość prawdziwa	Estymator	Wartość bezw. różnicy $\lvert\theta - \hat{\theta}\rvert$
Parametry beta			
β_0	0.0	0.0493	0.0493
β_1	3.0	3.6983	0.6983
β_2	3.0	3.4615	0.4615
β_3	0.0	2.1851	2.1851
Rho			
ρ	0.3	0.3157	0.0157
Gamma			
γ	2.0	1.8110	0.1890

Wartości estymatorów generalnie są zbliżone do prawdziwych wartości, ich wartości są jednak silnie zależne od próby.

Table 2: Norma supremum różnicy.

	$\hat{\beta}$	$\hat{\rho}$	$\hat{\gamma}$	$\hat{\Sigma}_{REML}$
sup-norm	2.185057	0.0156537	0.1889677	0.6381243

2 Zadanie 2

Wygenerujemy 500 replikacji wektora Y i skonstruujemy przy ich pomocy modele liniowe z których następnie wyznaczymy 500 replikacji wektora $\hat{\beta}$, $\hat{\rho}$ oraz $\hat{\gamma}$.

```
sim = function(Xsplit) {
  rep = 500; N = n*k
  beta_rep = matrix(nrow = rep, ncol = p); gamma_rep = numeric(length = rep)
  rho_rep = numeric(length = rep)

  for (i in 1:rep) {
```

```
    Y = lapply(Xsplit, function(X) rmvnorm(1, mean = X%*%beta, sigma = sigma))
    data_uni = lapply(1:n, function(i) cbind(t(Y[[i]]), rep(i, k), 1:k, Xsplit[[i]]))
    data_uni = do.call(rbind, data_uni)
    data_uni = data.frame(data_uni); colnames(data_uni) = c('y', 'id', 'T', paste0("X", 0:(p-1)))
    if(p == 4) {
      m = gls(y~. -id-T-X0, correlation = corCompSymm(form = ~1|id),
              weights = varIdent(form = ~1), method = "REML", data = data_uni) }
    else {
      m = gls(y~. -id-T-X0, correlation = corCompSymm(form = ~1|id),
              weights = varIdent(form = ~1), data = data_uni,
              control = glsControl(opt='optim')) }
    beta_rep[i,] = coef(m); covM = getVarCov(m)
    rho_rep[i] = cov2cor(covM)[1,2]; gamma_rep[i] = sqrt(covM[1,1])
  }
  colnames(beta_rep) = paste0("b", 0:(p-1))
  return(list(b = beta_rep, r = rho_rep, g = gamma_rep))
}
# Kowariancja asymptotyczna wektora beta
avar = lapply(1:n, function(i) t(Xsplit[[i]])%*%solve(sigma)%*%Xsplit[[i]])
avar = Reduce('+', avar); avar = solve(avar)
res_z2 = sim(Xsplit)
```

