Uogólnione modele liniowe – wprowadzenie praktyczne.

Agnieszka Piliszek

Wydział Matematyki i Nauk Informacyjnych

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. „Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych”, realizowane w ramach projektu „NERW 2 PW. Nauka – Edukacja — Rozwój – Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Spis treści

Rozdział 1

Praktyczne wprowadzenie do uogólnionych modeli liniowych

W modelach liniowych, które już znamy, zakłada się, że wariancja jest stała. Jednak w wielu sytuacjach, w wielu zbiorach danych rozrzut nie jest stały, więc potrzebne są inne metody. Najczęściej spotykane sytuacje, w których założenie stałości warinacji jest złamane to:

- odpowiedź (zmienna odpowiedzi) jest proporcją (w przedziale $(0, 1)$), czyli frakcją sukcesów pewnego rodzaju; wariancja nie może być stała z powodu ograniczonego nośnika;
- zmienna odpowiedzi zlicza (sukcesy);
- zmienna odpowiedzi jest dodatnia (ciągła) lub ma inny, ale ograniczony (przynajmniej z jednej strony) nośnik.

W tych wszystkich przypadkach związek zmiennej odpowiedzi ze zmiennymi objaśniającymi zwykle nie jest liniowy.

1.1 Przykłady

Zbiór danych `lung capasity` dotyczących pojemności płuc (zmienna `fev`) w zależności od wieku, płci, palenia (zmienna binarna) oraz wzrostu. Dane znajdują się w pakiecie `GLMsData`.

```
> library(GLMsData) # Ładuje pakiet GLMsData
> data(lungcap) # Wczytuje dane
> head(lungcap)
Age FEV   Ht  Gender Smoke
3  1.072  46      F     0
4  0.839  48      F     0
4  1.102  48      F     0
4  1.389  48      F     0
4  1.577  49      F     0
4  1.418  49      F     0
```

Rozważać można wiele różnorakich modeli ze zmienną objaśnianą `fev`. Postać modelu, którą wybieramy często wynika z *wiedzy eksperckiej*. Oto niektóre z możliwości zawierające zmiennej: `fev (y)`, `age` ($x1$), `height` ($x2$), `gender` ($x3$), `smoking status` ($x4$):

$$\begin{array}{rcl}
\mu & = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_4 \\
\mu & = & \beta_0 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_4 \\
\mu & = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \\
\mu & = & \beta_0 + \beta_1 \log x_1 + \beta_2 x_2 + \beta_4 x_4 \\
\mu & = & \beta_0 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_4 \\
1/\mu & = & \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_4 \\
\log \mu & = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_4 x_4 \\
\mu & = & \beta_0 + \exp(\beta_1 x_1) - \exp(\beta_2 x_2) + \beta_4 x_2^4
\end{array}$$

Warto zastanowić się nad powodami, które mogłyby wskazać na słuszność poszczególnych propozycji. Innym przykładem wartym rozważenia są dane o zemdleniach pilotów militarnych `gforces` znajdujących się w tej samej bibliotece.

Wprowadzimy teraz oznaczenia na funkcję wiarogodności, log-wiarogodności i funkcję prawdopodobieństwa. Gęstość rozkładu lub funkcję prawdopodobieństwa oznaczać będziemy tym samym symbolem

$$P(y; \theta),$$

gdzie y jest zmienną (być może wektorem), a θ parametrem lub wektorem parametrów. W przypadku próby losowej piszemy

$$P(y_1, \ldots, y_n; \theta) = \prod_{i=1}^{n} P(y_i; \theta).$$

Funkcja wiarogodności jest funkcją parametrów rozkładu, a wartości $y_1, \ldots, y_n$ są pobierane z danej próby i pełnią rolę parametrów. Podkreślamy to w zapisie poprzez zmienienie kolejności argumentów:

$$L(\theta; y) = \prod_{i=1}^{n} P(y_i; \theta).$$

Funkcja log-wiarogodności

$$l(\theta; y) = \log L(\theta; y) = \sum_{i=1}^{n} \log P(y_i; \theta).$$

Argument, w którym funkcje L i l osiągają maksimum będziemy zwyczajowo oznaczać przez dodanie *daszka*, np. $\hat{\theta}$.

Przykład

Łatwo można pokazać, że metoda najmniejszych kwadratów (używana w regresji liniowej) jest specjalnym przypadkiem metody największej wiarogodności. Rozważmy model regresji liniowej: $Y_i \sim \mathbb{N}(\mu_i, \sigma^2)$ z nieznanym parametrem $\mu_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$. Po obliczeniu dostajemy

$$l(\beta_0, \ldots, \beta_p; \sigma^2, y) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu_i)^2 .$$

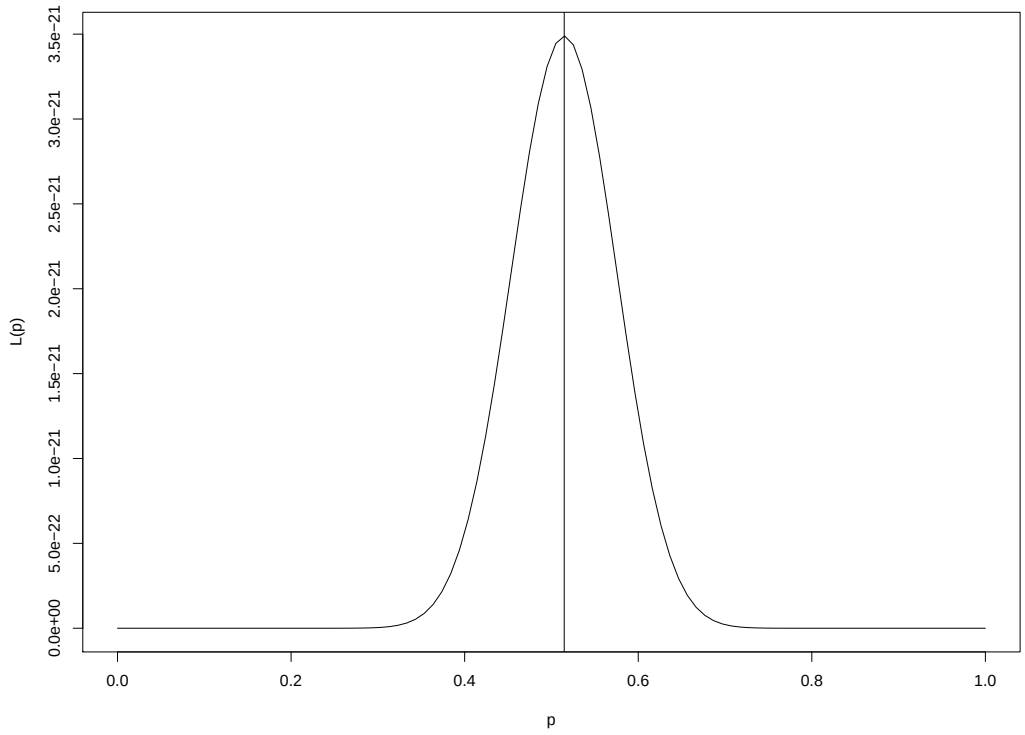
Wiarogodność zależy od $\beta_0, \ldots, \beta_p$ tylko przez sumę kwadratów różnic (σ^2 jest ustalone).

