Zaawansowane modele liniowe - Lista 1

Klaudia Weigel

1 Analiza danych

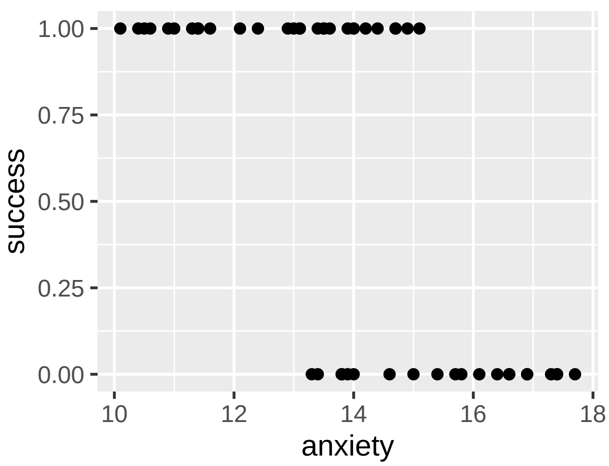
1.1 Zadanie 1

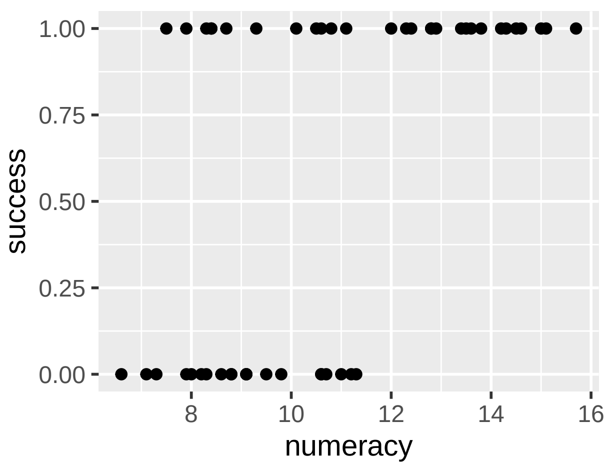
Mamy zbiór danych opisujący relację między prawdopodobieństwami przyjęcia na studia (success) a wynikami z testów rachunkowych (numeracy) i poziomem niepewności (anxiety).

```
data = read.table("lista_1.csv", sep = ",", header = TRUE)
data[1:5,]
```

```
##   X numeracy anxiety success
## 1 1      6.6    13.8       0
## 2 2      7.1    14.6       0
## 3 3      7.3    17.4       0
## 4 4      7.5    14.9       1
## 5 5      7.9    13.4       0
```

Spójrzmy również na wykresy sukcesu w zależności od zmiennej *anxiety* oraz zmiennej *numeracy*.

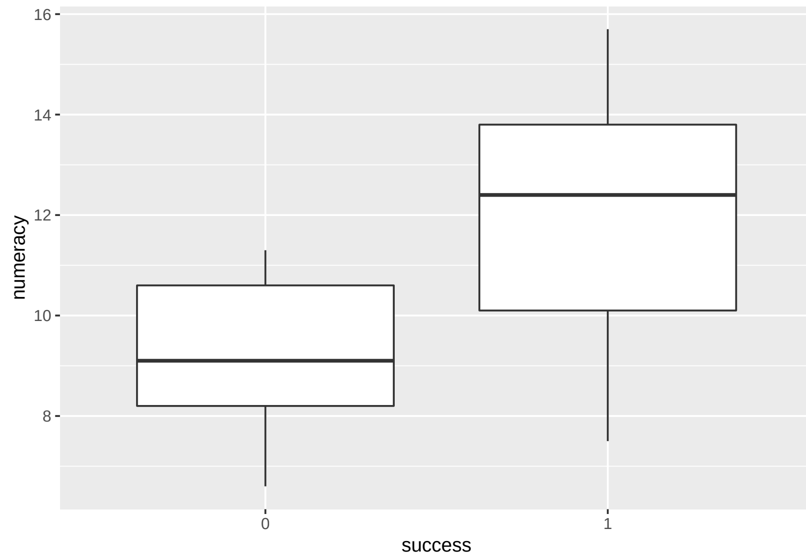




Z wykresów możemy podejrzewać, iż istnieje związek pomiędzy sukcesem a wartościami *numeracy* i *anxiety*.

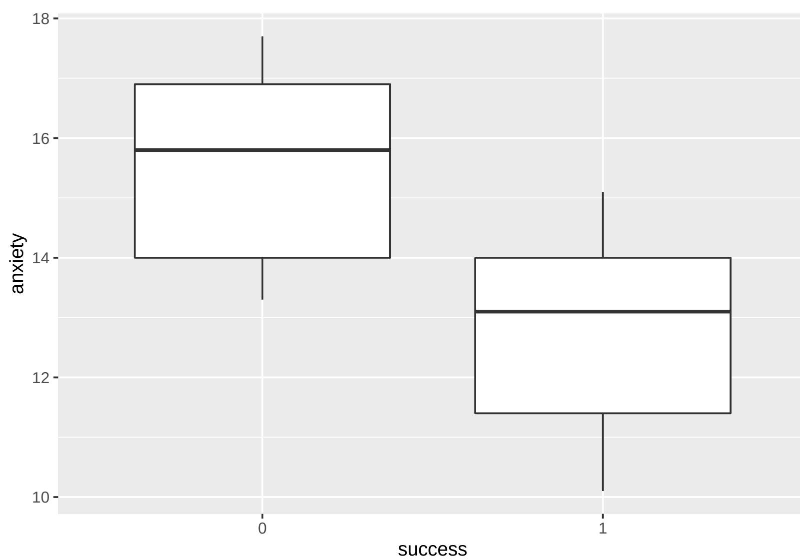
1.2 Zadanie 2

Narysujemy teraz boxplot dla zmiennej *numeracy* w rozbiciu na grupę przyjętych/nieprzyjętych osób.



Widzimy, że w grupie osób, które zostały przyjęte na studia mediana zmiennej *numeracy* jest wyższa niż w grupie nieprzyjętych. Połowa osób w grupie przyjętych osiąga wynik powyżej 12, gdzie w grupie nieprzyjętych taki wynik nie występuje. Sugeruje to, że dla ucznia, który osiągnął większy wynik *numeracy* prawdopodobieństwo przyjęcia na studia będzie większe.

1.3 Zadanie 3



Tym razem mediana dla *anxiety* jest niższa w grupie osób przyjętych. Możemy podejrzewać, że uczeń z mniejszym indeksem *anxiety* będzie miał większe szanse na przyjęcie.