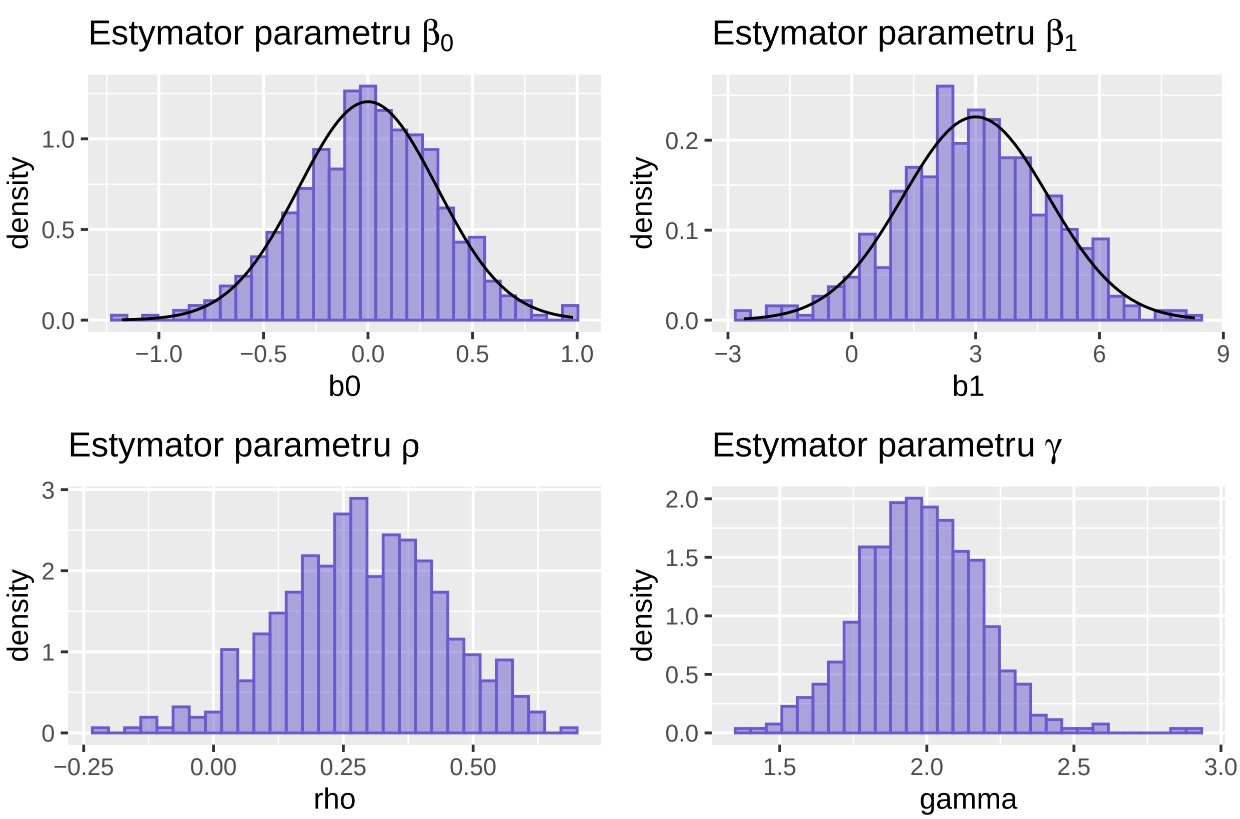

Figure 1: Histogramy dla $n = 20, k = 3, p = 4$.

Widzimy, ze histogramy są bliskie rozkładom asymptotycznym. Ponieważ rozmiar próby jest mały, to dane są dość mocno rozrzucone. Wartości są skoncentrowane wokół prawdziwych wartości, choć pojawiają się znaczące odchylenia.

3 Zadanie 3

Ponownie wykonamy symulacje z zadania 2 dla $n = 500$.

