

Laboratoria 9 i 10

Paweł Matłowski

18 05 2021

Zadanie nr 1

W tym zadaniu będziemy analizować dane na temat stanu pacjenta (zmienna objaśniająca *stan*) i o pewnych parametrach charakteryzujących pracę jego serca (zmienne objaśniane).

(a)

Podzielimy losowo zbiór danych w proporcji 70 : 30 na zbiór treningowy i zbiór testowy, tak aby frakcja pacjentów ze zmienną *stan* = 1 w zbiorze treningowym niewiele się różniła od frakcji takich pacjentów w zbiorze testowym.

```
smp_size <- floor(0.7 * nrow(logistyczna))
set.seed(420)
train_ind <- sample(seq_len(nrow(logistyczna)), size = smp_size)

train_set <- logistyczna[train_ind,]
test_set  <- logistyczna[-train_ind,]

train_fraction <- nrow(subset(train_set, STAN == 1))/nrow(train_set)
test_fraction  <- nrow(subset(test_set, STAN == 1))/nrow(test_set)
train_fraction
```

```
## [1] 0.8763441
```

```
test_fraction
```

```
## [1] 0.8518519
```

(b)

Za pomocą zbioru treningowego skonstruujemy model regresji logistycznej, opisujący zależność między binarną zmienną objaśnianą *stan* i wszystkimi zmiennymi objaśniającymi *RMS10*, ..., *AR*.

```
model1 <- glm(STAN~.-STAN, data = train_set, family = 'binomial')
```

(c)

Za pomocą testu ilorazu wiarygodności zweryfikujemy hipotezę zerową, która stwierdza, że żadna ze zmiennych objaśniających nie ma wpływu na zmienną objaśnianą.

```
nmod <- glm(STAN~1, data = train_set, family = 'binomial') ##"null" mod
lrtest(nmod, model1)
```

```
## Likelihood ratio test
```

```
##
```

```
## Model 1: STAN ~ 1
## Model 2: STAN ~ (RMS10 + RMS20 + RMS30 + PWD + A + DD + YA + YDD + AR) -
##      STAN
##      #Df  LogLik Df  Chisq Pr(>Chisq)
## 1      1 -69.591
## 2     10 -57.354  9 24.474   0.003612 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Na podstawie p-value w danym teście, odrzucamy hipotezę zerową na poziomie istotności $\alpha = 0.05$.

(d)

Dla każdego z parametrów $\beta_i, i = 2, \dots, p$ wyznaczmy wartości estymatorów $\hat{\beta}_i, SE_{\hat{\beta}_i}$ oraz na podstawie p-value odpowiedniego testu sprawdzimy istotność wszystkich zmiennych.

```
summary(model1)
```

```
##
## Call:
## glm(formula = STAN ~ . - STAN, family = "binomial", data = train_set)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8236   0.2109   0.3411   0.5230   1.5997
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.49033     3.37446  -1.923 0.054434 .
## RMS10        -33.96538    22.82629  -1.488 0.136753
## RMS20         19.93734    28.59881   0.697 0.485716
## RMS30         15.04430    18.84525   0.798 0.424692
## PWD           6.74665     2.00856   3.359 0.000782 ***
## A              0.00175     0.01335   0.131 0.895704
## DD           -1.41831     4.13166  -0.343 0.731388
## YA           -0.01144     0.02167  -0.528 0.597540
## YDD          -0.04175     4.71619  -0.009 0.992937
## AR            1.03361     1.65605   0.624 0.532534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 139.18  on 185  degrees of freedom
## Residual deviance: 114.71  on 176  degrees of freedom
## AIC: 134.71
##
## Number of Fisher Scoring iterations: 6
```

Wyznaczymy przedziały ufności dla wszystkich współczynników na poziomie ufności 0.95.

```
exp(confint(model1, , level = 0.95))
```

```
## Waiting for profiling to be done...
##                2.5 %          97.5 %
## (Intercept) 1.268295e-06 8.158381e-01
## RMS10       1.591908e-35 2.560526e+04
## RMS20       6.341436e-16 4.607521e+33
## RMS30       3.044251e-09 8.066114e+23
## PWD         2.144480e+01 6.091819e+04
## A           9.754619e-01 1.028458e+00
## DD          1.360657e-04 3.050103e+03
## YA          9.486501e-01 1.033566e+00
## YDD         1.009711e-04 1.254108e+04
## AR          1.373155e-01 9.783769e+01
```

(e)

Dla dwóch poziomów odcięcia $\pi_0 = \{0.5, 0.7\}$ wyznaczmy tabelę klasyfikacyjną używając zbioru testowego, a następnie wyznaczmy dla każdej z wartości współczynniki (*sensitivity*, *specifity*, *false positive rate*, *false negative rate*, *overall proportion of correct classifications*), które pomogą nam określić zdolności predykcyjne dopasowanego modelu.

$\pi_0 = 0.5$