1.4 Zadanie 4

Teoretyczny model regresji logistycznej z funkcją linkującą logit dla tego problemu to

$$\text{logit}(\mu_i) = \beta_0 + numeracy_i * \beta_1 + anxiety_i * \beta_2, \quad \text{logit} = \log \frac{\mu_i}{1-\mu_i}, \quad i = 1, \ldots, 50.$$

Do konstrukcji modelu wykorzystamy fukcję `glm`.

```r
reg = glm(success~numeracy + anxiety, data = data, family = "binomial")
```

Estymatory parametrów regresji

```r
reg$coefficients
```

```
## (Intercept)    numeracy     anxiety
##   14.238581    0.577352   -1.384069
```

Zatem nasz model to

$$\mathrm{logit}(\mu_i) = 14.239 + 0.577 * numeracy_i - 1.384 * anxiety_i.$$

Chcemy testować

$$H_{0i} : \beta_i = 0 \quad H_{1i} : \beta_i \neq 0.$$

Hipoteza testowa ma postać

$$T = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)} = \frac{\hat{\beta}_i}{\sqrt{J_{ii}^{-1}}}, \quad J = X'S(\beta)X, \ S = \mathrm{diag}(\hat{\mu}(1-\hat{\mu})).$$

Gdzie X to macierz planu. Statystyka T ma przy hipotezie zerowej asymptotycznie rozkład standardowy normalny. Test na poziomie istotności α odrzuca hipotezę zerową, gdy $T > |w_c| = \Phi^{-1}(1-\alpha/2)$.

```r
# p values of significance tests
summary(reg)[['coefficients']][,"Pr(>|z|)"]
```

```
## (Intercept)    numeracy     anxiety
## 0.036227472 0.019952273 0.003963398
```

Na poziomie istotności 0.05 wszystkie parametry są istotne.

Przewidywane p-stwo sukcesu u studenta, którego *anxiety*= 13, a *numeracy*=10.

```r
predict.glm(reg, newdata = data.frame(numeracy=c(10), anxiety=c(13)), type = "response")
```

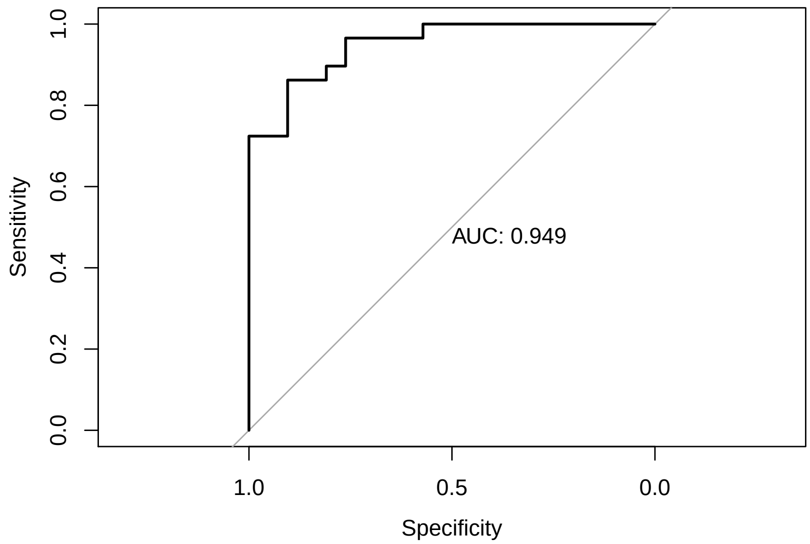
```
##         1
## 0.8827987
```

Prawdopodobieństwo, że taki student dostanie się na studia wynosi około 88%.

- Krzywa ROC

Statystyka nazywana czułością (sensitivity or true positive rate (TPR)) to proporcja prawdziwych odkryć (poprawna klasyfikacja) do wszystkich możliwych prawdziwych odkryć na zbiorze sukcesów (klasyfikacja = 1). Analogicznie zdefiniowana jest specyficzność z tym, że na zbiorze porażek (klasyfikacja = 0). Model przyjmujemy za dobry jeśli wartości obu tych statystyk są wysokie.

Krzywa ROC jest ilustracją wzajemnej zależności pomiędzy czułością i specyficznością predyktora w funkcji progu klasyfikacji s ($\hat{y}_i = 1$ jeśli $\mu_i(\hat{\beta}) \geq s$).




```
##
## Call:
## roc.formula(formula = data$success ~ fitted(reg), plot = TRUE,     print.auc = TRUE)
##
## Data: fitted(reg) in 21 controls (data$success 0) < 29 cases (data$success 1).
## Area under the curve: 0.9491
```

Nasz klasyfikator osiąga wartości bliskie górnemu lewemu rogowi (wysoka czułość, niskie 1-specyficzność). Model przyjmujemy za dobry.

1.5 Zadanie 5

Zkonstruujemy teraz modele regresji dla różnych funkcji linkujących.

1.5.1 Probit

Funkcja probit jest równa kwantylowi standardowaego rozkładu normalnego

$$\text{probit}(\mu_i) = \Phi^{-1}(\mu_i).$$

```
# probit
reg_prob = glm(success~numeracy + anxiety, data = data, family = binomial("probit"))
reg_prob$coefficients
```

```
## (Intercept)    numeracy     anxiety
##   8.2572551   0.3371040  -0.8038678
```

Model regresji probit to

$$\Phi^{-1}(\mu_i) = 8.257 + 0.337 * numeracy_i - 0.804 * anxiety_i.$$

```
# p values of significance tests
summary(reg_prob)[['coefficients']][,"Pr(>|z|)"]
```

```
## (Intercept)    numeracy     anxiety
## 0.024661662 0.013737795 0.001417764
```

Dla poziomu istotności 0.05 wszystkie parametry są istotne.

```r
predict.glm(reg_prob, newdata = data.frame(numeracy=c(10), anxiety=c(13)), type = "response")
```

```
##         1
## 0.8806045
```

Prawdopodobieństwo, że taki student dostanie się na studia wynosi około 88%, podobnie jak w przypadku regresjii logistycznej.