Rozważymy przykład.

```
library(GLMsData)
######### Wprowadzenie ##############
####      Maximum Likelihood for Estimating One Parameter
####      Przykład z [Dunn, Smyth 2018]
# SOI - standarized difference between the air pressure at Darwin and Tahiti
data(quilpie)
names(quilpie)
head(quilpie)


mu <- c(0.2, 0.4, 0.5, 0.6, 0.8) # Candidate valus to test
ll <- rep(0, 5)
for(i in 1:5)
  ll[i] <- sum(dbinom(quilpie$y, size =1, prob= mu[i], log=TRUE))
data.frame(Mu = mu, LogLikelihood = ll)
```

Zatem maksimum jest gdzieś pomiędzy 0.4 a 0.6, w okolicy 0.5.

```
x <- seq(0, 1, length = 100)
L = function(m){ sum(dbinom(quilpie$y, size = 1, prob = m, log = TRUE))}
mapply(L,c(0.4, 0.5))
plot(x, mapply(L, x), xlab = "p", ylab = " LogL(p)", type = "l")

L2 = function(m){ prod(dbinom(quilpie$y, size = 1, prob = m, log = FALSE))}
mapply(L,c(0.4, 0.5))
plot(x, mapply(L2, x), xlab = "p", ylab = "L(p)", type = "l")

muhat <- mean(quilpie$y); muhat
abline(v = muhat)          # pionowa linia w maksimum
```



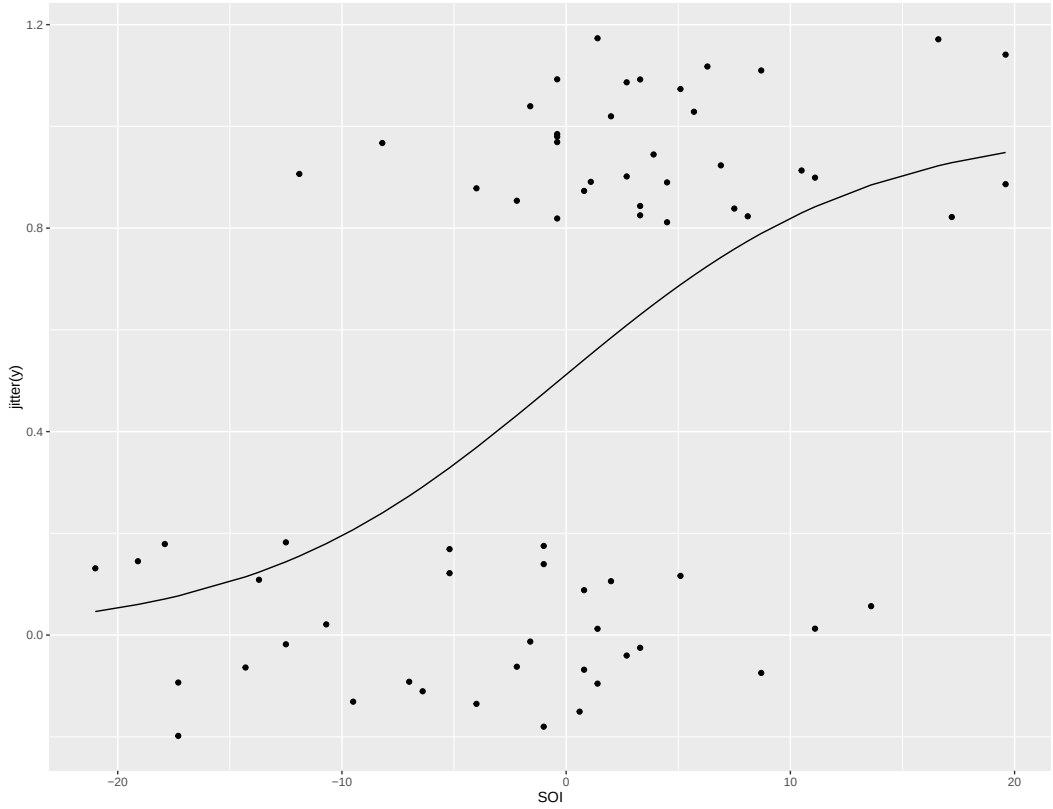
1.1.1 Zadanie: Iteracyjne szukanie maksimum MLE dla dwóch zmiennych objaśniających

```
# start assuming that there is no relationship between SOI and y
# $ B_0 = 0 B_1 = 0.5147
# Funckja obliczająca macierz informacji

MakeExpInf <- function(x, mu) {
  if (length(mu) == 1) mu <- rep(mu, dim(x)[1])
  mu <- as.vector(mu)
  return( t(x)%*% diag(mu*(1-mu))%*% x)
}

# Funkcja obliczająca mu:
MakeMu <- function(x, beta){
  eta <- x %*% beta
  return( 1/ (1+exp(-eta)))
}

# Funkcja obliczająca score vector
MakeScore <- function(x, y, beta) {
  mu <- MakeMu(x, beta)
  return( t(x) %*% (y-mu))
}

FitModelMLE <- function(y, x =NULL, maxits =8, add.constant = TRUE){
  if( is.null(x) ){
    allx <- cbind(Constant = rep(1, length(y)))}
  else{
    allx <- x
    if(add.constant){
      allx <- cbind(Constant = rep(1, length(y)), x)
    }
  }
```

```
  num.x.vars <- dim(allx)[2] - 1
  # Initials:
  beta <- c(mean(y), rep(0, num.x.vars))
  # Set up
  beta.vec <- array(dim = c(maxits, length(beta)))
  beta.vec[1,] <- beta
  mu <- MakeMu(allx, beta)
  score.vec <- MakeScore(allx, y, beta)
  inf.mat <- MakeExpInf(allx, mu)

  # Iterate to update
  for(i in (2:maxits)){
    beta <- beta + solve(inf.mat) %*% score.vec
    beta.vec[i,] <- beta
    mu <- MakeMu(allx, beta)
    score.vec <- MakeScore(allx, y, beta)
    inf.mat <- MakeExpInf(allx, mu)
  }
  # Compute log-likelihood
  LLH <- sum(y*log(mu) + (1-y)*log(1-mu) )
  return(list(coef = beta.vec[maxits, ],  # MLE of parameter estimates
              coef.vec = beta.vec,        # Estimates at each iteration
              LLH = LLH,                  # The maximum log-likelihood
              inf.mat = inf.mat,          # The information matrix
              score.vec = score.vec,      # The score vector
              mu = mu                     # The fitted values
  ))
}

m1.quilpie <- FitModelMLE(y = quilpie$y, x = quilpie$SOI)
```

Wykres modelu z danymi

```
# Wykres modelu z danymi
plot(jitter(y, 0.15)~ SOI, data = quilpie, pch = 19, axes = FALSE, las = 2,
            xlab = "July average SOI", ylab = "Rainfall exceeds threshold")
axis(side = 1, las = 2)
axis(side = 2, at=0:1, labels = c("NO", "Yes"), las = 2); box()

Y_fit <- 1/ (1+exp(-0.04813 - 0.1464 * quilpie$SOI))
library(ggplot2)
ggplot(quilpie, aes(SOI, jitter(y))) + geom_point()  + geom_line(aes(SOI, Y_fit))
```



Błędy estymowane

```
inf.mat.inverse <- solve(m1.quilpie$inf.mat)
inf.mat.inverse
std.errors <- sqrt(diag(inf.mat.inverse))
std.errors
```

1.2 Informacja: obserwowana i oczekiwana

Druga pochodna funkcji log-wiarogodności określa stopień stromości nachylenia. Im jest większa co do modułu, tym nachylenie funkcji l jest większe. Zatem wartość drugiej pochodnej w okolicy estymatora największej wiarogodności mówi o tym jak dobrze określony jest MLE. Mówi się tu o informacji obserwowanej. $J(\theta; y) = -\frac{d^2 l(\theta;y)}{d\theta^2} = -\frac{dU(\theta)}{d\theta}$. W przypadku wielowymiarowym jest to macierz. Definiujemy też informacje oczekiwaną

$$I(\theta) = \mathbb{E}(J(\theta; Y)).$$

Jest ona łatwiejsza do liczenia, nie zależy od y, jest dodatnia dla dowolnego argumentu (podczas gdy o J wiemy na pewno, że jest dodatnia tylko w $\theta = \hat{\theta}$).

Zadanie: błąd standardowy

Pokazać, że $I(\theta) = \mathbb{E}[U(\theta)] = \mathbb{V}\mathrm{ar}[U(\theta)]$.

Stosując rozwinięcie Taylora w punkcie $\theta = \hat{\theta}$ (patrz część teoretyczna kursu) otrzymujemy, że

$$\mathbb{V}\mathrm{ar}[\hat{\theta}] \approx 1/I(\theta).$$

Czyli odwrotność informacji obserwowanej aproksymuje błąd estymatora MLE.