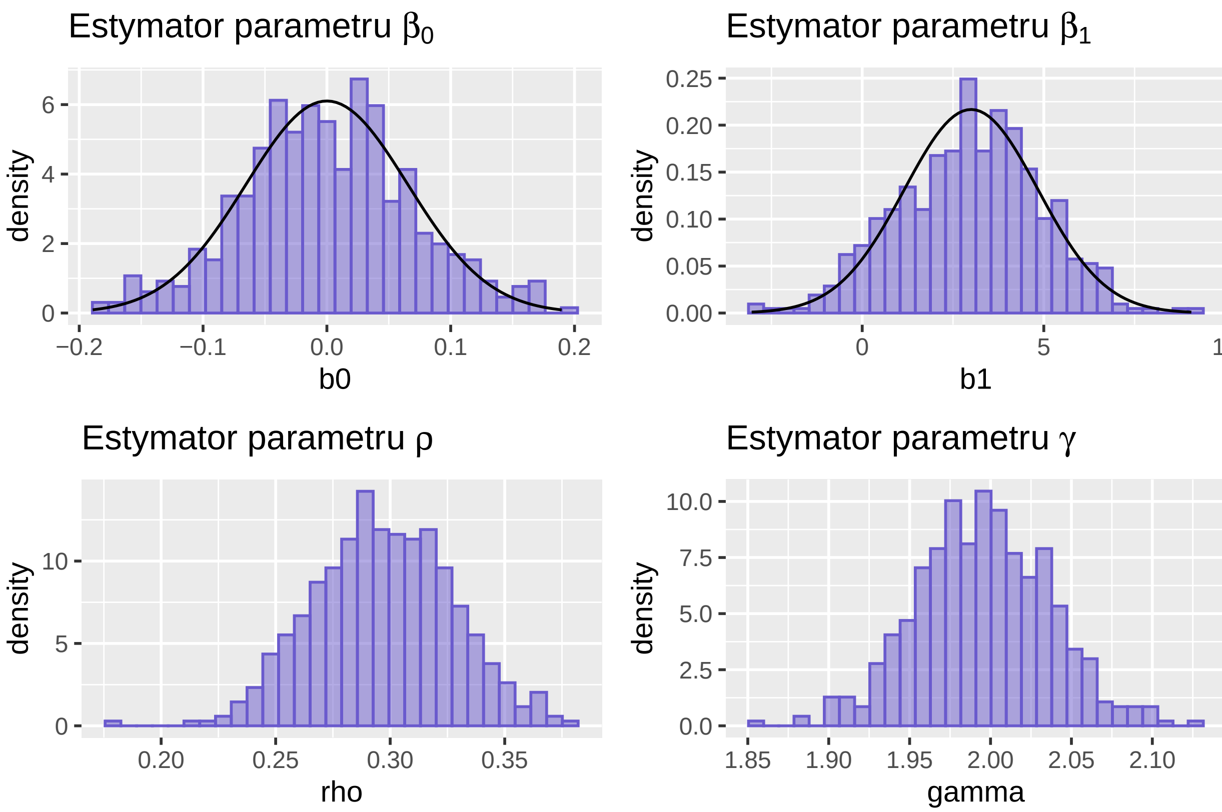

Figure 2: Histogramy dla $n = 500, k = 3, p = 4$.

Histogramy są bliskie rozkładom asymptotycznym. W porównaniu do przypadku gdy $n = 20$, wahania w danych są mniejsze i estymatory są wyznaczane z większą dokładnością.

4 Zadanie 4

Ponownie wykonamy symulacje z zadania 2 dla $k = 30$.

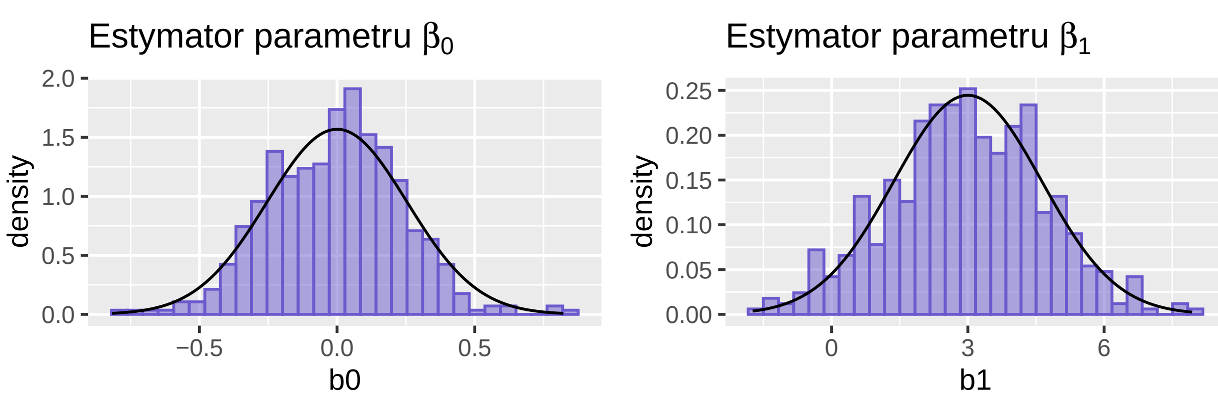

Figure 3: Histogramy $\hat{\beta}_0$ oraz $\hat{\beta}_1$, dla $n = 20, k = 30, p = 4$.