1.5.2 Cauchit

Funkcja cauchit jest równa kwantylowi standardowaego rozkładu Cauchy'ego

$$\text{cauchit}(\mu_i) = F^{-1}(\mu_i).$$

```r
#cauchit
reg_cauch = glm(success~numeracy + anxiety, data = data, family = binomial("cauchit"))
reg_cauch$coefficients
```

```
## (Intercept)    numeracy     anxiety
##  18.3829651   0.7322739  -1.7740788
```

Model regresji cauchit to

$$F^{-1}(\mu_i) = 18.383 + 0.732 * numeracy_i - 1.774 * anxiety_i.$$

```r
# p values of significance tests
summary(reg_cauch)[['coefficients']][,"Pr(>|z|)"]
```

```
## (Intercept)    numeracy     anxiety
##  0.13501166  0.12242822  0.07353505
```

Dla poziomu istotności 0.05 żaden z parametrów nie jest istotny.

```r
predict.glm(reg_cauch, newdata = data.frame(numeracy=c(10), anxiety=c(13)), type = "response")
```

```
##         1
## 0.8848509
```

W tym przypadku prawdopodobieństwo również wynosi około 88%.

1.5.3 Cloglog

Funkcja cloglog (complementary log-log function):

$$\text{cloglog}(\mu_i) = \log(-\log(1 - \mu_i)).$$

```r
#cloglog
reg_cll = glm(success~numeracy + anxiety, data = data, family = binomial("cloglog"))
reg_cll$coefficients
```

```
## (Intercept)    numeracy     anxiety
##   9.0005810   0.4024250  -0.9389748
```

Model regresji cloglog to

$$F^{-1}(\mu_i) = 9 + 0.402 * numeracy - 0.939 * anxiety.$$

```
# p values of significance tests
summary(reg_cll)[['coefficients']][,"Pr(>|z|)"]
```

```
## (Intercept)    numeracy     anxiety
## 0.053039856 0.008192355 0.004702357
```

Dla poziomu istotności 0.05 istotne są parametry z wyjątkiem interceptu.

```
predict.glm(reg_cll, newdata = data.frame(numeracy=c(10), anxiety=c(13)), type = "response")
```

```
##         1
## 0.8963072
```

Prawdopodobieństwo wynosi około 89%.

1.5.4 Najlpesze dopasowanie

Ponieważ wszystkie modele mają taką samą liczbę parametrów, do ich porównania mozemy wykorzystać statystykę Deviance. Wybieramy model z najmniejszą wartością tej statystyki.

```
reg$deviance; reg_prob$deviance; reg_cauch$deviance; reg_cll$deviance
```

```
## [1] 28.28562
```

```
## [1] 27.85445
```

