11 Regresja logistyczna i Poissona

11.1 Przykłady

Regresja logistyczna

Przykład. Rozważmy przykład dotyczący badania szansy ponownego ataku serca w ciągu roku od pierwszego ataku, w zależności od *treatment of anger* oraz *trait anxiety*. Zmienna zależna ma wartość 1, jeśli nastąpił ponowny atak, a 0 w przeciwnym razie.

```
y <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
x1 <- c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0)
x2 <- c(70, 80, 50, 60, 40, 65, 75, 80, 70, 60, 65, 50, 45, 35, 40, 50, 55, 45, 50, 60)
data_set <- data.frame(y, x1, x2)
head(data_set)
```

```
##   y x1 x2
## 1 1  1 70
## 2 1  1 80
## 3 1  1 50
## 4 1  0 60
## 5 1  0 40
## 6 1  0 65
```

```
# model logistyczny
model_1 <- glm(y ~ x1 + x2, data = data_set, family = 'binomial')
model_1
```

```
## 
## Call:  glm(formula = y ~ x1 + x2, family = "binomial", data = data_set)
## 
## Coefficients:
## (Intercept)           x1           x2  
##      -6.363       -1.024        0.119  
## 
## Degrees of Freedom: 19 Total (i.e. Null);  17 Residual
## Null Deviance:       27.73 
## Residual Deviance: 18.82     AIC: 24.82
```

```r
# podsumowanie modelu
# tj. reszty, estymacja punktowa, testy istotności dla współczynników regresji, AIC
summary(model_1)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2, family = "binomial", data = data_set)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.52106  -0.68746   0.00424   0.70625   1.88960
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.36347    3.21362  -1.980   0.0477 *
## x1          -1.02411    1.17101  -0.875   0.3818
## x2           0.11904    0.05497   2.165   0.0304 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 18.820  on 17  degrees of freedom
## AIC: 24.82
##
## Number of Fisher Scoring iterations: 4
```

```r
# zredukowany model logistyczny
model_2 <- glm(y ~ x2, data = data_set, family = 'binomial')
summary(model_2)
```

```
## 
## Call:
## glm(formula = y ~ x2, family = "binomial", data = data_set)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.62461  -0.83983  -0.01232   0.64540   2.10801  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  -7.0925     3.1709  -2.237   0.0253 *
## x2            0.1246     0.0553   2.254   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 19.601  on 18  degrees of freedom
## AIC: 23.601
## 
## Number of Fisher Scoring iterations: 4
```

```r
# regresja krokowa
AIC(model_1, model_2)
```

```
##         df      AIC
## model_1  3 24.82037
## model_2  2 23.60052
```

```r
step(model_1)
```

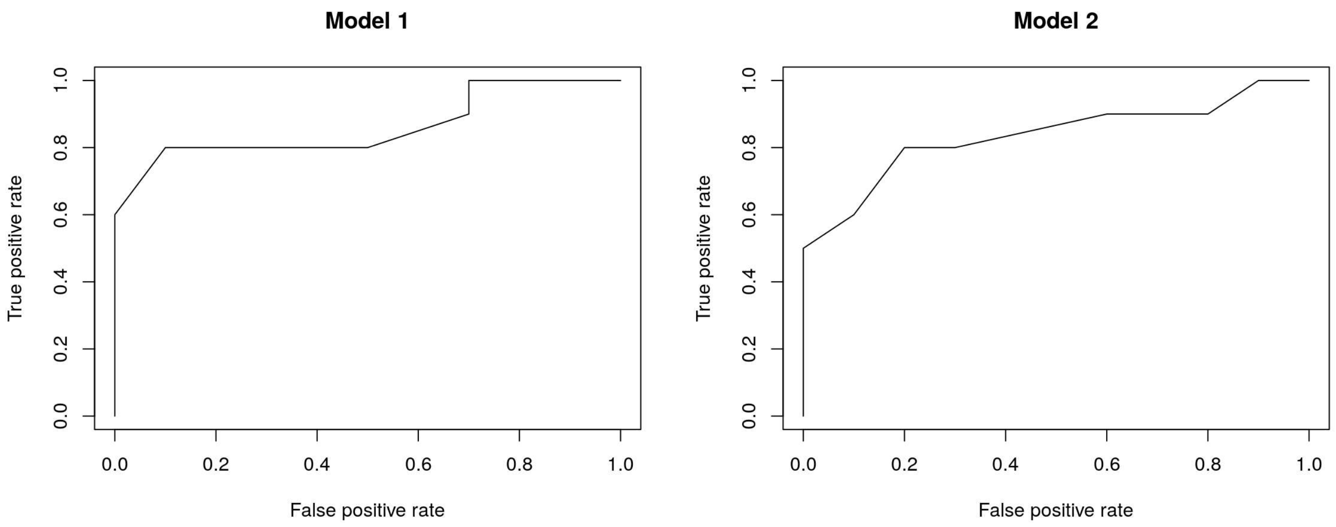
```
## Start:  AIC=24.82
## y ~ x1 + x2
##
##        Df Deviance    AIC
## - x1    1   19.601 23.601
## <none>      18.820 24.820
## - x2    1   25.878 29.878
##
## Step:  AIC=23.6
## y ~ x2
##
##        Df Deviance    AIC
## <none>      19.601 23.601
## - x2    1   27.726 29.726
```

```
##
## Call:  glm(formula = y ~ x2, family = "binomial", data = data_set)
##
## Coefficients:
## (Intercept)           x2
##     -7.0925       0.1246
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:       27.73
## Residual Deviance: 19.6  AIC: 23.6
```

```
# iloraz szans (ręcznie)
exp(coef(model_2)[2])
```

```
##       x2
## 1.132734
```

```r
# Wartość ta oznacza, że wraz ze wzrostem wartości zmiennej x2 o jedną jednostkę,
# przewidywane ryzyko ponownego zawału serca wzrasta o 13%.
# do krzywych ROC
library(ROCR)
pred_1 <- prediction(model_1$fitted, y)
pred_2 <- prediction(model_2$fitted, y)
# krzywe ROC
par(mfrow = c(1, 2))
plot(performance(pred_1, 'tpr', 'fpr'), main = "Model 1")
plot(performance(pred_2, 'tpr', 'fpr'), main = "Model 2")
```




```r
par(mfrow = c(1, 1))
# AUC
performance(pred_1, 'auc')@y.values
```

```
## [[1]]
## [1] 0.86
```