W przypadku, gdy θ jest wektorem

$$\mathbb{V}\mathrm{ar}[\hat{\beta}_j] \approx 1/I_{jj}(\beta)$$

oraz

$$\mathrm{se}(\hat{\beta}_j) \approx 1/\sqrt{I_{jj}(\beta)}.$$

Przykład regresji logistycznej

W rozważanym przykładzie `Quilpie rainfall` MLE $\hat{\beta}$ jest rozwiązaniem układu równań

$$U(\hat{\beta}) = \begin{bmatrix} \sum_{i=1}^n (y_i - \hat{\mu}_i) \\ \sum_{i=1}^n (y_i - \hat{\mu}_i) x_i \end{bmatrix} = 0.$$

Tutaj $\log(\hat{\mu}/(1-\hat{\mu})) = X\underline{\hat{\beta}}$. Rozwiązanie nie jest trywialne. Przykład iteracyjnego rozwiązania zaimplementujemy (patrz kod wcześniej).

1.3 Przypomnienie własności MLE dla rodzin wykładniczych

1. MLE jest niezmiennicze ze względu na przekształcenia, tj. jeśli $s(\cdot)$ jest funkcją różnowartościową, to $s(\hat{\xi})$ jest MLE $s(\xi)$.

2. MLE jest asymptotycznie nieobciążony.

3. MLE jest ma asymtotycznie najmniejszą wariancję wśród estymatorów asymptotycznie nieobciążonych.

4. MLE jest zgodny, tj. zbiega do prawdziwej wartości (wg prawdopodobieństwa).

5. MLE jest asymptotycznie normalny, tzn. że jeśli ξ_0 jest prawdziwą wartością ξ, to

$$\hat{\xi} \sim \mathbb{N}_q \left(\xi_0,\, I(\xi_0)^{-1}\right),$$

gdy $n \to \infty$, gdzie q jest wymiarem wektora ξ. Równoważnie można napisać, że

$$(\hat{\xi} - \xi_0)^T I(\xi_0)(\hat{\xi} - \xi_0) \sim \chi^2_q,$$

gdy $n \to \infty$.

Ostatnia własność uzasadnia twierdzenia asymptotyczne o rozkładzie statystyk testowych.

1.4 Testowanie hipotez: duże próby

Krótko o ideach stojących za poszczególnymi testami hipotez przy dużych wielkościach prób. Hipotezą zerową będzie $H_0 : \xi = \xi^0$ dla pewnej postulowanej wartości ξ^0. Hipoteza alternatywna jest zaprzeczeniem zerowej: $H_1 : \xi \neq \xi^0$.

Omówimy trzy metody testowania H_0.

1. Test Walda (*Wald test*) oparty jest na odległości euklidesowej $\hat{\xi}$ od ξ^0:

$$W = \frac{(\hat{\xi} - \xi^0)^2}{\mathbb{V}\mathrm{ar}[\hat{\xi}]},$$

gdzie $\hat{\mathbb{V}\mathrm{ar}}[\hat{\xi}] = 1/I(\hat{\xi})$. Jeśli H_0 jest prawdziwe, to W asymptotycznie ma rozkład χ^2_1. Gdy testujemy hipotezę dotyczącą tylko jednego parametru, to możemy zamiast W rozważyć $Z := \sqrt{W}$, którę będzie miało rozkład $\mathbb{N}(0,1)$ asymptotycznie. Zauważmy, że znakowane Z pozwala badać również hipotezę alternatywną w postaci jednostronnej, tj. $H_1 : \xi > \xi^0$ lub $H_1 : \xi < \xi^0$.

2. *The score test* (test oceny) bada nachylenie log-warogodności w otoczeniu punktu ξ^0. Z definicji nachylenie w $\hat{\xi}$ jest równe zero, więc jeśli nachylenie w ξ^0 jest bliskie 0, to wnioskujemy, ze ξ^0 jest bliskie $\hat{\xi}$:

$$S = \frac{U(\xi^0)^2}{I(\xi^0)}.$$

Jeśli H_0 jest prawdziwe, to S asymptotycznie ma rozkład χ^2_1. Podobnie jak w przypadku W, małe wartości S świadczą na korzyść H_0. Warto zauważyć, że do obliczenia S nie jest potrzebne $\hat{\xi}$.

Gdy testujemy hipotezę dot. jednego parametru, to możemy rozważyć znakowany pierwiastek z S, który będzie miał asymptotycznie rozkład normalny.

3. The *likelihood ratio test* (test ilorazu funkcji wiarogodności), LLR, opiera się na odległości $l(\xi^0)$ od $l(\hat{\xi})$:

$$L = 2\left[l(\hat{\xi}; y) - l(\xi^0; y)\right].$$

Statystyka L asymptotycznie ma rozkłąd χ^2_1.

Wszystkie trzy testy asymptytcznie mają rozkład χ^2_1. Statystyki te są równoważne dla $n \to \infty$.

Testy dla przykładu

Dla danych z `Quilpie rainfall` i modelu opartego tylko o y (ignorujemy `SOI`), rozważmy hipotezę $H_0 : \mu = 0.5$. Przypomnijmy, że

$$U(\mu) = \frac{\sum_{i=1}^n y_i - n\mu}{\mu(1-\mu)},$$

$$I(\mu) = \frac{\mu(1-\mu)}{n},$$

$$W = \frac{(\hat{\mu} - \mu^0)^2}{\hat{\mu}(1-\hat{\mu})/n}.$$

Przy tym $\hat{\mu} = 0.5147$, $n = 48$.

```
muhat <- mean(quilpie$y)
mu0 <- 0.5
n <- length(quilpie$y)
varmu <- muhat*(1-muhat)/n
W <- (muhat - mu0)^2/varmu; W
[1] 0.05887446
```

Statystyka *score*

$$S = \frac{U(\mu^0)^2}{I(\mu^0)} = \frac{(n\hat{\mu} - n\mu^0)^2}{n\mu_0(1-\mu^0)}$$

```
S <- (muhat - mu0)^2/(mu0*(1-mu0)/n); S
[1] 0.05882353
```

Test ilorazu wiarogodności

```
Lmu0 <- sum( dbinom(quilpie$y, 1, mu0, log = TRUE) )
Lmuhat <- sum( dbinom(quilpie$y, 1, muhat, log=TRUE) )
L <- 2*(Lmuhat - Lmu0); L
[1] 0.05883201
```

Warto jeszcze odnieść się do odpowiednich p-wartości.

```
P.W <- pchisq(W, df=1, lower.tail=FALSE) #Wald
P.S <- pchisq(S, df=1, lower.tail=FALSE) #Score
P.L <- pchisq(L, df=1, lower.tail=FALSE) #Likelihood ratio
round(c(Wald = P.W, Score = P.S, LLR = P.L), 5)


 Wald    Score   LLR
0.8082  0.80837 0.80835
```

Wnioskujemy, że dane są zgodne z hipotezą zerową, że $\mu = 0.5$.

1.4.1 Testy dotyczące wektora ξ

Rozważmy hipotezę $H_0 : \xi = \xi^0$, gdzie ξ^0 jest postulowaną wartością wektora ξ. Defnicje poszczególnych statystyk testowych:

$$W = (\hat{\xi} - \xi^0)^T I(\hat{\xi})(\hat{\xi} - \xi^0); \tag{1.1}$$

$$S = U(\xi^0)^T I(\xi)^{-1} U(\xi^0); \tag{1.2}$$

$$L = 2\left[l(\hat{\xi}; y) - l(\xi^0; y)\right]. \tag{1.3}$$

Każda z nich ma asymptotycznie (z $n \to \infty$) rozkład $\chi^2(q)$, gdzie q jest długością wektora ξ.