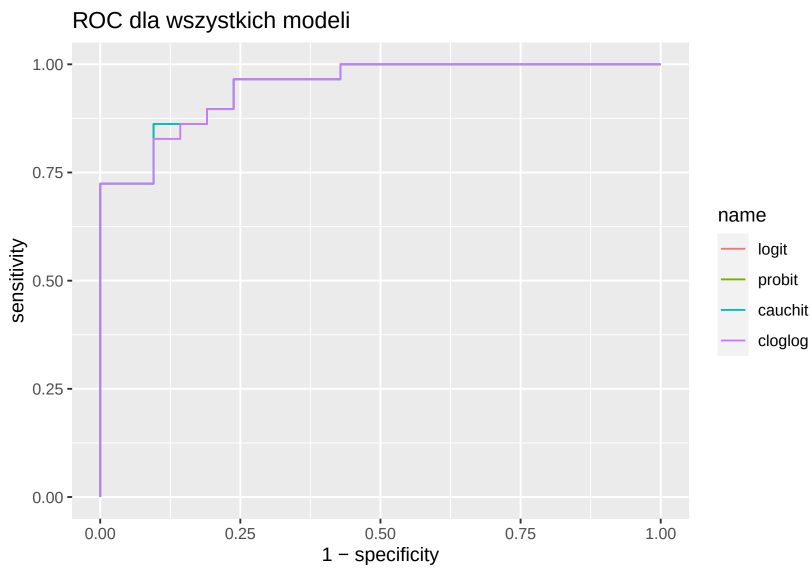
```
## [1] 31.11479
```

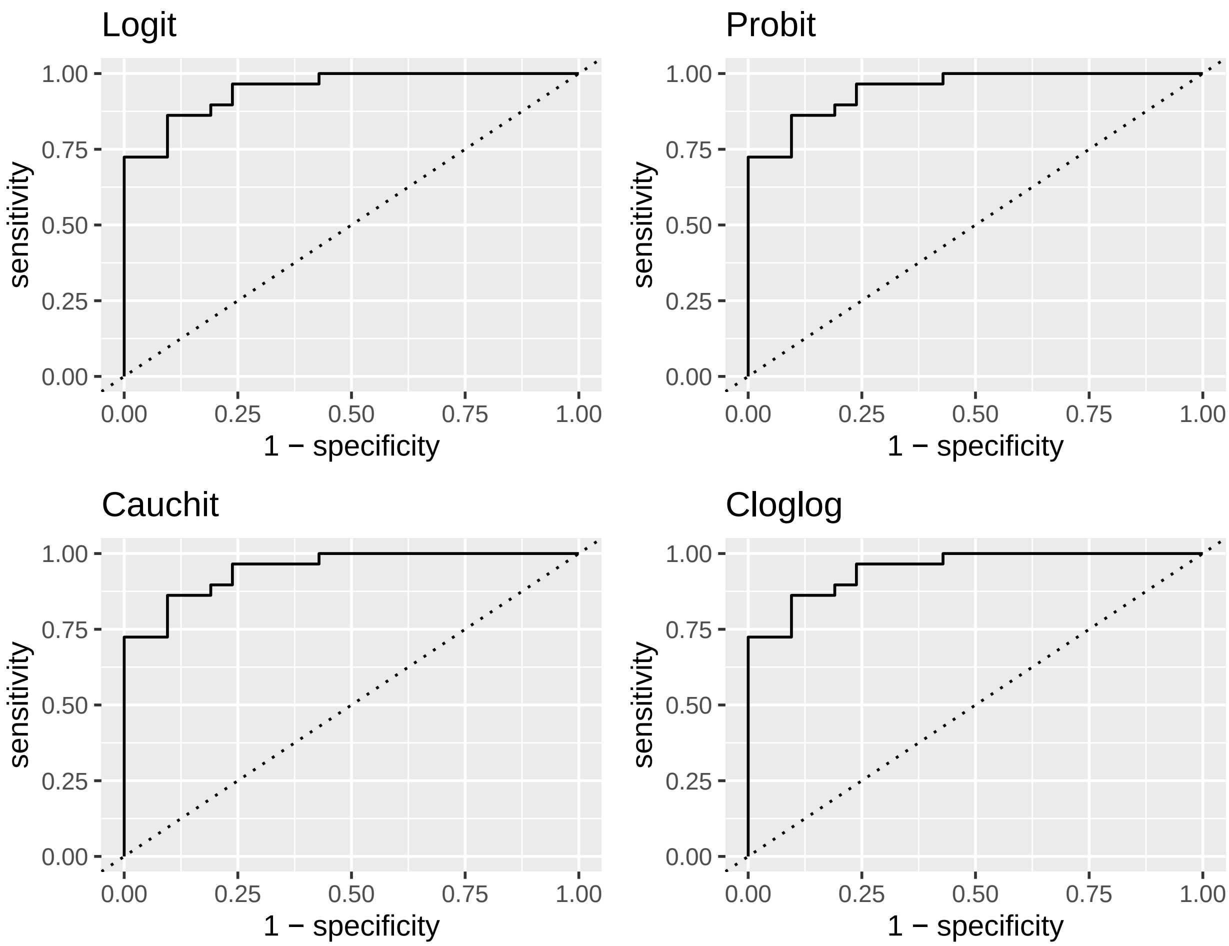
```
## [1] 27.99973
```

Najlepiej dopasowana zdaje się być funckja probit, następnie cauchit, logit i cloglog.

1.5.5 Krzywe ROC dla różnych modeli

```
rocobj1 = roc(data$success~fitted(reg))
rocobj2 = roc(data$success~fitted(reg_prob))
rocobj3 = roc(data$success~fitted(reg_cauch))
rocobj4 = roc(data$success~fitted(reg_cll))
```





Widzimy, że tylko krzywa ROC dla funkcji cloglog minimalnie różni się od pozostałych.

1.6 Zadanie 6

R tym zadaniu skupimy się na modelu z funkcja linkującą logit.

- Estymacja macierzy kowariancji

Macierz kowariancji wektora $\hat{\beta}$

$$\mathrm{Cov}(\hat{\beta}) = (X' S(\beta) X)^{-1}, \quad S = \mathrm{diag}(\hat{\mu}(1-\hat{\mu})).$$

```
prob = predict.glm(reg, data, type = "response")
S = diag(prob*(1-prob)) # nrow(data) by nrow(data)
X = as.matrix(cbind(1, data$numeracy, data$anxiety))
covM = solve(t(X)%*%S%*%X)
covM
```

```
##           [,1]        [,2]        [,3]
## [1,] 46.229542 -0.24821804 -3.06203011
## [2,] -0.248218  0.06155249 -0.02382592
## [3,] -3.062030 -0.02382592  0.23084660
```

```
vcov(reg)
```

```
##              (Intercept)    numeracy     anxiety
```

```
## (Intercept)  46.2198633 -0.24830459 -3.06128454
## numeracy     -0.2483046  0.06154567 -0.02381546
## anxiety      -3.0612845 -0.02381546  0.23078671
```

```
summary(reg)[['coefficients']][,"Std. Error"]^2
```

```
## (Intercept)    numeracy     anxiety
## 46.21986333  0.06154567  0.23078671
```

Obliczony ręcznie estymator macierzy kowariancji jak i macierz kowariancji zwracana przez R sa bardzo podobne.

- Test istostności obu zmiennych jednocześnie

Testujemy

$$H_0 : \beta_1 = \beta_2 = 0 \quad H_1 : \beta_1 \neq 0 \text{ lub } \beta_2 \neq 0.$$

Statystyka Deviance jest zdefiniowana

$$D(M(\hat{\beta}^{(r)})) = 2[\text{logLik}(\hat{\beta}^{(s)}) - \text{logLik}(\hat{\beta}^{(r)})],$$

gdzie przez $\text{logLik}(\hat{\beta}^{(s)})$ oznaczamy funkcję log-wiarogodności modelu saturowanego (liczba parametrów = liczba obserwacji), a przez $\text{logLik}(\hat{\beta}^{(r)})$ fukncję log-wiarogodności modelu zredukowanego (liczba parametrów < liczba obserwacji).

Oznaczmy przez M_1 model odpowiadający hipotezie zerowej (pewne współczynniki są równe zeru) oraz przez M_2 model związany z alternatywą (brak warunków). Statystyka:

$$\chi^2 = D(M_1) - D(M_2)$$

ma asyptotycznie rozkład χ^2, z liczbą stopni swobody równą liczbie zerowanych parametrów. Hipotezę zerową odrzucamy dla dużych wartości statystyki testowej, dokładniej gdy $\chi^2 > F^{-1}(1-\alpha)$, gdzie $F^{-1}(1-\alpha)$ jest kwantylem rzędu α z rozkładu χ^2 z liczbą stopni swobody jak wyżej.

Ponieważ interesuje nas model pełny i model tylko z interceptem, możemy skorzystać z wartości `null deviance` (deviance dla modelu tylko z β_0) oraz `residual deviance` (deviance dla modelu pełnego), zawartych w `summary`. Statystyka pochodzi z rozkładu χ^2 z 2 stopniami swobody.

```
T = reg$null.deviance - reg$deviance
T
```

```
## [1] 39.74358
```

```
1 - pchisq(T, df=2)
```

```
## [1] 2.343106e-09
```

P-wartość jest bardzo bliska zeru zatem na poziomie istotności 0.05 odrzucamy hipotezę zerową i stwierdzamy, że przynajmniej jeden parametr regresji jest różny od zera.

- Testowanie dopasowania modelu do danych

Chcemy teraz przetestować czy nasz model jest dobrze dopasowany do danych:

$$H_0 : \text{dane pochodzą z modelu} \quad H_1 : \text{dane nie pochodzą z modelu.}$$

Kiedy mamy do czynienia z danymi grupowanymi i kategorycznymi możemy w tym celu posłużyć się statystyka Deviance. Niestety dane, którymi dysponujemy są ciągłe, zatem wykorzystanie statystyki Deviance jest niemożliwe. Użyjemy testu Hosmera-Lemeshowa. Hipoteza zerowa tego testu mówi, że prawdopodobieństwa estymowane przez model odpowiadają prawdopodobieństwom rzeczywistym.

```
generalhoslem::logitgof(data$success, fitted(reg))
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  data$success, fitted(reg)
## X-squared = 2.3013, df = 8, p-value = 0.9704
```

P-wartość jest większa niż 0.05. Zatem na tym poziomie istotności nie mamy podstaw, aby twierdzić że nasz model jest źle dopasowany.