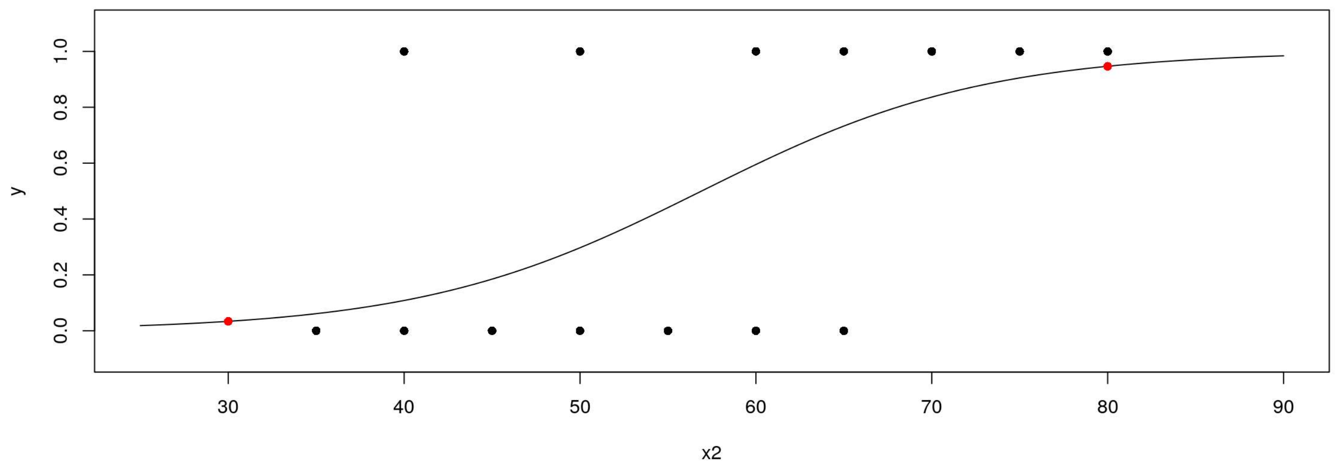
```r
performance(pred_2, 'auc')@y.values
```

```
## [[1]]
## [1] 0.835
```

```r
# predykcja
(predict_glm <- predict(model_2,
                        data.frame(x2 = c(30, 80)),
                        type = 'response'))
```

```
##          1          2
## 0.03378247 0.94676209
```

```r
# Uwzględniamy argument type = 'response' w celu uzyskania przewidywanego prawdopodobie
# Domyślne przewidywane są zlogarytmowane ilorazy szans (prawdopodobieństwa w skali log
x_temp <- seq(min(x2) - 10, max(x2) + 10, length.out = 100)
y_temp <- exp(coef(model_2)[1] + coef(model_2)[2] * x_temp) /
  (1 + exp(coef(model_2)[1] + coef(model_2)[2] * x_temp))
plot(x_temp, y_temp, type = "l", xlab = "x2", ylab = "y", ylim = c(-0.1, 1.1))
points(x2, y, pch = 16)
points(c(30, 80), predict_glm, pch = 16, col = "red")
```