Przykład Tym razem testowana będzie hipoteza $H_0 : \underline{\beta} = [0, 0]^T$.

```
##### Test Walda #####
m1.quilpie$coef
beta0 <-c(0,0); betahat <- m1.quilpie$coef
distance <- betahat - beta0
W.global <- t(distance) %*% m1.quilpie$inf.mat%*%distance
p.W.global <- pchisq(W.global, df = 2, lower.tail = FALSE)
round(c(W.global, P = p.W.global), 6)
```

Dane nie są zgodne z H_0.

```
#### Score Test ####
U <- MakeScore(cbind(1, quilpie$SOI), quilpie$y, beta0)
inf.mat.score <- MakeExpInf(cbind(1, quilpie$SOI), 0.5)
inf.mat.inverse <- solve(inf.mat.score)
S.global <- t(U) %*% inf.mat.inverse %*% U
p.S.global <- pchisq(S.global, df = 2, lower.tail =  FALSE)
round(c(score.stat=S.global, P=p.S.global), 6)
```

Dane nie są zgodne z H_0.

```
#### Test Likelihood Ratio ####
mu <- m1.quilpie$mu
Lbeta0 <- sum(dbinom(quilpie$y, 1, 0.5, log=TRUE))
Lbetahat <- sum(dbinom(quilpie$y, 1, mu, log=TRUE))
L.global <- 2*(Lbetahat - Lbeta0)
p.L.global <- pchisq(L.global, df=2, lower.tail=FALSE)
round(c(LLR.stat = L.global, P = p.L.global), 6)
```

Dane nie są zgodne z H_0.

Wartość p-wartości testu Walda jest ok. 10 razy większa od pozostałych.

1.4.2 Testowanie hipotez mówiących o pewnym podzbiorze parametrów

Nie sposób pominąć problemu testowania hipotez, które dotyczą tylko wybranych parametrów.

Podzielimy ξ na dwie części: ξ_1 długości q_1 i ξ_2 długości q_2, takie ze $\xi^T = [\xi_1^T, \xi_2^T]$, $q_1 + q_2 = q$. Hipoteza zerowa $H_0 : \xi_2 = \xi_2^0$ będzie testowana przeciwko dwustronnej alternatywie.

Dzielimy macierz informacji na bloki:

$$I(\hat{\xi}) = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix},$$

gdzie I_{11} jest maceirzą $q_1 \times q_1$, a I_{22} macierzą wymiaru $q_2 \times q_2$. Podobnie dzielimy macierz odwrotną odpowiednio oznaczając bloki

$$I(\hat{\xi})^{-1} = \begin{bmatrix} I^{11} & I^{12} \\ I^{21} & I^{22} \end{bmatrix},$$

gdzie $I^{22} = (I_{22} - I_{21}I_{11}^{-1}I_{12})^{-1}$ (sprawdzić dla $q_1 = q_2 = 1$. Niech ponadto $\xi^\star = [\hat{\xi}_1^t, \xi_2^{0T}]^T$. Wtedy

$$W = (\hat{\xi}_2 - \xi_2^0)^T (I^{22})^{-1} (\hat{\xi}_2 - \xi_2^0);$$
$$S = U(\xi^\star)^T I(\xi^\star)^{-1} U(\xi^\star);$$
$$L = 2[l(\hat{\xi}; y) - l(\xi^\star; y)].$$

Każda z nich ma asymptotycznie rozkład $\chi^2_{q_2}$.

Spróbuj sam!

Dla danych `Quilpie rainfall` zbadać hipotezę $H_0 : \beta_1 = 0$ w modelu $\mu = \beta_0 + \beta_1$`SOI`. Należy pamiętać, że trzeba również znaleźć $\xi^\star$. (Laboratoria).

Szczególnym przypadkiem jest ξ_2 długości jeden, $q_2 = 1$. Wtedy hipoteza zerowa przybiera często spotykaną postać

$$H_0 : \beta_j = \beta_j^0$$

dla pewnego $j \in \{1, 2, \ldots, q\}$. W tej sytuacji możemy stosować statystykę

$$Z = \frac{(\hat{\xi}_j - \xi_j^0)^2}{\sqrt{\mathbb{V}\mathrm{ar}(\hat{\xi}_j)}},$$

gdzie $Z \sim \mathbb{N}(0, 1)$ gdy $n \to \infty$.

Uwaga! Test Walda, najprostszy do przeprowadzenia, w pewnych sytaucja może być złudny. Zdarza sie, że W maleje, gdy różnica $\hat{\xi}_j - \xi_j$ rośnie(!). Jest to tzw. efekt Haucka Donnera.

1.4.3 Przedziały ufności testów

Teoretycznie przedział ufności jest łatwo wyznaczyć. Natomiast w praktyce, przedział można wyznaczyć *explicite* tylko dla testu Walda. W przypadku jednej zmiennej:

$$\hat{\xi}_j - z_{\alpha/2}\sqrt{\mathbb{V}\mathrm{ar}(\hat{\xi}_j)} < \xi_j < \hat{\xi}_j + z_{\alpha/2}\sqrt{\mathbb{V}\mathrm{ar}(\hat{\xi}_j)}.$$

Dla rozpatrywanych danych `Quilpie rainfall` w modelu ze zmienną objasniającą `SOI` znajdziemy przedział ufności dla β_1. Wartość log-wiarogodności w $\hat{\beta}_0$ i $\hat{\beta}_1$ wynosi $l(\hat{\beta}_0, \hat{\beta}_1; y) = -37.95$. Natomiast dla $\alpha = 5\%$ kwantyl $\chi^2_{1,1-\alpha} = 3.841$. Zatem granicami przedziału ufności są rozwiązania równania

$$2\left[-37.95 - l(\hat{\beta}_0, \beta_1; y)\right] = 3.841$$

ze względu na β_1.

Rozwiązując numerycznie dostajemy ostatni wiersz poniższej tabeli.

Rodzaj testu	Od	Do
Wald:	0.06238	0.2305
Score:	0.06552	0.2289
LLR:	0.07191	0.2425

Zauważyć które przedziały są symetryczne względem $\hat{\beta}_1 = 0.1464$ oraz że przedział testu Score zawiera się w przedziale testu Walda.

1.5 Diagnostyka dopasowania

Wyobraźmy sobie, że wyestymowaliśmy $\underline{\beta}$, czyli znaleźliśmy wektor $\underline{\hat{\beta}}$. Innymi słowy, dopasowaliśmy model. Jak zmierzyć jakość dopasowania tego modelu? Potrzebujemy jakiejś (dobrej!) miary rozrzutu pomiędzy obserwacjami i dopasowanym modelem. Tutaj w lub w_i oznacza wagę obserwacji (*prior weights*). Jeśli nie przyjmujemy inaczej, to jest ona równa 1.

- Używana w klasie regresji liniowej $\sum(y_i - \hat{\pi}_i)^2$ nie jest właściwą miarą dopasowania, bo nie ma sensu dla rozkładu, który nie jest symetryczny, ani dla odpowiedzi binarnej. (Dodatkowo zakłada jednorodność wariancji, a w rodzinach wykładniczych (generalnie) wariancja zależy od średniej.)

- **Rezydua Pearsona**: problemu ze zmienną wariancją pozbywamy się wydzielając przez nią (ale jest to tylko przybliżenie!)

$$r_P = \frac{y - \hat{\mu}}{\sqrt{V(\hat{\mu})/w}}.$$

  `resid(fit, type = "pearson")`.

- Dewiancja (ang. *deviance*) – metoda powszechnie stosowana w przypadku estymacji parametrów za pomocą MLE (czyli tak jak jest w GLM).

Dewiancja

Dewiancja modelu `modl` z estymatorem $\hat{\pi}$ wynosi

$$D_{\texttt{modl}} = 2\left[l(\texttt{perfect fit}) - l(\texttt{modl})\right].$$

Inaczej

$$D(\underline{\mathbf{y}}, \hat{\pi}) = 2\left[l(\underline{\mathbf{y}}, \underline{\mathbf{y}}) - l(\hat{\pi}, \underline{\mathbf{y}})\right].$$

Dewiancja (w pewnym sensie) porównuje nasz dopasowany model z doskonałym dopasowaniem. *Rezyduami dewiancyjnymi* nazywamy znakowany pierwiastek z dewiancji dla jednej obserwacji:

$$r_D = \text{sign}(y - \hat{\mu})\sqrt{wD(y, \hat{\mu})}.$$

W `R` są to rezydua zwracane w domyślnym trybie przez funkcję `resid()`. Jeśli spełniony jest warunek $\tau \leq 1/3$, to rezydua dewiancyjne mają asymptotycznie rozkład normalny.