- Parametr epsilon

W przeciwieństwie do regresjii liniowej w regresji logistycznej nie jesteśmy w stanie znaleźć dokładnej wartości estymatora $\hat{\beta}$, przez co szukamy oszacowania. Funkcja `glm` do znalezienia tego oszacowania używa iteracyjnej metody *iteratively reweighted least squares*. Parametr ϵ w funkcji `glm` kontroluje kiedy nasze oszacowanie jest wystarczająco dobre i możemy zatrzymać iteracje. Z dokumantacji `glm` mamy, że algorytm zbiegł gdy:

$$\frac{|dev - dev_{old}|}{|dev| + 0.1} < \epsilon.$$

Domyślnie jest ustawiony na 10^{-8}.

Table 1: Porównanie dla różnych ϵ

eps	iter	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$
10^{-1}	3	12.89008	0.5375846	-1.263953
10^{-2}	4	14.09247	0.5735451	-1.371306
10^{-3}	5	14.23683	0.5773136	-1.383920
10^{-6}	6	14.23858	0.5773520	-1.384069
10^{-8}	6	14.23858	0.5773520	-1.384069

2 Symulacje

2.1 Zadanie 1

Mamy model

$$\mu = X\beta, \quad X = \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & X_{n3} \end{pmatrix}, X_{ij} \sim N(0, 1/400), \beta = \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}.$$

Wyznaczymy macierz informacji Fishera dla tego modelu w punkcie β i asymptotyczną macierz kowariancji estymatorów najwiekszej wiarogodnosci.

```
n = 400
p=3
X = matrix(rnorm(n*p, 0, 1/20), nrow = n, ncol = p)
beta = c(3,3,3)
prob = as.vector(exp(X%*%beta)/(1 + exp(X%*%beta)))
S = diag(prob*(1-prob))
# macierz informacji Fishera
J = t(X)%*%S%*%X
# macierz kowariancji
covM = solve(J)
```

Macierz informacji Fishera to

```r
J
```

```
##              [,1]         [,2]        [,3]
## [1,]  0.252164377  0.008715272 -0.01491197
## [2,]  0.008715272  0.277393450 -0.02277031
## [3,] -0.014911971 -0.022770313  0.27705509
```

Macierz kowariancji to odwrotności J

```r
covM
```

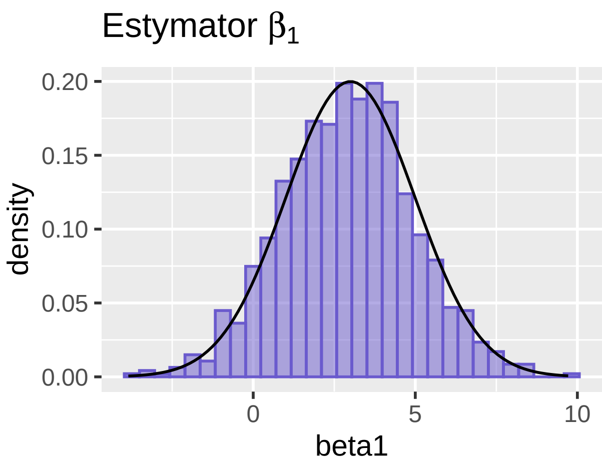
```
##            [,1]       [,2]      [,3]
## [1,]  3.9815547 -0.1082333 0.2054043
## [2,] -0.1082333  3.6324160 0.2927117
## [3,]  0.2054043  0.2927117 3.6445030
```

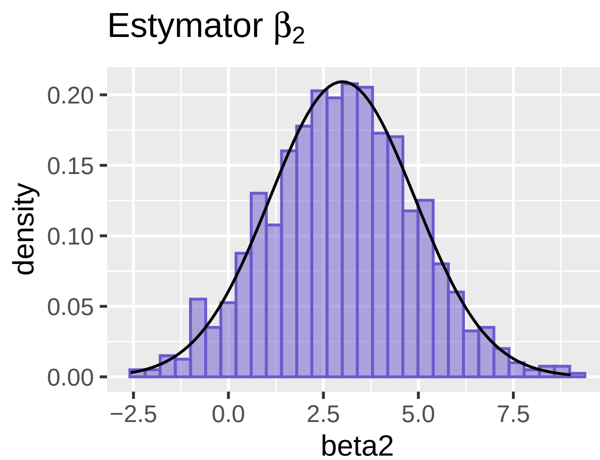
Wygenerujemy teraz 1000 replikacji wektora odpowiedzi zgodnie z powyższym modelem i na podstawie każdej replikacji wyznaczymy estymator wektora β.

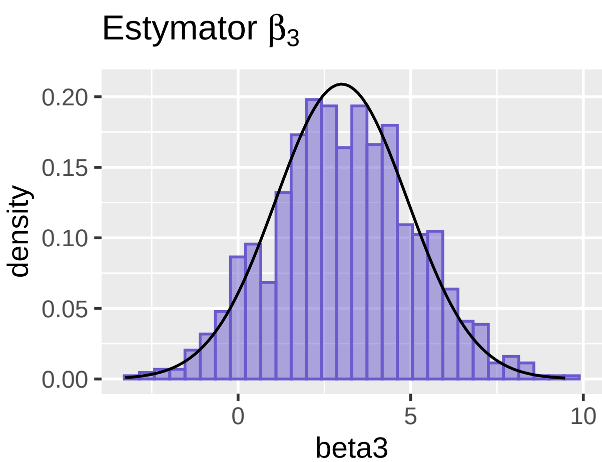
- Histogramy estymatorów i ich rozkład asymptotyczny dla $n = 400$.

```r
experiment = function(X, prob) {
  Y= rbinom(n, 1, prob = prob)
  reg = glm(Y~X-1, family = 'binomial')
  coefs = reg$coefficients
  return(c(reg$coefficients[1], reg$coefficients[2], reg$coefficients[3]))
}

res = replicate(1000, experiment(X, prob))
res = t(res)
colnames(res) = c("beta1", "beta2", "beta3")
```







Widzimy, że histogramy są bliskie rozkładom asymptotycznym w przypadku gdy $n = 400$.