Regresja Poissona

Nie zawsze interesuje nas prawdopodobieństwo sukcesu. Dość często jesteśmy zainteresowani liczbą sukcesów (ogólnie liczebnościami). W tej sytuacji najbardziej popularny jest model Poissona, który zakłada, że zmienna zależna ma rozkład Poissona i

$$h(x) = \ln(x), \quad E(Y) = \exp(\mathbf{X}\boldsymbol{\beta}).$$

Przykład. W zbiorze danych student_award.RData, zmienna `num_awards` podaje liczbę nagród zdobytych przez uczniów szkoły średniej przez rok, zmienna `math` jest zmienną ciągłą i reprezentuje wyniki uczniów na końcowym egzaminie z matematyki, a zmienna `prog` jest zmienną jakościową z trzema poziomami wskazującymi rodzaj programu, ma który uczniowie byli zapisani (“General” - ogólny, “Academic” - akademicki, “Vocational” - zawodowy). Chcemy opisać związek między liczbą nagród a wynikiem egzaminu z matematyki i programem.

```
load(url("http://ls.home.amu.edu.pl/data_sets/student_award.RData"))
head(student_award)
```

```
##   num_awards math       prog
## 1          0   41 Vocational
## 2          0   41    General
## 3          0   44 Vocational
## 4          0   42 Vocational
## 5          0   40 Vocational
## 6          0   42    General
```

```
model_1 <- glm(num_awards ~ math + prog, data = student_award, family = "poisson")
model_1
```

```
## 
## Call:  glm(formula = num_awards ~ math + prog, family = "poisson", data = student_awa
## 
## Coefficients:
##    (Intercept)            math    progAcademic  progVocational  
##       -5.24712         0.07015         1.08386         0.36981  
## 
## Degrees of Freedom: 199 Total (i.e. Null);  196 Residual
## Null Deviance:       287.7 
## Residual Deviance: 189.4     AIC: 373.5
```

```
summary(model_1)
```

```
##
## Call:
## glm(formula = num_awards ~ math + prog, family = "poisson", data = student_award)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2043  -0.8436  -0.5106   0.2558   2.6796
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -5.24712    0.65845  -7.969 1.60e-15 ***
## math            0.07015    0.01060   6.619 3.63e-11 ***
## progAcademic    1.08386    0.35825   3.025  0.00248 **
## progVocational  0.36981    0.44107   0.838  0.40179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.45  on 196  degrees of freedom
## AIC: 373.5
##
## Number of Fisher Scoring iterations: 6
```

```r
# Możemy również przetestować ogólny efekt programu, porównując pełny model
# z modelem bez zmiennej program. Test chi-kwadrat wskazuje, że program,
# jest statystycznie istotnym predyktorem liczby nagród.
model_2 <- update(model_1, . ~ . - prog)
anova(model_1, model_2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: num_awards ~ math + prog
## Model 2: num_awards ~ math
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       196     189.45
## 2       198     204.02 -2  -14.572 0.0006852 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(model_1, model_2)
```

```
##         df      AIC
## model_1  4 373.5045
## model_2  2 384.0762
```

```r
step(model_1)
```

```
## Start:  AIC=373.5
## num_awards ~ math + prog
##
##        Df Deviance    AIC
## <none>      189.45 373.50
## - prog  2   204.02 384.08
## - math  1   234.46 416.51
```

```
##
## Call:  glm(formula = num_awards ~ math + prog, family = "poisson", data = student_awa
##
## Coefficients:
##    (Intercept)            math    progAcademic  progVocational
##       -5.24712         0.07015         1.08386         0.36981
##
## Degrees of Freedom: 199 Total (i.e. Null);  196 Residual
## Null Deviance:       287.7
## Residual Deviance: 189.4     AIC: 373.5
```