Saturacja: model saturowany/ model wysycony ma miejsce, gdy obserwacji jest tyle ile estymowanych parametrów $(p+1)$, o ile macierz objaśniająca jest pełnego rzędu (w przeciwnym wypadku nie da się dopasować modelu!). Wówczas $\hat{\pi} = \underline{\mathbf{y}}$ (to powinno być jasne).

Przy pewnych założeniach statystyki związane z dewiancją i rezyduami Pearsona mają asymptotycznie rozkład χ^2. Jak podaje **?** warunkiem wystarczającym jest w przypadku dewiancji, aby:

$$\tau := \frac{\phi V(y)}{(y - b_y)^2} \leq \frac{1}{3}.$$

A w przypadku statystyki Pearsona, aby:

$$\tau \leq \frac{1}{5}.$$

Co te warunki oznaczają dla konkretnych modeli, powiemy później.

1.5.1 Inne sposoby badania jakości dopasowania modelu logistycznego

W klasycznej regresji liniowej rozważaliśmy

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{y_i - \bar{y}_i},$$

gdzie $\bar{y}_i$ – średnia wartość odpowiedzi. To był procent wariancji wyjaśnionej przez model.

1.6 Przykładowe zadania

1. Korzystając z funkcji `rexp()` wygenerować $n = 100$ losowych wartości z roskładu wykładniczego z $\mu = 1$.

- Naszkicować wykres funkcji wiarogodności dla μ od 0.75 do 1.25. Dodać pionowe proste wskazujące $\hat{\mu}$ i $\mu^0 = 1$.
- Przetestować hipotezę $H_0 : \mu = 1$ korzystając z testów Walda, Score i LLR.
- Stworzyć wykres statystyk Walda, Score i LLR względem μ. Poziomą prostą wskazać wartości krytyczne χ^2_1. Porównać wartości statystyk testowych dla różnych wartości $\hat{\mu}$.
- Znaleźć 95% przedział ufności dla statystyki Walda.

2. Rozważmy problem esymacji średniej dla próby $y_1, \ldots, y_n$ z rozkładu normalnego o znanej wariancji σ^2.
   - Znajdź funckję wiarogodności i log-wiarogodności.
   - Znajdź score function.
   - Korzystając ze score function znajdź MLE μ.
   - Znajdź informację obserwowaną i oczekiwaną dot. μ.
   - Znajdź błąd standardowy $\hat{\mu}$.
   - Sprawdź prawdziwość hipotezy $H_0 : \mu = 0$ za pomocą testów Walda, Score i LLR.
   - Pokaż, że w tym przypadku $S = W = L$.

3. Dla rozkładu wykładniczego

$$P(y; \mu) = \exp(-y/\mu)/\mu,$$

   dla $\mu > 0$ i $y > 0$ estymujemy średnią μ na podstawie próby $y_1, \ldots, y_n$.
   - Znajdź funckję wiarogodności i log-wiarogodności.
   - Znajdź score function.
   - Korzystając ze score function znajdź MLE μ.
   - Znajdź informację obserwowaną i oczekiwaną dot. μ.
   - Pokaż, że błąd standardowy $\hat{\mu}$ to $\text{se}(\hat{\mu}) = \hat{\mu}/\sqrt{n}$. (Uwaga na zapis!)
   - Pokaż, że dla $H_0 : \mu = 1$: $W = (\hat{\mu}-1)^2/(\hat{\mu}^2/n)$, $S = n(\hat{\mu}-1)^2$ i $L = 2n(\hat{\mu} - \log\hat{\mu} - 1)$.
   - Sporządź wykresy W, S i L dla μ między 0.5 a 2 dla $n = 10$ oraz dla $n = 100$. Skomentuj.

Rozdział 2

Modelowanie proporcji – model logistyczny

2.1 Przypomnienie

Dane binarne to dane, w których zmienna odpowiedzi jest dwuwartościowa. Sukces bądź porażka, kodowane zwykle za pomocą 1 i 0. Spójrzmy na losową składową GLM dla takich danych.

Niech Y będzie binarną zmienną odpowiedzi.

Niech $\pi := \mu := \mathbb{P}(Y = 1)$, $1 - \mu = \mathbb{P}(Y = 0)$. Zauważmy, że $\mu = \mathbb{E}Y$.

Inaczej: $P(y, \mu) = \mu^y (1 - \mu)^{1-y}$ – rozkład zadany przez 1 parametr $\mu \in (0, 1)$.

Wówczas $\mathbb{E}Y = \mu$, $\mathbb{V}\mathrm{ar}(Y) = \mu(1 - \mu)$.

2.2 Dane zgrupowane – model dwumianowy

Przykład

Mamy dziesięciu pacjentów z 2 różnych szpitali (X_1), których poddajemy dwóm różnym terapiom (X_2). Wynik leczenia kodujemy za pomocą zmiennej Y. Zatem $\{y_i = 1\}$ oznacza, że $i$ty pacjent wyzdrowiał, a $\{y_i = 0\}$ oznacza, że nie wyzdrowiał.

i	Y	X_1	X_2
1	0	1	1
2	0	1	2
3	0	2	1
4	0	1	2
5	1	1	2
6	1	1	1
7	1	1	2
8	1	2	2
9	1	1	2
10	1	2	2

Dane te możemy zapisać w mniejszej tabeli poprzez zgrupowanie:

i	m_i	Z_i	F_i	X_1	X_2
1	2	1	0.5	1	1
2	5	3	0.6	1	2
3	1	0	0	2	1
4	2	2	1	2	2

Spróbujmy wykonać operację grupowania danych w `R`. Dane `lizardRaw` z pakietu `gRim` dotycza zwyczajów budowy gniazd przez jaszczurki dwóch gatunków (dist, anoli).

```
> data(lizardRAW)
> head(lizardRAW)
   diam height species
1   >4  >4.75    dist
2   >4  >4.75    dist
3  <=4 <=4.75   anoli
4   >4 <=4.75   anoli
5   >4 <=4.75    dist
6  <=4 <=4.75   anoli
```

Teraz agregujemy dane, aby dla każdego różnego wektora wartości był dokładnie jeden wiersz w tabeli:

```
> as.data.frame(ftable(lizardRAW))
   diam height species Freq
1  <=4 <=4.75   anoli   86
2   >4 <=4.75   anoli   35
3  <=4  >4.75   anoli   32
4   >4  >4.75   anoli   11
5  <=4 <=4.75    dist   73
6   >4 <=4.75    dist   70
7  <=4  >4.75    dist   61
8   >4  >4.75    dist   41
```

2.3 Dopasowanie modelu – ocena jakości

2.3.1 Dewiancja dla proporcji

Twierdzenie 1.

$$\mathcal{D}(\underline{\mathbf{y}}, \hat{\mu}) = -2\sum_{i=1}^{n} n_i \left[\bar{y}_i \log\left(\frac{\bar{y}_i}{\hat{\mu}_i}\right) + (1-\bar{y}_i)\log\left(\frac{1-\bar{y}_i}{1-\hat{\mu}_i}\right)\right].$$

Gdy $n_i = 1$ dla każdego i, to:

$$\mathcal{D}(\underline{\mathbf{y}}, \hat{\mu}) = -2\sum_{i=1}^{n} \log\left(1 - |y_i - \hat{\pi}_i|\right), \quad D(\underline{\mathbf{y}}, \hat{\mu}) \geq 0$$

i równość zachodzi tylko gdy $\hat{\pi} = \underline{\mathbf{y}}$.