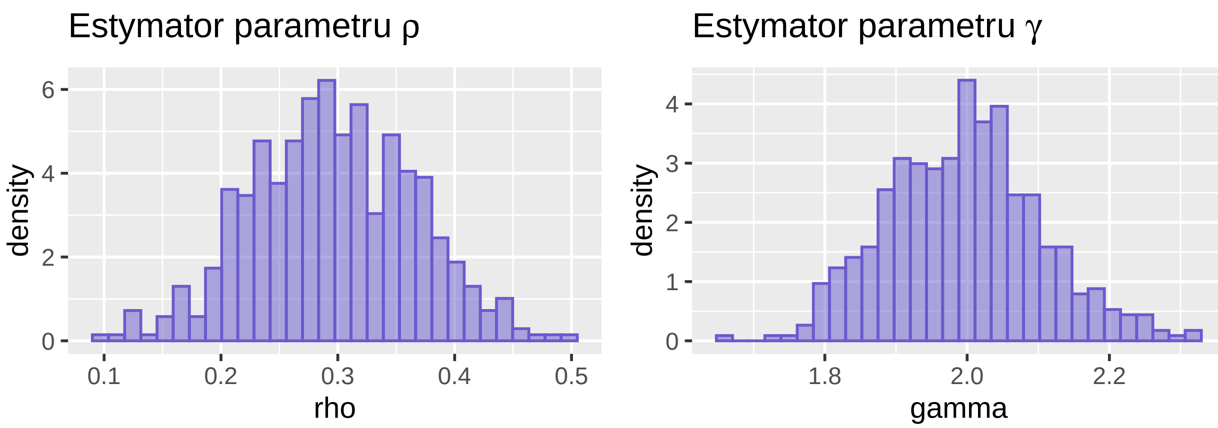

Figure 4: Histogramy $\hat{\rho}$ oraz $\hat{\gamma}$ dla $n = 20, k = 30, p = 4$.

Estymatory dla $k = 30$ są zbliżone do przypadku gdy $n = 20$, choć wartości skrajne są nieco mniejsze.

5 Zadanie 5

Tym razem zwiększymy wartość p do 40.