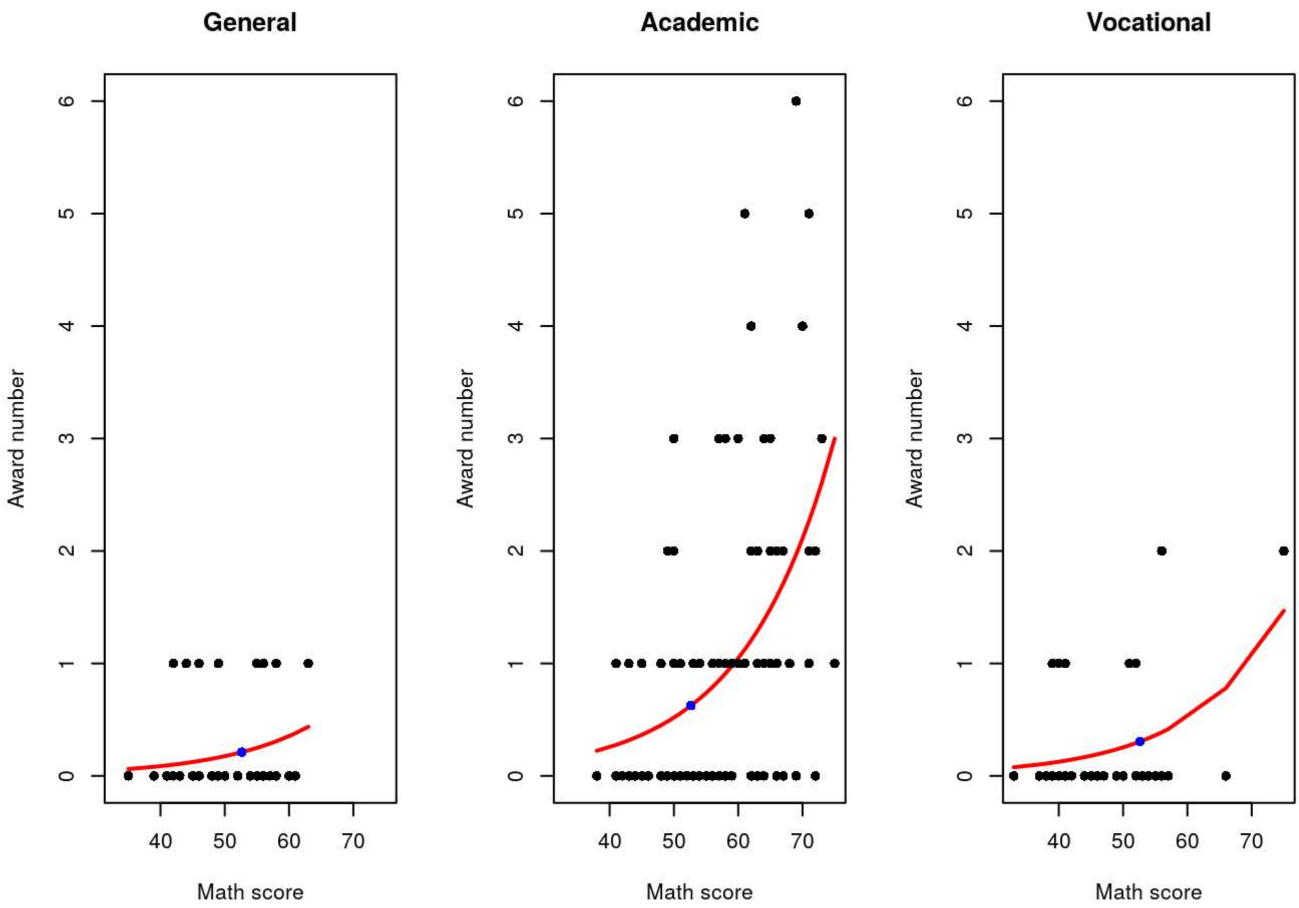
```r
(data_new <- data.frame(math = mean(student_award$math),
                        prog = factor(1:3, levels = 1:3,
                                      labels = levels(student_award$prog))))
```

```
##     math       prog
## 1 52.645    General
## 2 52.645   Academic
## 3 52.645 Vocational
```

```r
(pred <- predict(model_1, data_new, type = "response"))
```

```
##         1         2         3
## 0.2114109 0.6249446 0.3060086
```

```r
student_award$num_award_hat <- predict(model_1, type = "response")
# sortowanie według programu, a następnie według wyniku z matematyki
student_award <- student_award[with(student_award, order(prog, math)), ]
par(mfrow = c(1, 3))
plot(student_award$math[student_award$prog == "General"],
     student_award$num_award_hat[student_award$prog == "General"],
     type = "l", lwd = 2, col = "red",
     xlim = c(min(student_award$math), max(student_award$math)), ylim = c(0, 6),
     xlab = "Math score", ylab = "Award number", main = "General")
points(student_award$math[student_award$prog == "General"],
       student_award$num_awards[student_award$prog == "General"], pch = 16)
points(mean(student_award$math), pred[1], pch = 16, col = "blue", lwd = 4)
plot(student_award$math[student_award$prog == "Academic"],
     student_award$num_award_hat[student_award$prog == "Academic"],
     type = "l", lwd = 2, col = "red",
     xlim = c(min(student_award$math), max(student_award$math)), ylim = c(0, 6),
     xlab = "Math score", ylab = "Award number", main = "Academic")
points(student_award$math[student_award$prog == "Academic"],
       student_award$num_awards[student_award$prog == "Academic"], pch = 16)
points(mean(student_award$math), pred[2], pch = 16, col = "blue", lwd = 4)
plot(student_award$math[student_award$prog == "Vocational"],
     student_award$num_award_hat[student_award$prog == "Vocational"],
     type = "l", lwd = 2, col = "red",
     xlim = c(min(student_award$math), max(student_award$math)), ylim = c(0, 6),
     xlab = "Math score", ylab = "Award number", main = "Vocational")
points(student_award$math[student_award$prog == "Vocational"],
       student_award$num_awards[student_award$prog == "Vocational"], pch = 16)
points(mean(student_award$math), pred[3], pch = 16, col = "blue", lwd = 4)
```




```
par(mfrow = c(1, 1))
```