Dowód. Dowód przeprowadzamy dla $n_i = 1$. Przypadek pozostały jest analogiczny.
Zauważamy, że w modelu wysyconym logwiarogodność wynosi 0:

$$l(\underline{\mathbf{y}}, \underline{\mathbf{y}}) = 0,$$

ponieważ $l(y_i, y_i) = y_i \log y_i + (1 - y_i) \log(1 - y_i) = 0$.
Zatem

$$\begin{aligned} D(\underline{\mathbf{y}}, \hat{\pi}) = & \quad -2l(\hat{\pi}, \underline{\mathbf{y}}) = -2 \textstyle\sum_{i=1}^{n} \left[y_i \log(\hat{\pi}_i) + (1 - y_i) \log(1 - \hat{\pi}_i) \right] \\ = & \quad -2 \textstyle\sum_{i=1}^{n} \left[\log \hat{\pi}_i I(y_i = 1) + \log(1 - \hat{\pi}_i) I(y_i = 0) \right] \\ = & \quad -2 \textstyle\sum_{i=1}^{n} \log \left(1 - |y_i - \hat{\pi}_i| \right). \end{aligned}$$

Żeby powyższe wyrażenie było równe 0, to koniecznie jest, aby każdy element był równy 0, a więc $|y_i - \hat{\pi}_i| = 0$. ▲

Warto mieć świadomość pomiędzy charakterem danych indywidualnych i zgrupowanych:

- konstrukcja modelu wysyconego: dla indywidualnych danych dopasowuje się dokładnie do $y_1, \ldots, y_n$, natomiast dla danych grupowych – do proporcji: $\bar{y}_1, \ldots, \bar{y}_n$.
- w pierwszym przypadku osobny estymator $\hat{\pi}_i$ dla każdej obserwacji, a w drugim dla każdej grupy.

2.3.2 Dewiancja jako statystyka testowa dopuszczalności modelu

Warunek $\tau \leq 1/3$ oznacza, że $n_i y_i > 3$ i $\mu_i(1 - y_i) \geq 3$ dla każdego i. W przypadku danych binarnych ($n_i = 1$ dla każdego i) $\mathcal{D}(\underline{\mathbf{y}}, \hat{\mu})$ nie ma asymptotycznie (przy $n \to \infty$) rozkładu χ^2 (bo stopnie swobody zwiększają się wraz z n).

Asymptotyczne zachowanie dewiancji

Dla danych z rozkładu dwumianowego $\mathcal{D}(\underline{\mathbf{y}}, \hat{\mu})$ ma asymptotycznie rozkład χ^2 o $N - \dim(\underline{\mathbf{x}})$ stopniach swobody. Innymi słowy, niech N będzie liczbą grup, $n_i \to \infty$ – licznością $i$tej grupy $i = 1, 2, \ldots, N$. Wtedy, przy prawdziwości $H_0 : \{y_i$ jest obserwacją z rozkładu $\text{bn}(n_i, \mu), i = 1, 2, \ldots, N\}$, zachodzi

$$\mathcal{D}(\bar{\underline{\mathbf{y}}}, \hat{\mu}) \to \mathcal{D} \sim \chi^2_{(N-(p+1))}.$$

Zwykle uważa się, że dopasowanie modelu jest **niedostateczne** jeśli $\mathcal{D}(\bar{\underline{\mathbf{y}}}, \hat{\mu}) > z_{1-\alpha}$ – kwantyl rzędu $(1 - \alpha)$ z rozkładu $\chi^2(N - (p + 1))$, gdzie α jest ustalonym przez nas poziomem istotności (np. $\alpha = 0.05$).

Procent dewiancji objaśnionej przez model. Niech l_m – log-wiarogodność analizowanego modelu, l_s, l_0 – analogi dla modelu wysyconego (saturated) i minimalnego (null). Oczywiście $l_0 \leq l_m \leq l_s$. Procent dewiancji objaśnionej:

$$K := \frac{l_m - l_0}{l_s - l_0} \in [0, 1].$$

$K \approx 0$ – badany model nie poprawia dopasowania w stosunku do modelu minimalnego.
$K \approx 1$ – badany model dopasowuje się tak dobrze jak wysycony.

2.3.3 Statystyka Pearsona – alternatywa dla dewiancji

$$\chi^2_{\mathrm{P}} = \sum_{i=1}^{N} n_i \frac{(\bar{y}_i - \hat{\pi}_i)^2}{\hat{\pi}_i(1 - \hat{\pi}_i)},$$

gdzie $\hat{\pi}_i = h(\underline{\mathbf{x}}_i^T \hat{\underline{\beta}})$. Gdy N jest ustalone i $n_i \to \infty$ dla $i = 1, 2, \ldots, N$, to χ^2_{P} asymptotycznie ma rozkład χ^2 z $N - \dim(\underline{\mathbf{x}})$ stopniami swobody.

Uwaga 1. *Rezydua Pearsona*

$$r_P(y_i, \hat{\pi}_i) := \frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1 - \hat{\pi}_i)/n_i}}$$

nie mają jednostkowej wariancji, ale

$$\mathbb{V}\mathrm{ar}(r_P(Y_i, \hat{\pi}_i)|\hat{\pi}_i) = 1.$$

Rezydua dewiancyjne

$$r_{\mathcal{D}}(\bar{y}_i, \hat{\pi}_i) = \mathrm{sign}(\bar{y}_i - \hat{\pi}_i)\sqrt{n_i \left[\bar{y}_i \log \frac{\bar{y}_i}{\hat{\pi}_i} + (1 - \bar{y}_i) \log \frac{1 - \bar{y}_i}{1 - \hat{\pi}_i}\right]}$$

Funkcja signum na początku zapewnia, że znak $r_{\mathcal{D}}$ jest taki sam jak znak r_p – rezyduum Pearsona. Zauważmy, że

$$\mathcal{D}(\underline{\mathbf{y}}, \hat{\pi}) = \sum_i r^2_{\mathcal{D}}(\bar{y}_i, \hat{\pi}_i).$$

Znamy asymptotyczne zachowanie dewiancji. Jednak mogą pojawić się sytuacje, w których przedstawione wyżej statystyki nie są użyteczne, tj. gdy nasze grupy są małych liczności. Innym problemem jest niejednostkowa wariancja r_P i r_D. Radzimy sobie z tym poprzez **standaryzację**.

Do *przybliżonej* estymacji wariancji używa się tzw. macierzy daszkowej (*hat matrix*)

$$\hat{H} := (W^{1/2})^T X (X^T W X)^{-1} X^T W^{1/2},$$

gdzie $W = \mathrm{diag}(n_i \hat{\pi}_i(1 - \hat{\pi}_i))$. Oznaczmy $\hat{H} = [h_{ij}]_{i,j=1,\ldots,N}$.

Standaryzowane rezydua dewiancyjne

$$r_{\mathcal{D},s}(\bar{y}_i, \hat{\pi}_i) = \frac{r_{\mathcal{D}}(\bar{y}_i, \hat{\pi}_i)}{\sqrt{1 - h_{ii}}},$$

gdzie h_{ii} jest elementem macierzy

Standaryzowane rezydua Pearsona

$$r_{P,s}(\bar{y}_i, \hat{\pi}_i) = \frac{r_{P}(\bar{y}_i, \hat{\pi}_i)}{\sqrt{1 - h_{ii}}},$$

gdzie h_{ii} jest elementem macierzy

2.3.4 Wpływ $i$tej obserwacji na estymatory

Zdarza się, że w danych występują obserwacje odstające, które powodują znaczne przesunięcie estymatorów w pewnym kierunku. Jest to sytuacja niepożądana, ponieważ zbyt duży wpływ obserwacji nietypowych oddala estymatory od prawdziwych wartości estymowanych. Jedną z miar wpływu danej obserwacji jest **odległość Cooka**. Oczywiście, nie każda wpływowa obserwacja musi być odstająca (!).

Odległość Cook'a $i$tej obserwacji definiujemy następująco

$$c_{(i)} := (\hat{\underline{\beta}}_{(i)} - \hat{\underline{\beta}})^T X^T W X (\hat{\underline{\beta}}_{(i)} - \hat{\underline{\beta}}),$$

gdzie $W = \mathrm{diag}(\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n))$, $\hat{\underline{\beta}}_{(i)}$ - estymator MLE pochodzący z dopasowania modelu do danych bez i-tej obserwacji.