- Obciążenie estymatorów

Obciążonie estymatora $\hat{\beta}_i$ parametru β_i to

$$\text{bias}(\hat{\beta}_i) = \mathbb{E}(\hat{\beta}_i) - \beta_i.$$

```
get_bias = function(estimate, truth) {
  mean(estimate) - truth
}
biases = apply(res, 2, get_bias, truth=3)
biases
```

```
##        beta1        beta2        beta3
## -0.093738826 -0.002762434  0.076681040
```

- Estymacja macierzy kowariancji wektora estymatorów

Próbkowa macierz kowariancji:

```
cov(res)
```

```
##            beta1      beta2     beta3
## beta1 4.05883802 0.07737892 0.3369331
## beta2 0.07737892 3.72599763 0.2076382
## beta3 0.33693313 0.20763818 4.2249538
```

Asymptotyczna macierz kowariancji:

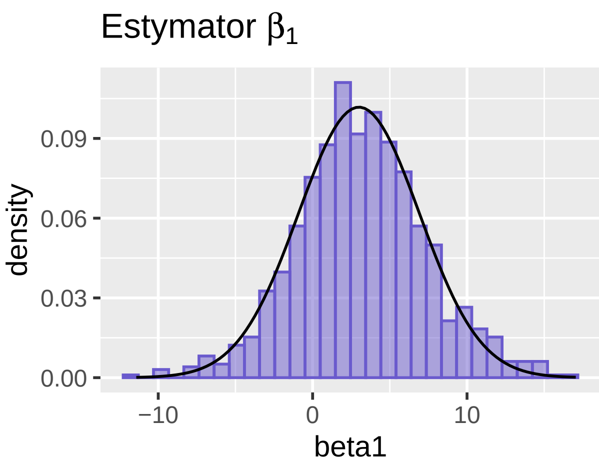
```
covM
```

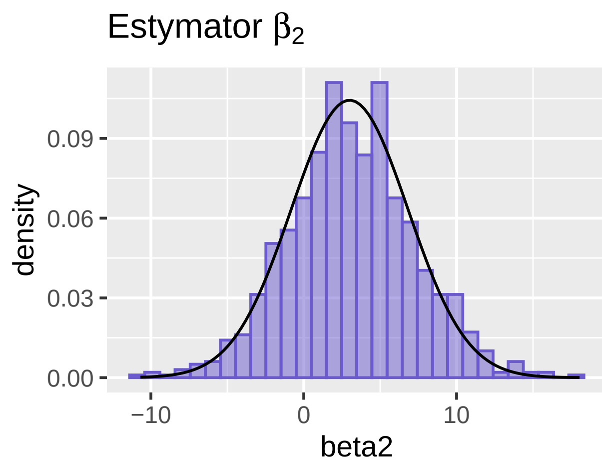
```
##             [,1]       [,2]      [,3]
## [1,]  3.9815547 -0.1082333 0.2054043
## [2,] -0.1082333  3.6324160 0.2927117
## [3,]  0.2054043  0.2927117 3.6445030
```

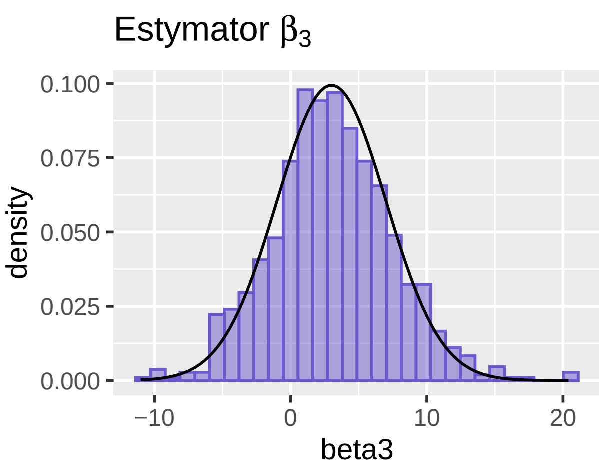
Estymowana macierz kowariancji, jak i asymptotyczna macierz kowariancji są do siebie zbliżone.

2.2 Zadanie 2

Powtórzymy teraz doświadczenie z zadania 1 dla $n = 100$







Zauważmy że w porównaniu do poprzedniego zadania, gdy $n = 400$, dane są dużo bardziej rozrzucone. Estymatory przyjmują wartości od około -10 do 20, gdzie w poprzednim zadaniu wahania były w obrębie od -5 do 10.

```
biases2 = apply(res2, 2, get_bias, truth=3)
biases2
```

```
##      beta1      beta2      beta3
## 0.17553005 0.13573548 0.08531185
```

```
cov(res2)
```

```
##            beta1      beta2      beta3
## beta1 17.5450641  0.2405458  0.6968907
## beta2  0.2405458 17.0071683  2.8089129
## beta3  0.6968907  2.8089129 19.6571816
```

```
covM2
```

```
##            [,1]       [,2]       [,3]
## [1,] 15.3666472  0.4635381  0.9175608
## [2,]  0.4635381 14.6091126  2.2307134
## [3,]  0.9175608  2.2307134 16.0953651
```

Elementy obu macierzy są porównywalne, lecz różnice pomiędzy nimi są większe niż w poprzednim punkcie. Słabsza zbieżność do rozkładu asymptotycznego, w przypadku małej ilości obserwacji.

2.3 Zadanie 3

Sprawdzimy teraz jak na estymatory oddziaływuje korelacja pomiędzy predyktorami. Wiersze macierzy planu generujemy z rozkładu normalnego $N(0, \Sigma)$, gdzie

$$\Sigma = \frac{1}{n}S, \ S_{ii} = 1, \ S_{i,j} = 0.3 \text{ dla } i \neq j.$$

```
library(mvtnorm)
n=400
covX3 = matrix(0.3, nrow = 3, ncol = 3)
diag(covX3) = 1
covX3 = (1/n)*covX3
X3 = rmvnorm(n, mean = rep(0, 3), sigma = covX3)
```

Teoretyczna macierz kowariancji

```
covX3
```

```
##         [,1]    [,2]    [,3]
## [1,] 0.00250 0.00075 0.00075
## [2,] 0.00075 0.00250 0.00075
## [3,] 0.00075 0.00075 0.00250
```