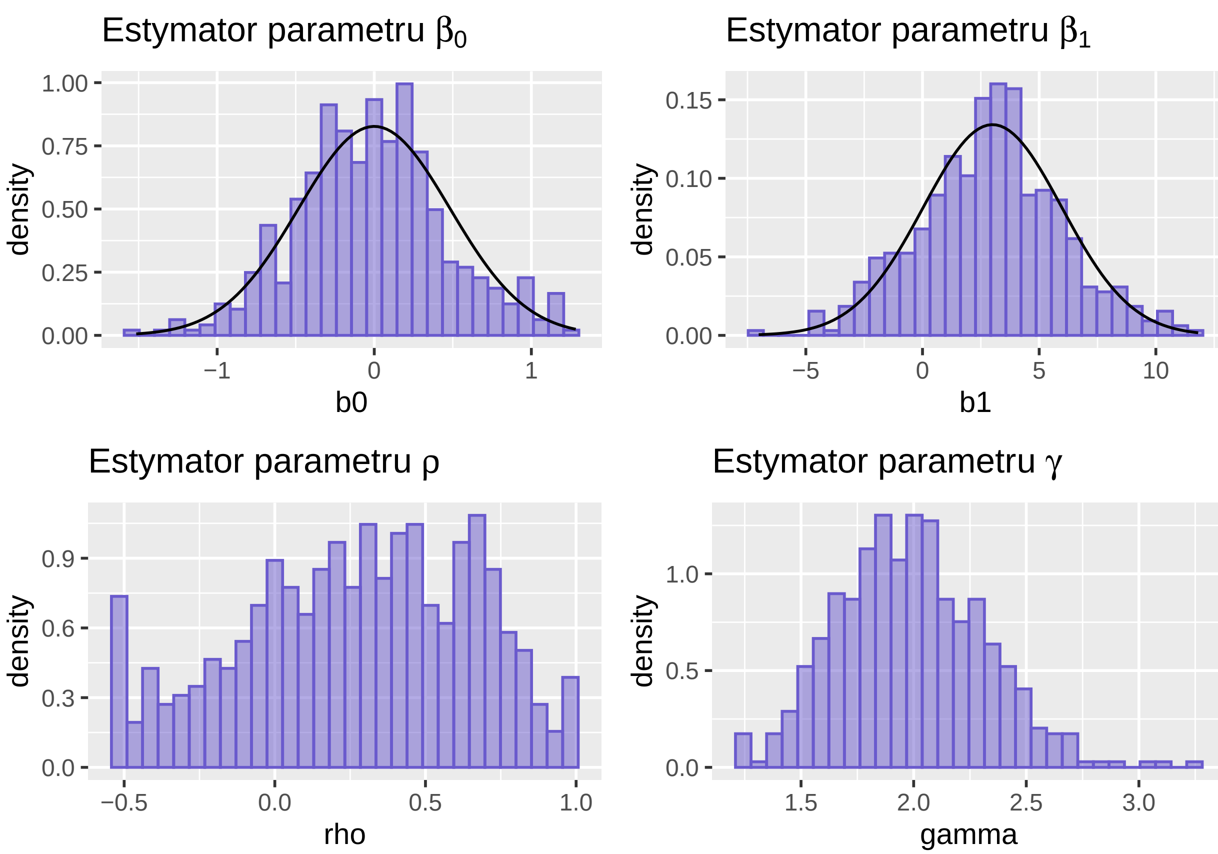

Figure 5: Histogramy dla $n = 20, k = 3, p = 40$.

Estymatory osiągają bardziej skrajne wartości niż w poprzednich zadaniach. Estymacja dla wielu przypadków jest mało dokładna.

6 Podsumowanie wyników z zadań 2-5

Table 3: Własności estymatora $\hat{\beta}$.

			Obciążenie($\hat{\beta}_i$) = $E(\hat{\beta}_i) - \beta_i$				$\|\|\hat{\beta}_i\|\|_{sup}$			
n	k	p	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$
20	3	4	0.002	-0.033	0.031	-0.240	1.177	8.310	7.583	6.138
500	3	4	0.000	-0.028	-0.031	0.046	0.190	9.063	8.058	5.770
20	30	4	-0.005	-0.053	-0.021	0.199	0.822	7.937	8.584	5.329
20	3	40	-0.012	-0.082	-0.014	0.245	1.515	11.814	13.054	13.346

Table 4: Wariancja i średnia estymatorów $\hat{\beta}_i$.

n	k	p	$var(\hat{\beta}_0)$	$var(\hat{\beta}_1)$	$var(\hat{\beta}_2)$	$var(\hat{\beta}_3)$	$E(\hat{\beta}_0)$	$E(\hat{\beta}_1)$	$E(\hat{\beta}_2)$	$E(\hat{\beta}_3)$
20	3	4	0.116	3.296	2.565	3.452	0.002	2.967	3.031	-0.240
500	3	4	0.005	3.826	3.590	3.302	0.000	2.972	2.969	0.046
20	30	4	0.057	2.918	2.889	2.801	-0.005	2.947	2.979	0.199
20	3	40	0.229	9.368	13.263	20.306	-0.012	2.918	2.986	0.245

Table 5: Własności macierzy $\hat{\Sigma}_{REML}$.

n	k	p	Obciążenie	Średnia	Wariancja
Rho					
20	3	4	-0.0143211	0.2856789	0.0236220
500	3	4	-0.0029085	0.2970915	0.0009127
20	30	4	-0.0041718	0.2958282	0.0048728
20	3	40	-0.0266906	0.2733094	0.1457461
Gamma					
20	3	4	-0.0223781	1.9776219	0.0412943
500	3	4	-0.0045628	1.9954372	0.0016911
20	30	4	-0.0047753	1.9952247	0.0115972
20	3	40	-0.0128289	1.9871711	0.1043919

Najgorzej zachowują się estymatory w przypadku $p = 40$. Większa ilość predyktorów wpływa na zwiększenie obciążenia estymatora jak i wariancji, zatem estymacja jest mniej dokładna. Osiągane wartości skrajne są znacznie większe niż dla pozostałych przypadków. Drugie największe obciążenie mają estymatory dla $n = 20$, co jest naturalne jako że mała ilość obiektów nie pozwala nam na dokładne wyestymowanie parametrów. Najlepsze wyniki są osiągane dla $n = 500$ oraz dla $k = 30$. Warto też zauważyć, że estymatory ρ oraz γ mają generalnie ujemne obciążenie, czyli estymator kowariancji REML raczej ściąga wartości do zera.

7 Zadanie 6

Powtórzymy zadanie 2, tym razem używając do estymacji macierzy kowariancji metody ML.