2.4 Porównywanie modeli

W przypadku modeli zagnieżdżone możemy korzystać z dewiancji.

2.4.1 Modele niezagnieżdżone

Porównujemy satystyki dopasowania poszczególnych modeli, ale należy zwrócić uwagę na liczbę parametrów w porównywanych modelach.

1. Jeśli modele mają jednakową liczbę parametrów, to można obliczyć dla nich osobno statystyki dopasowania i je porównać;

2. Jeśli nie, tzn jest inna liczba parametrów, to statystyki dopasowania *faworyzują* model o większej liczbie parametrów;

Stąd wprowadzane są kary, np.:

$$\text{AIC} = -2l(\hat{\underline{\beta}}) + 2 \cdot (\#\text{dop. parametrów}),$$

$$\text{BIC} = -2l(\hat{\underline{\beta}}) + \log(n)(\#\text{dop. parametrów}).$$

2.5 Przykładowe zadania

1. Dopasować do danych ze zbioru **bliss**

conc	dead	number
0	2	30
1	8	30
2	15	30
3	23	30
4	27	30

   model logistyczny $y \sim$ conc.

   Utworzyć rozwiniętą kopię zbioru **bliss** (w postaci danych niegrupowanych), na przykład za pomocą instrukcji `rep`. Dopasować model logistyczny. Porównać współczynniki z uzyskanymi w poprzednim punkcie.

2. Dopasować model logistyczny do następujących danych:

y	x
1	1
1	2
1	3
0	4
0	5
0	6

   Wyjaśnić przyczynę dużej liczby iteracji w algorytmie Fisher Scoring podawanej przez R.

3. Zbiór **malaria** zawiera informacje na temat liczby osób posiadających przeciwciała (*Sposi-tive*) pośród wszystkich badanych osób (*Number*) w danej grupie wiekowej (*Age*). (Przeciwciała produkowane przez organizm jako ochrona przed malarią pozostają w organizmie także po wyzdrowieniu i są wykrywane przez test serologiczny – osoby z przeciwciałami mają dodatni wynik testu serologicznego.)

1. Dopasować model regresji logistycznej używając wieku jako jedynej zmiennej objaśniającej.

2. Dopasować model regresji liniowej dla logitów proporcji z wagami $n(proporcja)(1 - proporcja)$. Porównać wyniki tego modelu i modelu z poprzedniego punktu.

3. Używając modelu logistycznego, oszacować wiek, dla którego prawdopodobieństwo dodatniego odczynu wynosi 1/4.

4. Skonstruować przedział ufności dla prawdopodobieństwa dodatniego odczynu w wieku 20 lat.

5. Narysować wykres frakcji przypadków dodatniego odczynu serologicznego w zależności od wieku wraz z dopasowaną krzywą.

4. Zbiór **finance** zawiera dane dotyczące kondycji finansowej 46 przedsiębiorstw na podstawie czterech wskaźników finansowych.

1. Dopasować model logistyczny. Przetestować hipotezę, że zbiór zawiera zmienne istotne i obliczyć procent dewiacji wyjaśnianej przez model.

2. Za pomocą instrukcji 'drop1' dokonać sekwencyjnego usunięcia z modelu nieistotnych zmiennych. Porównać mniejszy model z modelem wyjściowym. Obliczyć procent dewiacji wyjaśnianej.

3. Za pomocą instrukcji'step' dokonać sekwencyjnego usunięcia z modelu nieistotnych zmiennych. Porównać mniejszy model z modelem wyjściowym. Obliczyć procent dewiacji wyjaśnianej.

4. Rozpatrzyć rezydua oparte na dewiacjach. Wyliczyć standaryzowane rezydua i narysować ich wykres kwantylowy.

5. Wyrzucić obserwacje potencjalnie odstające, dopasować powtórnie model i obliczyć dla niego procent dewiacji wyjaśnionej.

5. (Test Hosmera-Lemeshowa) Zbiór **HosLemData** zawiera zmienną objaśnianą y i zmienną objaśniającą x.

1. Dopasować model regresji logistycznej do danych z **HosLemData** i zastanowić się nad możliwością zbadania jakości dopasowania za pomocą testu opartego na dewiancjach i testu Pearsona.

2. Zaimplementować test Hosmera-Lemeshowa z liczbą grup $g = 10$. Co z niego wynika?

3. Przeprowadzić na tych samych danych testy Hosmera-Lemeshowa z $g = 9$, $g = 11$ i $g = 12$. Co z nich wynika?

Rozdział 3

Zliczanie

3.1 Regresja Poissonowska

Przypomnijmy, że założenia modelu Poissona są następujące:

(a) Nie ma górnego ograniczenia na liczbę tego co zliczamy lub ograniczenie to jest bardzo duże (np. ograniczeniem na liczbę zachorowań jest liczba wszystkich osób w danej populacji);

(b) Zliczane zdarzenia są (lub mogą być uznane za) niezależne.

Wprowadźmy oznaczenie $P(y, \mu) := \mathbb{P}(Y = y)$, jeśli $Y \sim \text{Pois}(\mu)$.

Model regresji poissonowskiej:

Niech:

- $(y_i, \underline{\mathbf{x}}_i)_{i=1,\ldots,n}$ – n niezależnych obserwacji,
- $\mu_i := \mathbb{E}(Y_i|x_i)$,
- $Y_i|x_i \sim \text{Pois}(\mu_i)$ – składowa losowa modelu.

Składowa systemowa (czyli dalsze założenia modelu):

$$\mu_i = \exp(\underline{\mathbf{x}}_i^T \underline{\beta}) \Leftrightarrow \log \mu_i = \underline{\mathbf{x}}_i^T \underline{\beta}.$$

Zauważmy, że $\mu_i > 0$ dla każdego $i = 1, \ldots, n$.

Ważną zaletą regresji poissonowskiej z logarytmiczną funkcją łączącą jest łatwość interpretacji współczynników. Dzięki temu lepiej rozumiemy, co oznaczają konkretne wartości estymatorów. Spójrzmy:

$$\mu(\underline{\mathbf{x}}) = \exp(\underline{\mathbf{x}}^T \underline{\beta}) = e^{\beta_0} \cdot e^{x_1\beta_1} \cdot \ldots \cdot e^{x_p\beta_p}$$

czyli β_j ma multiplikatywny wpływ na μ:

$$\frac{\mu(x_1,\ldots,x_{j-1},x_j+1,x_{j+1},\ldots,x_p)}{\mu(x_1,\ldots,x_{j-1},x_j,x_{j+1},\ldots,x_p)} = e^{\beta_j}$$

albo

$$\log\mu(x_1,\ldots,x_{j-1},x_j+1,x_{j+1},\ldots,x_p) - \log\mu(x_1,\ldots,x_{j-1},x_j,x_{j+1},\ldots,x_p) = \beta_j.$$

Widzimy, że zwiększenie wartości $j$tej zmiennej objaśniającej o 1 powoduje zmianę $\mu(\underline{\mathbf{x}})$ e^{β_j} razy.

Inne możliwe funkcje łączące dla rozkładu Poissona:

- $h(x) = x^2$. Wtedy $\sqrt{\mu} = \underline{\mathbf{x}}_i^T\underline{\beta}$. Jednak $\mu(\underline{\mathbf{x}}) = (\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)^2$ jest trudne w interpretacji!
- $h(x) = |x|$, czyli $\mu(\underline{\mathbf{x}})) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p|$. Tutaj problemem jest nieróżniczkowalność w 0.