Zobaczmy, czy rzeczywiście otrzymaliśmy macierz zgodna z założeniami

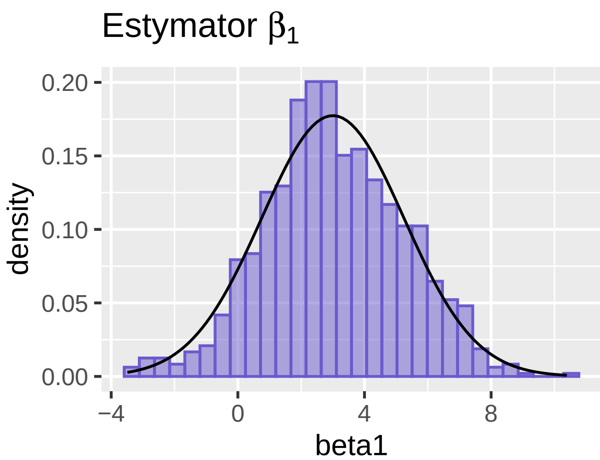
```
cov(X3)
```

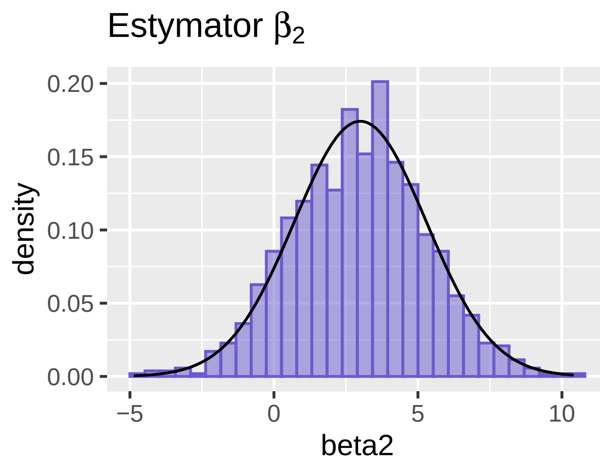
```
##              [,1]         [,2]         [,3]
## [1,] 0.0023390840 0.0006046897 0.0006921742
## [2,] 0.0006046897 0.0022303125 0.0006156311
## [3,] 0.0006921742 0.0006156311 0.0022305371
```

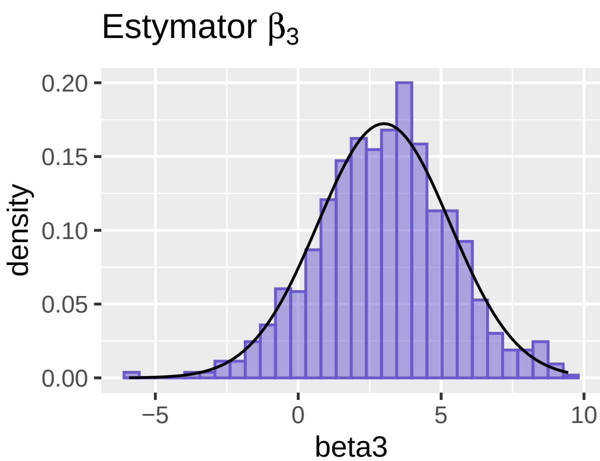
```
cov2cor(cov(X3))
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.2647444 0.3030315
## [2,] 0.2647444 1.0000000 0.2760152
## [3,] 0.3030315 0.2760152 1.0000000
```

Predyktory są skorelowane.








```
biases3 = apply(res3, 2, get_bias, truth=3)
biases3
```

```
##       beta1       beta2       beta3
##  0.05310544 -0.06012733  0.03074645
```

```
cov(res3)
```

```
##           beta1     beta2     beta3
## beta1  4.804437 -1.023677 -1.037418
## beta2 -1.023677  5.363034 -1.069060
## beta3 -1.037418 -1.069060  5.379508
```

```
covM3
```

```
##            [,1]       [,2]      [,3]
## [1,]  5.0583017 -0.9556393 -1.232506
## [2,] -0.9556393  5.2412546 -1.071426
## [3,] -1.2325062 -1.0714262  5.359215
```

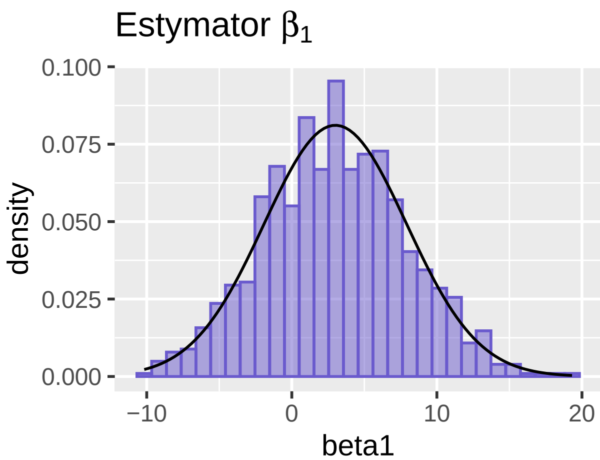
2.3.1 Większa korelacja

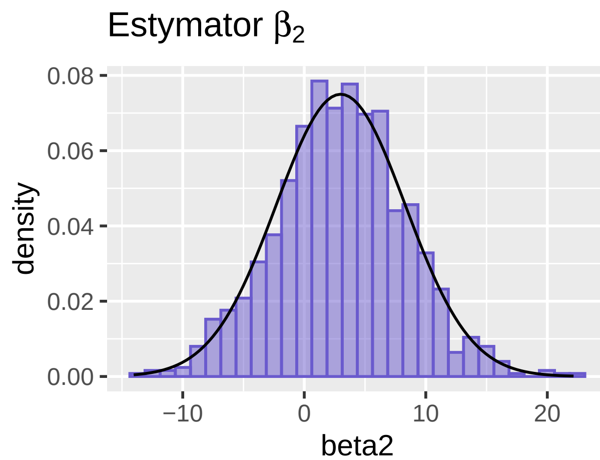
Poglądowo zobaczmy co się dzieje, jeśli predyktory są bardzo mocno skorelowane (na poziomie 0.9).