```
real_values <- test_set$STAN
predictions <- predict(model1, newdata = test_set)

class_tab <- table(real_values, predictions>0.5)
true_negative <- class_tab[1,1]
false_positive <- class_tab[1,2]
false_negative <- class_tab[2,1]
true_positive <- class_tab[2,2]
sensitivity <- (true_positive/(true_positive+false_negative)) * 100
specifity <- (true_negative/(true_negative+false_positive)) * 100
false_positive_rate <- false_positive/(false_positive+true_negative)
false_negative_rate <- false_negative/(false_negative+true_negative)
overall_proportion_of_correct_classifications <- (true_negative+true_positive)/length(test_set$STAN)
kable(class_tab)
```

| | FALSE | TRUE |
|---|-------|------|
| 0 | 1 | 11 |
| 1 | 4 | 65 |

$\pi_0 = 0.7$

```
class_tab_2 <- table(real_values, predictions>0.7)
true_negative_2 <- class_tab_2[1,1]
false_positive_2 <- class_tab_2[1,2]
false_negative_2 <- class_tab_2[2,1]
true_positive_2 <- class_tab_2[2,2]
sensivity_2 <- (true_positive_2/(true_positive_2+false_negative_2)) * 100
specifity_2 <- (true_negative_2/(true_negative_2+false_positive_2)) * 100
false_positive_rate_2 <- false_positive_2/(false_positive_2+true_negative_2)
false_negative_rate_2 <- false_negative_2/(false_negative_2+true_negative_2)
overall_proportion_of_correct_classifications_2 <- (true_negative_2+true_positive_2)/length(test_set$ST)
kable(class_tab_2)
```

| | FALSE | TRUE |
|---|-------|------|
| 0 | 3 | 9 |
| 1 | 4 | 65 |

```
results_1 <- c(sensivity,specifity,false_positive_rate,false_negative_rate,
              overall_proportion_of_correct_classifications)
results_2 <- c(sensivity_2,specifity_2,false_positive_rate_2,false_negative_rate_2,
              overall_proportion_of_correct_classifications_2)
results_table <- matrix(c(results_1, results_2), ncol = 2, byrow = FALSE)
rownames(results_table) <- c('sensivity','specifity','false positive rate',
                             'false negative rate',
                             'overall proportion of correct classification')
kable(results_table, col.names = c(0.5, 0.7), row.names = TRUE)
```

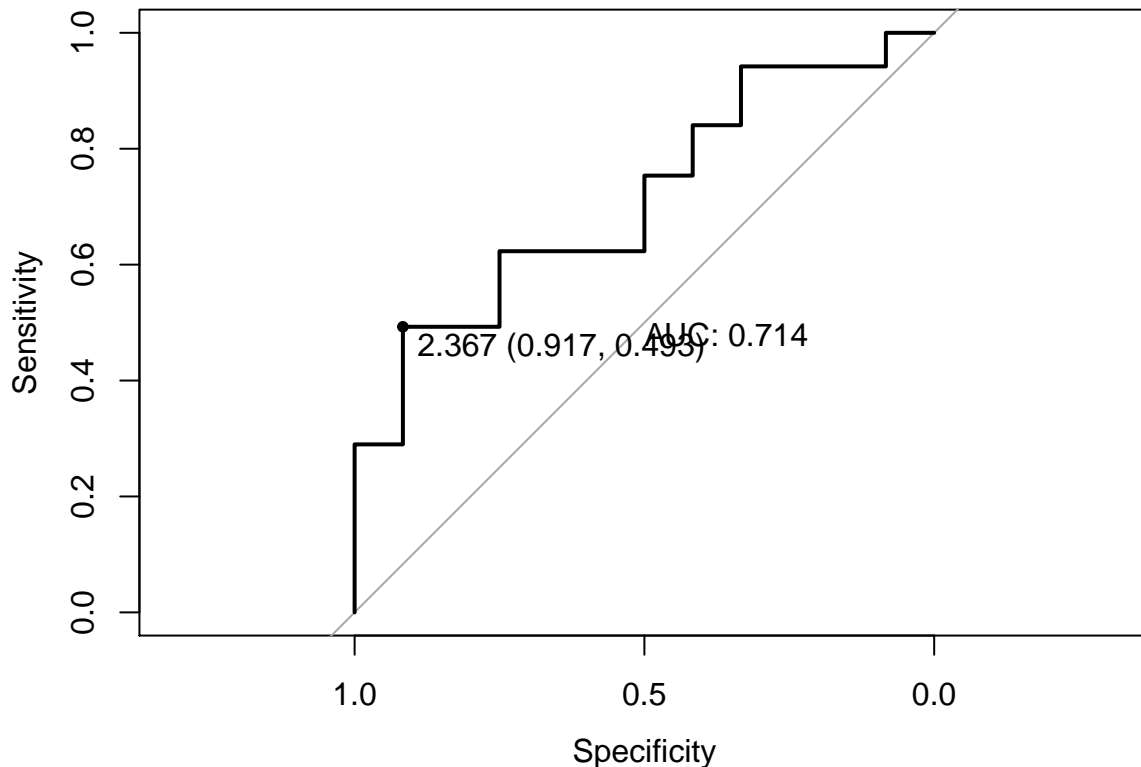
| | 0.5 | 0.7 |
|--|------------|------------|
| sensivity | 94.2028986 | 94.2028986 |
| specifity | 8.3333333 | 25.0000000 |
| false positive rate | 0.9166667 | 0.7500000 |
| false negative rate | 0.8000000 | 0.5714286 |
| overall proportion of correct classification | 0.8148148 | 0.8395062 |

(f)

Korzystając ze zbioru testowego narysujemy krzywą ROC.