11.2 Zadania

Zadanie 1. W jednym badaniu klinicznym oceniono wpływ poziomów enzymu LDH i zmian poziomów bilirubiny na zdrowie pacjentów z przewlekłym zapaleniem wątroby. Uzyskane wyniki są zawarte w pliku liver_data.RData. Zmienne to: `bilirubin` - zmiana stężenia bilirubiny we krwi, `ldh` - stężenie enzymu LDH w ciele pacjenta, `condition` - zmiana stanu pacjenta (`Yes` - pogorszenie, `No` - brak pogorszenia).

```
##   bilirubin ldh condition
## 1       0.9  75        No
## 2       0.8 150        No
## 3       0.6 250        No
## 4       0.8 375       Yes
## 5       3.2 160       Yes
## 6       1.7 106        No
```

1. Dopasuj model regresji logistycznej do tych danych. Jakie są wartości estymatorów współczynników regresji?

```
##
## Call:  glm(formula = condition ~ bilirubin + ldh, family = "binomial",
##     data = liver_data)
##
## Coefficients:
## (Intercept)    bilirubin          ldh
##    -8.13113      2.88050      0.02464
##
## Degrees of Freedom: 38 Total (i.e. Null);  36 Residual
## Null Deviance:       54.04
## Residual Deviance: 33.11     AIC: 39.11
```

2. Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
##
## Call:
## glm(formula = condition ~ bilirubin + ldh, family = "binomial",
##     data = liver_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.05593  -0.79191   0.04353   0.57765   2.11829
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.131132   2.639959  -3.080  0.00207 **
## bilirubin    2.880497   1.105836   2.605  0.00919 **
## ldh          0.024635   0.008764   2.811  0.00494 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 54.040  on 38  degrees of freedom
## Residual deviance: 33.114  on 36  degrees of freedom
## AIC: 39.114
##
## Number of Fisher Scoring iterations: 6
```

3. Czy model ten może być zredukowany za pomocą regresji krokowej?

```
## Start:  AIC=39.11
## condition ~ bilirubin + ldh
##
##             Df Deviance    AIC
## <none>           33.114 39.114
## - ldh        1   46.989 50.989
## - bilirubin  1   48.726 52.726
```

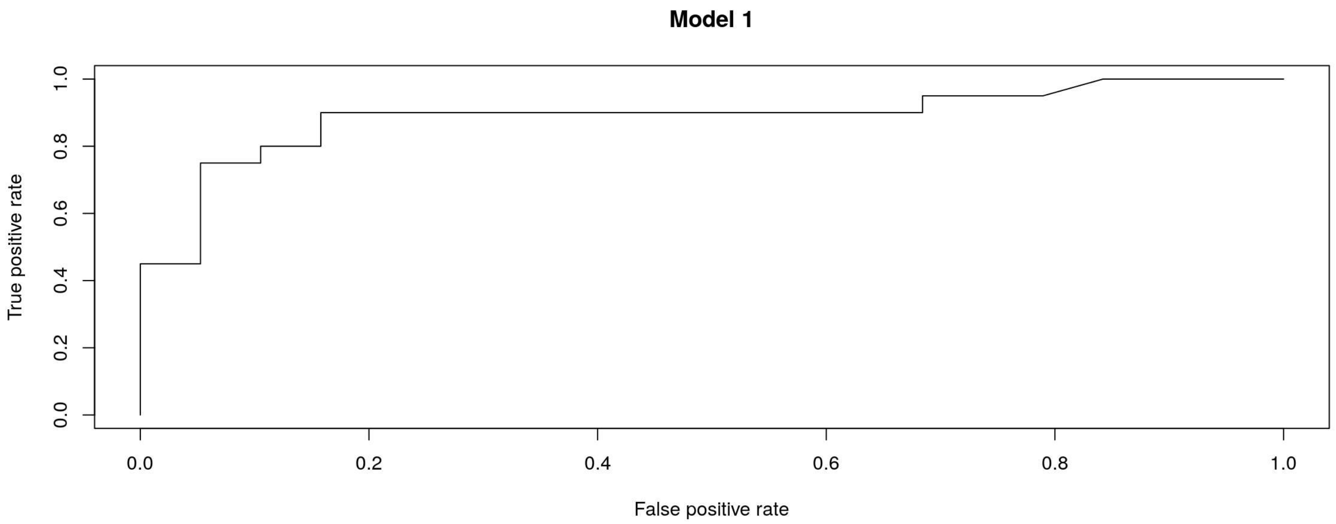
```
##
## Call:  glm(formula = condition ~ bilirubin + ldh, family = "binomial",
##     data = liver_data)
##
## Coefficients:
## (Intercept)    bilirubin          ldh
##    -8.13113      2.88050      0.02464
##
## Degrees of Freedom: 38 Total (i.e. Null);  36 Residual
## Null Deviance:       54.04
## Residual Deviance: 33.11     AIC: 39.11
```

4. Zinterpretuj współczynniki modelu.