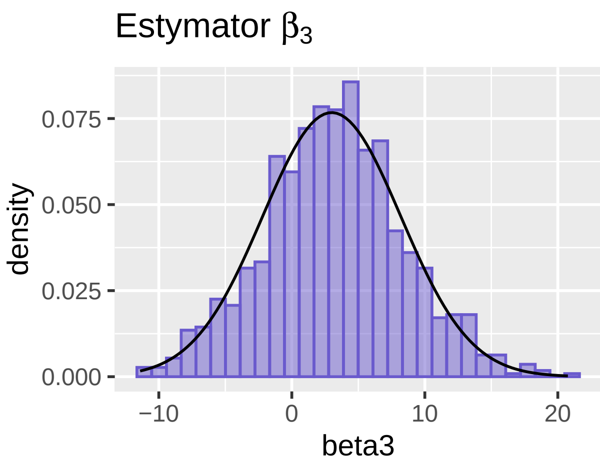
```
cov2cor(cov(X3_2))
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.8986131 0.8954491
```

```
## [2,] 0.8986131 1.0000000 0.8995552
## [3,] 0.8954491 0.8995552 1.0000000
```







Rozrzut danych jest duży od ok -20 do 20, w wielu przypadkach estymatory są wyznaczane bardzo niedokładnie.

```r
biases3_2 = apply(res3_2, 2, get_bias, truth=3)
biases3_2
```

```
##       beta1       beta2       beta3
## -0.01367519  0.01981552  0.17007467
```

```r
cov(res3_2)
```

```
##           beta1     beta2     beta3
## beta1  24.10047 -12.65717 -11.54493
## beta2 -12.65717  28.34533 -13.26667
## beta3 -11.54493 -13.26667  27.48362
```

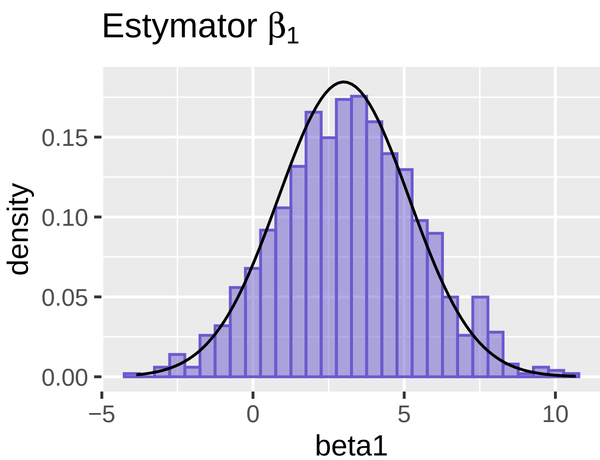
```r
covM3_2
```

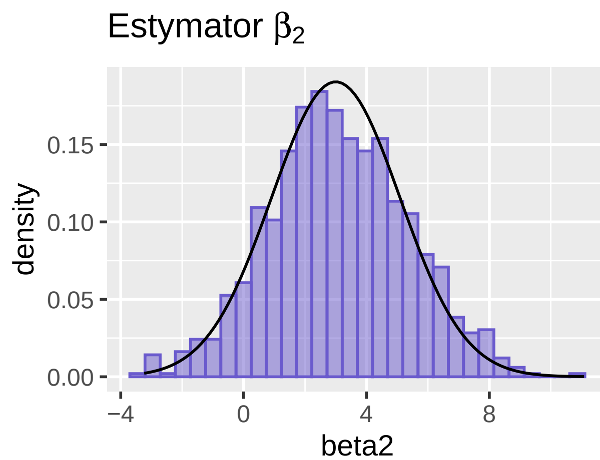
```
##           [,1]      [,2]      [,3]
## [1,]  24.19237 -12.39173 -11.53193
## [2,] -12.39173  28.29677 -13.45376
## [3,] -11.53193 -13.45376  27.03067
```

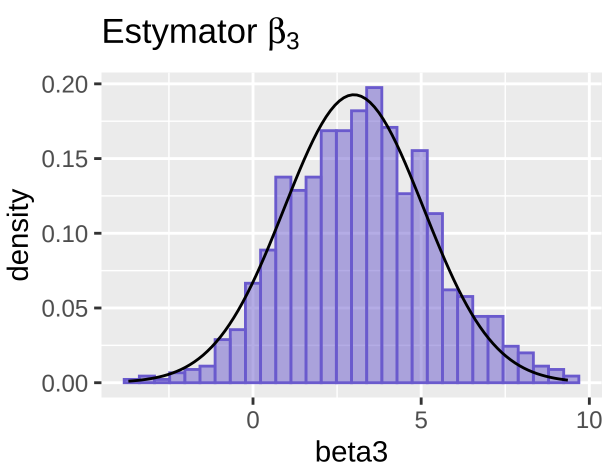
Duża wariancja, dane mocno różnią się od średniej.

2.4 Zadanie 4

Powtórzymy zadanie 1 w przypadku gdy liczba regresorów jest równa 20 ($p = 20$).








```r
biases4 = apply(res4, 2, get_bias, truth=3)
biases4
```

```
##     beta1     beta2     beta3
## 0.1766511 0.1065596 0.2230217
```

Średnie obciążenie jest więsze, niż w poprzednich punktach.

```r
cov(res4)
```

```
##           beta1      beta2      beta3
## beta1 5.4059891  0.3804727  0.2019445
## beta2 0.3804727  5.1234424 -0.2430808
## beta3 0.2019445 -0.2430808  4.6658058
```

```r
covM4[1:3, 1:3]
```

```
##           [,1]       [,2]       [,3]
## [1,] 4.6735232 0.19478039 0.31794559
## [2,] 0.1947804 4.38556186 0.00862077
## [3,] 0.3179456 0.00862077 4.28532328
```

2.5 Zadanie 5

Table 2: Porównanie modeli, średnia i obciążenie.

	mean($\hat{\beta}_1$)	mean($\hat{\beta}_2$)	mean($\hat{\beta}_3$)	bias($\hat{\beta}_1$)	bias($\hat{\beta}_2$)	bias($\hat{\beta}_3$)
M1	2.906261	2.997238	3.076681	-0.0937388	-0.0027624	0.0766810
M2	3.175530	3.135736	3.085312	0.1755301	0.1357355	0.0853118
M3	3.053105	2.939873	3.030746	0.0531054	-0.0601273	0.0307465
M4	3.176651	3.106560	3.223022	0.1766511	0.1065596	0.2230217

Table 3: Porównanie modeli, wariancja.

	var($\hat{\beta}_1$)	var($\hat{\beta}_2$)	var($\hat{\beta}_3$)
M1	4.058838	3.725998	4.224954
M2	17.545064	17.007168	19.657182
M3	4.804437	5.363034	5.379508
M4	5.405989	5.123442	4.665806

Największą wariancję ma model z zadania 2, co wydaje sie naturalne, jako że już wcześniej zaobserwowaliśmy rozrzut danych większy niż w pozostałych przypadkach. Pozostałe modele dają podobne wyniki. Najlepiej dane estymuje model z zadania 1, wariancje estymatorów są najmniejsze, podobnie obciążenia. Dla modelu z korelacją między predyktorami (M3) wariancja jest większa od tej w modelu 1. Większa ilość predyktorów wpływa na zwiększenie obciążenia estymatora jak i wariancji.