```
res.roc <- roc(real_values, predictions)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
plot.roc(res.roc, print.auc = TRUE, print.thres = "best")
```



Sugerując się tą krzywą, możemy powiedzieć że nasz model nie ma bardzo dobrych własności predykcyjnych, ale w niektórych przypadkach mogą być zdecydowanie zadowalające.

Zadanie nr 2

(a) i (b)

W tym zadaniu będziemy analizować dane na temat remisji raka (zmienna objaśniana *remisja* przyjmuje wartość 1, gdy choroba jest w stanie remisji oraz 0, gdy nie jest). Skonstruujemy model regresji logistycznej wykorzystując metodę **selekcji postępującej**. Przyjmujemy poziom istotności 0.35 dla zmiennej, która ma zostać wprowadzona do modelu oraz poziom istotności 0.35 dla zmiennej, która ma pozostać w modelu. Po każdym kroku selekcji, za pomocą testu ilorazu wiarygodności podejmiemy decyzję, czy model aktualny czy poprzedni, lepiej nadaje się do zastosowania. Dodatkowo, wyznaczymy dla każdej ze zmiennej modelu β estymatory: $\hat{\beta}$, $SE_{\hat{\beta}}$, p -value, AIC .

```
cancer_remission <- read_excel("cancer_remission.xlsx",
                               col_types = c("numeric", "numeric", "numeric",
                                              "numeric", "numeric", "numeric",
                                              "numeric", "skip", "skip", "skip",
                                              "skip"))
#cancer_remission$remission <- as.character(cancer_remission$remission)
attach(cancer_remission)
```

Krok pierwszy

```
model_null <- glm(remission~1, data = cancer_remission, family = 'binomial')
model_cell <- glm(remission~cell, data = cancer_remission, family = 'binomial')
```

```

model_smea <- glm(remission~smea, data = cancer_remision, family = 'binomial')
model_infil <- glm(remission~infil, data = cancer_remision, family = 'binomial')
model_li <- glm(remission~li, data = cancer_remision, family = 'binomial')
model_blast <- glm(remission~blast, data = cancer_remision, family = 'binomial')
model_temp <- glm(remission~temp, data = cancer_remision, family = 'binomial')

```

```

#testy ilorazu wiarygodności
lrtest(model_null, model_cell)

```

```

## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ cell
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    1 -17.186
## 2    2 -15.896  1   2.58    0.1082
lrtest(model_null, model_smea)

```

```

## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ smea
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    1 -17.186
## 2    2 -16.640  1  1.0921    0.296
lrtest(model_null, model_infil)

```

```

## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ infil
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    1 -17.186
## 2    2 -16.201  1  1.9698    0.1605
lrtest(model_null, model_li)

```

```

## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ li
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    1 -17.186
## 2    2 -13.037  1  8.2988  0.003967 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
lrtest(model_null, model_blast)

```

```

## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ blast
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    1 -17.186

```

```
## 2 2 -15.410 1 3.5513 0.0595 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

lrtest(model_null, model_temp)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ 1
## Model 2: remission ~ temp
## #Df LogLik Df Chisq Pr(>Chisq)
## 1 1 -17.186
## 2 2 -16.837 1 0.698 0.4035
```

Model uzależniony od zmiennej *li* osiągnął najmniejszą wartość p oraz jest ona mniejsza od przyjętego poziomu istotności, zatem po pierwszym kroku przyjmujemy ten model do dalszej analizy. Wykorzystując funkcję *summary* odczytamy estymatory parametrów.