```
## bilirubin
##  17.82313
```

```
##      ldh
## 1.024941
```

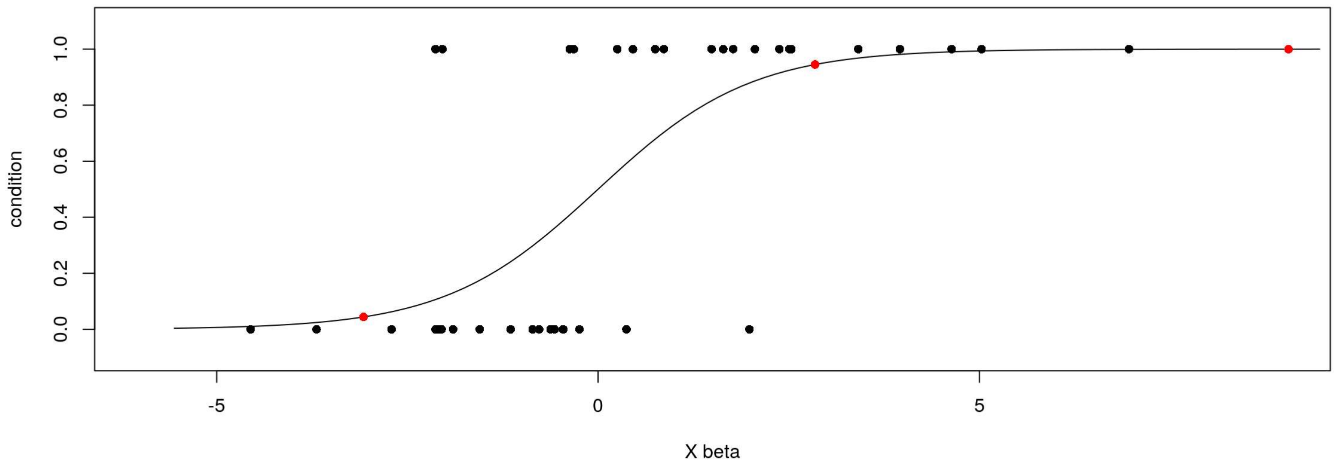
5. Narysuj krzywą ROC i oblicz AUC dla modelu.




```
## [[1]]
## [1] 0.8881579
```

6. Dokonaj predykcji zmiennej `condition` dla trzech pacjentów scharakteryzonych następująco: (`bilirubin` , `ldh`) = (0.9, 100), (2.1, 200), (3.4, 300). Zilustruj wyniki na wykresie.

```
##          1          2          3
## 0.04414365 0.94505776 0.99988299
```



7. Powyższy wykres pokazuje, że istnieją dwie obserwacje odstające dla pacjentów z pogorszeniem i jedna obserwacja odstająca dla pacjentów bez pogorszenia. Zidentyfikuj je i wykonaj powyższą analizę dla danych bez tych trzech wartości odstających. Jak zmieniają się wyniki?

   1.

   ```
   ##
   ## Call:  glm(formula = condition ~ bilirubin + ldh, family = "binomial",
   ##     data = liver_data_wo)
   ##
   ## Coefficients:
   ## (Intercept)    bilirubin          ldh
   ##    -72.7256      30.2781       0.1947
   ##
   ## Degrees of Freedom: 35 Total (i.e. Null);  33 Residual
   ## Null Deviance:       49.91
   ## Residual Deviance: 6.207     AIC: 12.21
   ```

   2.

```
##
## Call:
## glm(formula = condition ~ bilirubin + ldh, family = "binomial",
##     data = liver_data_wo)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.93161  -0.01879   0.00000   0.00047   1.89807
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -72.7256    45.3298  -1.604    0.109
## bilirubin    30.2781    18.9417   1.598    0.110
## ldh           0.1947     0.1235   1.577    0.115
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 49.9066  on 35  degrees of freedom
## Residual deviance:  6.2068  on 33  degrees of freedom
## AIC: 12.207
##
## Number of Fisher Scoring iterations: 10
```

3.