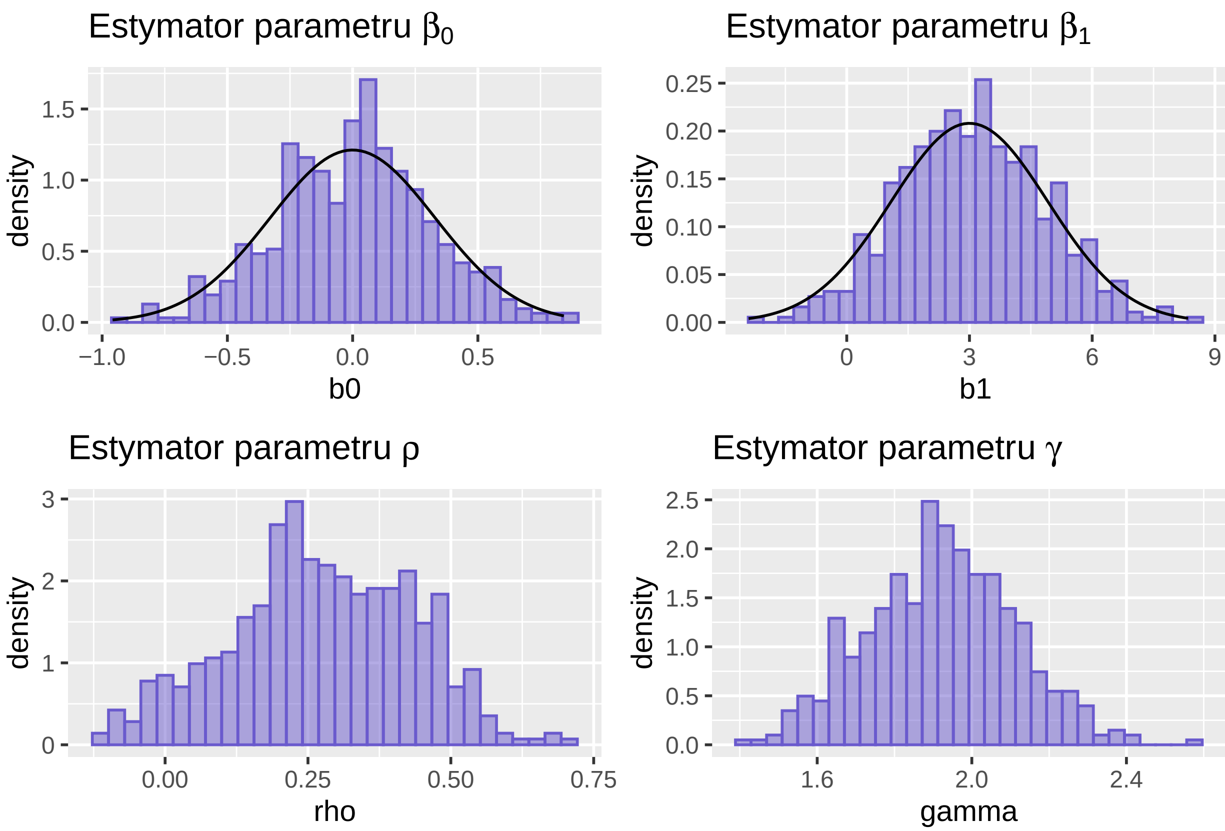

Figure 6: Histogramy dla $n = 20, k = 3, p = 4$, estymacja metodą ML.

Table 6: Własności $\hat{\Sigma}_{ML}$

	n	k	p	Obciążenie	Średnia
Rho					
	20	3	4	-0.0262871	0.2737129
Gamma					
	20	3	4	-0.0786789	1.9213211

Widzimy, że wartość bezwzględna obciążenia estymatorów jest większa niż w przypadku estymacji metodą REML. Obciążenie jest ujemne, a wartości średnie estymatorów są mniejsze od prawdziwych wartości parametrów. Potwierdza to teorię, iż estymator uzyskany metodą ML ściąga do zera silniej niż estymator REML.

Table 7: Własności estymatora $\hat{\beta}$, metoda ML.

	Obciążenie	Sup-norm	Średnia	Wariancja
$\hat{\beta}_0$	0.005	1.177	0.005	0.096
$\hat{\beta}_1$	0.080	8.310	3.080	3.167
$\hat{\beta}_2$	0.043	7.583	3.043	3.384
$\hat{\beta}_3$	0.007	6.138	0.007	3.249

Jeśli chodzi o estymatory $\hat{\beta}$, to nie widać znaczącej różnicy w porównaniu do wyników z poprzedniego zadania.