```
summary(model_li)

##
## Call:
## glm(formula = remission ~ li, family = "binomial", data = cancer_remission)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9448  -0.6465  -0.4947   0.6571   1.6971
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.777      1.379  -2.740  0.00615 **
## li              2.897      1.187   2.441  0.01464 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 26.073  on 25  degrees of freedom
## AIC: 30.073
##
## Number of Fisher Scoring iterations: 4
```

Krok drugi

Powtarzamy powyższą analizę biorąc model uzależniony od zmiennej *li* jako model bazowy.

```
model_li_cell <- glm(remission~li+cell, data = cancer_remission, family = 'binomial')
model_li_smeat <- glm(remission~li+smeat, data = cancer_remission, family = 'binomial')
model_li_infil <- glm(remission~li+infil, data = cancer_remission, family = 'binomial')
model_li_blast <- glm(remission~li+blast, data = cancer_remission, family = 'binomial')
model_li_temp <- glm(remission~li+temp, data = cancer_remission, family = 'binomial')

#testy ilorazu wiarygodności
lrtest(model_li, model_li_cell)
```

```
## Likelihood ratio test
```

```
##
## Model 1: remission ~ li
## Model 2: remission ~ li + cell
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    2 -13.037
## 2    3 -12.170  1  1.7322    0.1881
```

```
lrtest(model_li, model_li_smeas)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li
## Model 2: remission ~ li + smeas
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    2 -13.037
## 2    3 -12.969  1  0.1356    0.7127
```

```
lrtest(model_li, model_li_infil)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li
## Model 2: remission ~ li + infil
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    2 -13.037
## 2    3 -12.745  1  0.5825    0.4453
```

```
lrtest(model_li, model_li_blast)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li
## Model 2: remission ~ li + blast
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    2 -13.037
## 2    3 -12.991  1  0.0918    0.7619
```

```
lrtest(model_li, model_li_temp)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li
## Model 2: remission ~ li + temp
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    2 -13.037
## 2    3 -12.324  1  1.4251    0.2326
```

Dodajemy do naszego modelu zmienną *cell* i przechodzimy do dalszej analizy.

```
summary(model_li_cell)
```

```
##
## Call:
## glm(formula = remission ~ li + cell, family = "binomial", data = cancer_remission)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0026  -0.6582  -0.4477   0.7134   1.4990
```



```
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.586      6.274  -1.528  0.1266
## li           2.879      1.252   2.299  0.0215 *
## cell        6.292      6.152   1.023  0.3065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 24.341  on 24  degrees of freedom
## AIC: 30.341
##
## Number of Fisher Scoring iterations: 6
```

Krok trzeci

Powtarzamy powyższą analizę biorąc model uzależniony od zmiennych *li* i *cell* jako model bazowy.

```
model_li_cell_smeas <- glm(remission~li+cell+smeas, data = cancer_remision,
                           family = 'binomial')
model_li_cell_infil <- glm(remission~li+cell+infil, data = cancer_remision,
                           family = 'binomial')
model_li_cell_blast <- glm(remission~li+cell+blast, data = cancer_remision,
                           family = 'binomial')
model_li_cell_temp <- glm(remission~li+cell+temp, data = cancer_remision,
                           family = 'binomial')
```

#testy ilorazu wiarygodności