```
## Start:  AIC=12.21
## condition ~ bilirubin + ldh
##
##             Df Deviance    AIC
## <none>             6.207 12.207
## - ldh        1   38.422 42.422
## - bilirubin  1   44.216 48.216
```

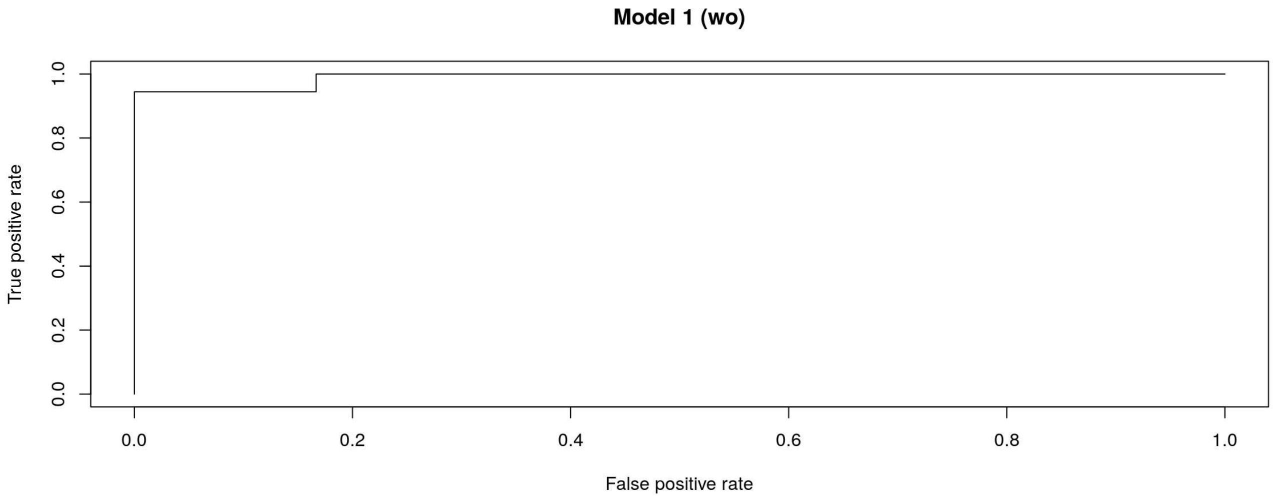
```
##
## Call:  glm(formula = condition ~ bilirubin + ldh, family = "binomial",
##     data = liver_data_wo)
##
## Coefficients:
## (Intercept)    bilirubin          ldh
##    -72.7256      30.2781       0.1947
##
## Degrees of Freedom: 35 Total (i.e. Null);  33 Residual
## Null Deviance:       49.91
## Residual Deviance: 6.207     AIC: 12.21
```

4.

```
##    bilirubin
## 1.411294e+13
```

```
##      ldh
## 1.214999
```

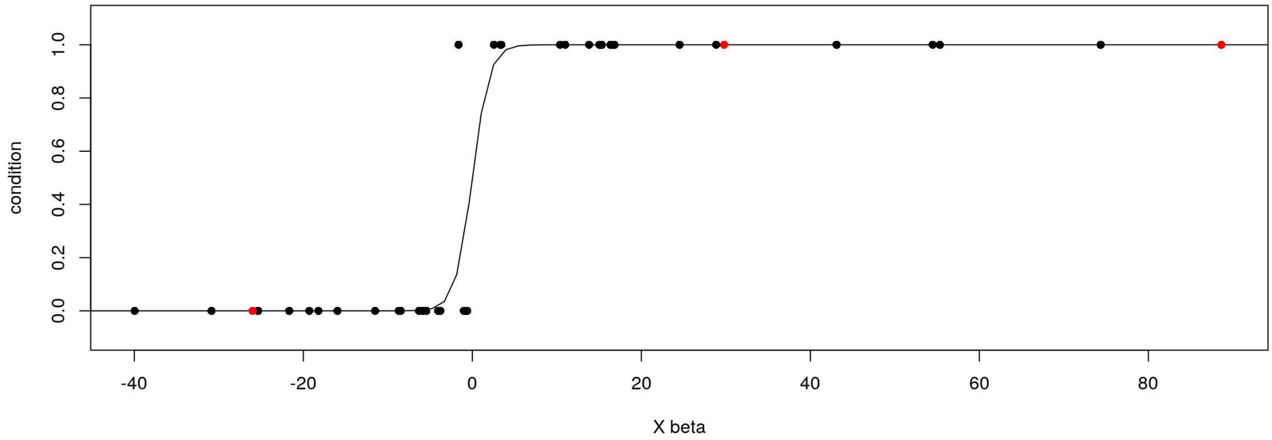
5.




```
## [[1]]
## [1] 0.9907407
```

6.

```
##            1            2            3
## 5.104082e-12 1.000000e+00 1.000000e+00
```



Zadanie 2. Użyj modelu regresji Poissona do zestawu danych `moths` (wpływ siedliska na liczbę moli) z pakietu `DAAG`. Użyj zlogarytmowanej zmiennej `meters` jako zmiennej objaśniającej, a liczby moli `A` jako zmiennej objaśnianej.

```
##   meters A  P   habitat
## 1     25 9  8    NWsoak
## 2     37 3 20    SWsoak
## 3    109 7  9 Lowerside
## 4     10 0  2 Lowerside
## 5    133 9  1 Upperside
## 6     26 3 18 Disturbed
```

1. Dopasuj model regresji Poissona do tych danych. Jakie są wartości estymatorów współczynników regresji?