```
lrtest(model_li_cell, model_li_cell_smeas)
```

```
## Likelihood ratio test
```

```
##
```

```
## Model 1: remission ~ li + cell
```

```
## Model 2: remission ~ li + cell + smeas
```

```
##   #Df LogLik Df  Chisq Pr(>Chisq)
```

```
## 1    3 -12.170
```

```
## 2    4 -12.146  1 0.0483    0.8261
```

```
lrtest(model_li_cell, model_li_cell_infil)
```

```
## Likelihood ratio test
```

```
##
```

```
## Model 1: remission ~ li + cell
```

```
## Model 2: remission ~ li + cell + infil
```

```
##   #Df LogLik Df  Chisq Pr(>Chisq)
```

```
## 1    3 -12.170
```

```
## 2    4 -12.151  1 0.0391    0.8433
```

```
lrtest(model_li_cell, model_li_cell_blast)
```

```
## Likelihood ratio test
```

```
##
```

```
## Model 1: remission ~ li + cell
```

```
## Model 2: remission ~ li + cell + blast
```

```
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    3 -12.170
## 2    4 -12.093  1 0.1545    0.6943
lrtest(model_li_cell, model_li_cell_temp)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li + cell
## Model 2: remission ~ li + cell + temp
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1    3 -12.170
## 2    4 -10.977  1 2.3874    0.1223
```

Dodajemy do naszego modelu zmienną *temp* i przechodzimy do dalszej analizy.

```
summary(model_li_cell_temp)

##
## Call:
## glm(formula = remission ~ li + cell + temp, family = "binomial",
##      data = cancer_remission)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02043  -0.66313  -0.08323   0.81282   1.65887
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   67.634     56.888   1.189  0.2345
## li             3.867      1.778   2.175  0.0297 *
## cell          9.652      7.751   1.245  0.2130
## temp        -82.074     61.712  -1.330  0.1835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 34.372  on 26  degrees of freedom
## Residual deviance: 21.953  on 23  degrees of freedom
## AIC: 29.953
##
## Number of Fisher Scoring iterations: 7
```

Krok czwarty

Powtarzamy powyższą analizę biorąc model uzależniony od zmiennych *li*, *cell* i *temp* jako model bazowy.

```
model_li_cell_temp_smeat <- glm(remission~li+cell+temp+smeat, data = cancer_remission,
                                family = 'binomial')
model_li_cell_temp_infil <- glm(remission~li+cell+temp+infil, data = cancer_remission,
                                family = 'binomial')
model_li_cell_temp_blast <- glm(remission~li+cell+temp+blast, data = cancer_remission,
                                family = 'binomial')

#testy ilorazu wiarygodności
lrtest(model_li_cell_temp, model_li_cell_temp_smeat)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li + cell + temp
## Model 2: remission ~ li + cell + temp + smear
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    4 -10.977
## 2    5 -10.929  1 0.0954    0.7574
lrtest(model_li_cell_temp, model_li_cell_temp_infil)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li + cell + temp
## Model 2: remission ~ li + cell + temp + infil
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    4 -10.977
## 2    5 -10.935  1 0.0842    0.7716
lrtest(model_li_cell_temp, model_li_cell_temp_blast)
```

```
## Likelihood ratio test
##
## Model 1: remission ~ li + cell + temp
## Model 2: remission ~ li + cell + temp + blast
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1    4 -10.977
## 2    5 -10.966  1 0.0209    0.8852
```

Widzimy, że dodanie jakiegokolwiek ze zmiennej pozostałych nie poprawi nam jakości modelu, ponieważ p-value dla wszystkich zmiennych jest większe od przyjętego poziomu istotności.

Ostatecznie, optymalny model według metody **selekcji postępującej** jest uzależniony od zmiennych: *li*, *cell* i *temp*.

(c)

Wykorzystując model otrzymany w podpunkcie (a), skonstruujemy tabelę zawierającą:

- wartości zmiennych $(y, x) = (remiss, cell, smear, ifnil, li, blast, temp)$
- prognozowane przez model prawdopodobieństwo $\pi_0(x)$
- lewy i prawy koniec 95% przedziału ufności (ilorazu wiarygodności) dla $\pi_0(x) := P(remiss = 1 | \mathbf{X} = \mathbf{x})$

```
inverse_logit = function(x){  
  exp(x)/(1+exp(x))  
}  
df_table <- data.frame(cancer_remission)  
predicted <- predict(model_li_cell_temp, data = cancer_remission, type='link', se.fit=TRUE)  
df_table$pi_x <- inverse_logit(predicted$fit)  
df_table$confint_high <- inverse_logit(predicted$fit + (predicted$se.fit*qnorm(0.975)))  
df_table$confint_low <- inverse_logit(predicted$fit - (predicted$se.fit*qnorm(0.975)))  
kable(df_table)
```

| remission | cell | smear | infil | li | blast | temp | pi_x | confint_high | confint_low |
|-----------|------|-------|-------|-----|-------|-------|-----------|--------------|-------------|
| 1 | 0.80 | 0.83 | 0.66 | 1.9 | 1.100 | 0.996 | 0.7226489 | 0.9709308 | 0.1689203 |
| 1 | 0.90 | 0.36 | 0.32 | 1.4 | 0.740 | 0.992 | 0.5787391 | 0.8376193 | 0.2678769 |
| 0 | 0.80 | 0.88 | 0.70 | 0.8 | 0.176 | 0.982 | 0.1045990 | 0.6341884 | 0.0078100 |
| 0 | 1.00 | 0.87 | 0.87 | 0.7 | 1.053 | 0.986 | 0.2825773 | 0.6568252 | 0.0749793 |
| 1 | 0.90 | 0.75 | 0.68 | 1.3 | 0.519 | 0.980 | 0.7141804 | 0.9487569 | 0.2521795 |
| 0 | 1.00 | 0.65 | 0.65 | 0.6 | 0.519 | 0.982 | 0.2708868 | 0.6895121 | 0.0585194 |
| 1 | 0.95 | 0.97 | 0.92 | 1.0 | 1.230 | 0.992 | 0.3215554 | 0.5951615 | 0.1325485 |
| 0 | 0.95 | 0.87 | 0.83 | 1.9 | 1.354 | 1.020 | 0.6072319 | 0.9528731 | 0.1057171 |
| 0 | 1.00 | 0.45 | 0.45 | 0.8 | 0.322 | 0.999 | 0.1663164 | 0.5612346 | 0.0301751 |
| 0 | 0.95 | 0.36 | 0.34 | 0.5 | 0.000 | 1.038 | 0.0015693 | 0.6896189 | 0.0000011 |
| 0 | 0.85 | 0.39 | 0.33 | 0.7 | 0.279 | 0.988 | 0.0728520 | 0.4998246 | 0.0061407 |
| 0 | 0.70 | 0.76 | 0.53 | 1.2 | 0.146 | 0.982 | 0.1728570 | 0.8720619 | 0.0063664 |
| 0 | 0.80 | 0.46 | 0.37 | 0.4 | 0.380 | 1.006 | 0.0034575 | 0.4652987 | 0.0000138 |
| 0 | 0.20 | 0.39 | 0.08 | 0.8 | 0.114 | 0.990 | 0.0001850 | 0.9648173 | 0.0000000 |
| 0 | 1.00 | 0.90 | 0.90 | 1.1 | 1.037 | 0.990 | 0.5712204 | 0.8397279 | 0.2530256 |
| 1 | 1.00 | 0.84 | 0.84 | 1.9 | 2.064 | 1.020 | 0.7146954 | 0.9718896 | 0.1536176 |
| 0 | 0.65 | 0.42 | 0.27 | 0.5 | 0.114 | 1.014 | 0.0006223 | 0.6266526 | 0.0000002 |
| 0 | 1.00 | 0.75 | 0.75 | 1.0 | 1.322 | 1.004 | 0.2228888 | 0.6367021 | 0.0448348 |
| 0 | 0.50 | 0.44 | 0.22 | 0.6 | 0.114 | 0.990 | 0.0015425 | 0.7964415 | 0.0000006 |
| 1 | 1.00 | 0.63 | 0.63 | 1.1 | 1.072 | 0.986 | 0.6491095 | 0.9055470 | 0.2630488 |
| 0 | 1.00 | 0.33 | 0.33 | 0.4 | 0.176 | 1.010 | 0.0169297 | 0.5047489 | 0.0002909 |
| 0 | 0.90 | 0.93 | 0.84 | 0.6 | 1.591 | 1.020 | 0.0062175 | 0.5606165 | 0.0000307 |
| 1 | 1.00 | 0.58 | 0.58 | 1.0 | 0.531 | 1.002 | 0.2526057 | 0.6359717 | 0.0613730 |
| 0 | 0.95 | 0.32 | 0.30 | 1.6 | 0.886 | 0.988 | 0.8701089 | 0.9848055 | 0.4091050 |
| 1 | 1.00 | 0.60 | 0.60 | 1.7 | 0.964 | 0.990 | 0.9313166 | 0.9957251 | 0.4411429 |
| 1 | 1.00 | 0.69 | 0.69 | 0.9 | 0.398 | 0.986 | 0.4605092 | 0.7852916 | 0.1661227 |
| 0 | 1.00 | 0.73 | 0.73 | 0.7 | 0.398 | 0.986 | 0.2825773 | 0.6568252 | 0.0749793 |

Użyliśmy funkcji *inverse logit*, aby ze skali logarytmicznej wrócić do skali probabilistycznej 0 – 1.