```
##
## Call:  glm(formula = A ~ log(meters), family = "poisson", data = moths)
##
## Coefficients:
## (Intercept)  log(meters)
##      1.2058       0.1506
##
## Degrees of Freedom: 40 Total (i.e. Null);  39 Residual
## Null Deviance:       257.1
## Residual Deviance: 248.3     AIC: 367
```

2. Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
##
## Call:
## glm(formula = A ~ log(meters), family = "poisson", data = moths)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4366  -1.7754  -1.1501   0.7331   9.2711
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.20577    0.17814   6.769  1.3e-11 ***
## log(meters)  0.15065    0.05068   2.972  0.00295 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 257.11  on 40  degrees of freedom
## Residual deviance: 248.25  on 39  degrees of freedom
## AIC: 366.97
##
## Number of Fisher Scoring iterations: 6
```

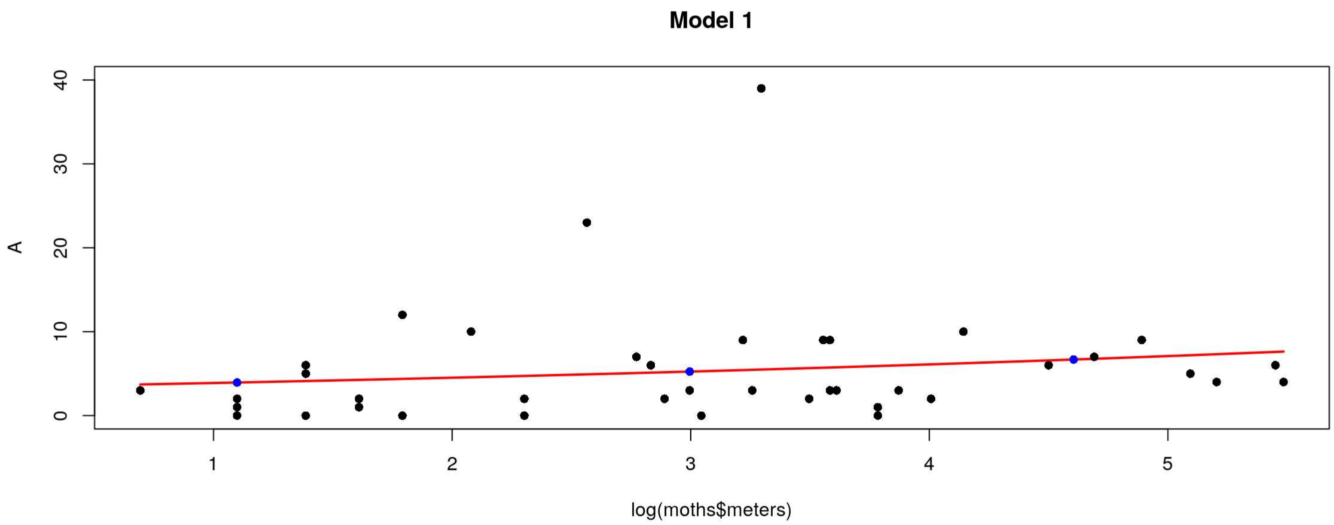
3. Czy model ten może być zredukowany za pomocą regresji krokowej?

```
## Start:  AIC=366.97
## A ~ log(meters)
##
##                Df Deviance    AIC
## <none>              248.25 366.97
## - log(meters)  1    257.11 373.83
```

```
##
## Call:  glm(formula = A ~ log(meters), family = "poisson", data = moths)
##
## Coefficients:
## (Intercept)  log(meters)
##      1.2058       0.1506
##
## Degrees of Freedom: 40 Total (i.e. Null);  39 Residual
## Null Deviance:       257.1
## Residual Deviance: 248.3     AIC: 367
```

4. Dokonaj predykcji zmiennej `A` dla `meters = 3, 20, 100` . Zilustruj wyniki na wykresie.

```
##        1        2        3
## 3.940363 5.243913 6.682717
```



5. Wykonaj powyższą analizę dla zmiennej `P` jako zmiennej zależnej.

```
## 1.
```

```
##
## Call:  glm(formula = P ~ log(meters), family = "poisson", data = moths)
##
## Coefficients:
## (Intercept)  log(meters)
##      0.8643       0.1372
##
## Degrees of Freedom: 40 Total (i.e. Null);  39 Residual
## Null Deviance:       217.8
## Residual Deviance: 212.8     AIC: 309
```

```
## 2.
```

```
##
## Call:
## glm(formula = P ~ log(meters), family = "poisson", data = moths)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8679  -2.3492  -1.1408   0.6247   5.7649
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8643     0.2145   4.030 5.58e-05 ***
## log(meters)   0.1372     0.0614   2.234   0.0255 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 217.82  on 40  degrees of freedom
## Residual deviance: 212.82  on 39  degrees of freedom
## AIC: 309.05
##
## Number of Fisher Scoring iterations: 6
```

```
## 3.
```

```
## Start:  AIC=309.05
## P ~ log(meters)
##
##               Df Deviance    AIC
## <none>             212.82 309.05
## - log(meters)  1   217.82 312.04
```

```
##
## Call:  glm(formula = P ~ log(meters), family = "poisson", data = moths)
##
## Coefficients:
## (Intercept)  log(meters)
##      0.8643       0.1372
##
## Degrees of Freedom: 40 Total (i.e. Null);  39 Residual
## Null Deviance:       217.8
## Residual Deviance: 212.8     AIC: 309
```

4.

```
##        1        2        3
## 2.759453 3.579565 4.463761
```