Będziemy analizować przykład pochodzący z [2018GLM 10.6]. W badaniu nad zachowaniem skrzypłoczy (zwanych też mieczogonami, a po angielsku *horsehoe crab*) (*Limulus polyphemus*) [1] próbowano zaobserwować liczbę samców (`Sat`) tego gatunku 'krążących' wokół nie swojej samicy w zależności od jej cech: koloru `Col`, wagi `Wt`, stanu jej szkieletu/kręgosłupa `Spine` i szerokości pancerza `Width`. Badacze chcieli odpowiedzieć na pytanie: jakie czynniki wpływają na atrakcyjność samicy w oczach 'skrzypłoczy–satelit'?

```
> library(GLMsData)
> data(hcrabs)
> head(hcrabs)
  Col  Spine Width Sat   Wt
1   M NoneOK  28.3   8 3050
2  DM NoneOK  22.5   0 1550
3  LM BothOK  26.0   9 2300
4  DM NoneOK  24.8   0 2100
5  DM NoneOK  26.0   4 2600
6   M NoneOK  23.8   0 2100
```

Wartości przyjmowane przez zmienną `col` (kolor): L – light, LM – medium light, M – medium, DM – dar medium, D – dark powinny być zrozumiałe. Zmienna `Spine` mówi ile stron szkieletu skrzypłocza jest nienaruszonych (jedna – `OneOK`, dwie – `BothOK` lub zero – `NoneOK`). Te zmienne są uporządkowane w naturalny sposób, więc tak je będziemy reprezentować:

[1] Co ciekawe zwierzęta te nie są krabami, mają niebieską krew i dla tej krwi, wykorzystywanej w medycynie, są one zbierane przez ludzi. Po pobraniu krwi są z powrotem wrzucane do oceanu.

```
> hcrabs$Col <- ordered(hcrabs$Col, levels = c("LM", "M", "DM", "D"))
> hcrabs$Spine <- ordered(hcrabs$Spine, levels = c("NoneOK", "OneOK","BothOK"))
```

Analizę rozpoczniemy od obejrzenia wykresów zależności liczby `Sat` od poszczególnych cech samicy.

```
with(hcrabs,{
  logSat <- log(Sat+1)
  plot( jitter(Sat) ~ Wt, ylab="Sat", las=1)
  plot( jitter(logSat) ~ log(Wt), ylab="log(Sat+1)", las=1)
  plot( logSat ~ Col, ylab="log(Sat+1)", las=1)
  plot( jitter(Sat) ~ Width, ylab="Sat", las=1)
  plot( jitter(logSat) ~ log(Width), ylab="log(Sat+1)", las=1)
  plot( logSat ~ Spine, ylab="log(Sat+1)", las=1)
})
```

Pierwszy wykres – zależność od wagi (`Wt`) nie wskazuje na duże zróżnicowanie. Odrobinę więcej widać w przeskalowaniu logarytmicznym: można mieć wrażenie, że cięższe samice mają trochę więcej satelitów. Z kolei jaśniejszy kolor wydaje się być dużo atrakcyjniejszy (trzeci wykres). Szerokość pancerza (`Width`) też trochę zwiększa liczbę potencjalnych partnerów. Ostatni z wykresów pokazuje wykresy pudełkowe liczby satelitów względem stanu szkieletu. Trudno jest wyciągnąć wnioski.

Musimy pamiętać o związkach między poszczególnymi cechami:

```
> with(hcrabs,{
+   plot( log(Wt) ~ log(Width), las=1 )  # praktycznie liniowy związek
+   plot( log(Wt) ~ Col, las=1 )         # cięższe są jaśniejsze albo
                                         # jaśniejsze są cięższe (średnio)
+   plot( log(Wt) ~ Spine, las=1 )       # bez uszkodzeń są cięższe, ale
                                         # z jednym są najlżejsze (średnio)
+ })
```

```
> coef(lm( log(Wt) ~ log(Width), data=hcrabs ))
(Intercept) log(Width)
 -0.60        2.56
```

Więc `WT = e^{-0.6}\cdot (Width)^{2.56}`. Całkiem zgodne z fizyką.

Czy spełnione są założenia modelu Poissona? Zmienna `Sat` jest zliczaniem, wartości nie są ograniczone. Z drugiej strony mieczogony są zwierzętami stadnymi i jako taki, mogą zachowywać się zgodnie z zasadą *rich get richer*, a więc nie niezależnie. Jeśli tak, to można to próbować objąć modelem Poissona z nadwyżką rozproszenia (overdispersion), który omówiony był w Teoretycznym Kursie GLMów.

```
> # dopasowanie zywkłego modelu Poissona
> crabs.mP <- glm(Sat ~ log(Wt) + log(Width) + Spine + Col,
+              family=poisson, data=hcrabs)
> summary(crabs.mP)

Call:
glm(formula = Sat ~ log(Wt) + log(Width) + Spine + Col, family = poisson,
    data = hcrabs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1062  -1.8222  -0.4923   0.8702   4.8754

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.16015    1.99295  -5.600 2.15e-08 ***
log(Wt)       1.75627    0.46757   3.756 0.000173 ***
log(Width)   -0.47766    1.34521  -0.355 0.722526
Spine.L      -0.03786    0.08272  -0.458 0.647207
Spine.Q       0.15293    0.16125   0.948 0.342936
Col.L        -0.35880    0.15587  -2.302 0.021345 *
Col.Q         0.10484    0.12284   0.853 0.393407
Col.C         0.05789    0.09123   0.635 0.525714
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 540.95  on 165  degrees of freedom
AIC: 912.24

Number of Fisher Scoring iterations: 6
```

Dopasowanie modelu quasipoissona

```
> crabs.mqP <- glm(Sat ~ log(Wt) + log(Width) + Spine + Col,
+                  family=quasipoisson, data=hcrabs)
> summary(crabs.mqP)

Call:
glm(formula = Sat ~ log(Wt) + log(Width) + Spine + Col, family = quasipoisson,
    data = hcrabs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1062  -1.8222  -0.4923   0.8702   4.8754

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -11.16015    3.56504  -3.130  0.00206 **
log(Wt)       1.75627    0.83641   2.100  0.03727 *
log(Width)   -0.47766    2.40635  -0.199  0.84290
Spine.L      -0.03786    0.14797  -0.256  0.79840
Spine.Q       0.15293    0.28845   0.530  0.59671
Col.L        -0.35880    0.27883  -1.287  0.19997
Col.Q         0.10484    0.21974   0.477  0.63392
Col.C         0.05789    0.16319   0.355  0.72324
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for quasipoisson family taken to be 3.199899)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 540.95  on 165  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

Oczywiście estymowane parametry są w obu modelach takie same (zgodnie ze znaną teorią). Estymator rozproszenia w przybliżeniu $\hat{\varphi} = 3.20$.

Przeprowadzamy test F (który jest odporny na nadwyżkę rozproszenia)

```
> anova(crabs.mqP, test="F")
Analysis of Deviance Table
Model: quasipoisson, link: log
Response: Sat
Terms added sequentially (first to last)
           Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                         172     632.79
log(Wt)     1   83.084       171     549.71 25.9645 9.429e-07 ***
log(Width)  1    0.007       170     549.70  0.0023    0.9621
Spine       2    1.125       168     548.58  0.1758    0.8389
Col         3    7.630       165     540.95  0.7948    0.4984
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Tylko zmienna `Wt`, czyli waga, jest istotna. Jakie wnioski można wyciągnąć? Czy te zwierzęta lepiej wyczuwają wagę niż widzą pozostałe cechy? (tak sugerują autorzy [2018GLM]).

```
> # model z tylko jedną zmienną objaśniającą
> crabs.m2 <- glm(Sat~ log(Wt), family= quasipoisson, data = hcrabs)
> printCoefmat(coef(summary(crabs.m2)), digits = 3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -12.568      2.664   -4.72  4.9e-06 ***
log(Wt)        1.744      0.339    5.15  7.0e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Sprawdźmy jeszcze symulacyjnie jak dobre jest przybliżenie dewiancji rozkładem χ^2. Jeśli model jest dobry, to powinno być dobre. Dobre, oznacza bliskie wartościom teoretycznym: liczba stopni swobody wynosi 171, stąd hipoteza zerowa głosi, że dewiancja ma rozkład $\chi^2(171)$, w którym wartość oczekiwana wynosi 171, a wariancja 2×171 (a więc odchylenie standardowe ≈ 18.5.

```
> # symulacyjne sprawdzenie dopasowania modelu
> # za pomocą dewiancji rezidualnej
> x <- log(hcrabs$Wt)
> dev <- rep(NA, 100)
> n <- length(hcrabs$Sat)
> mu <- fitted(crabs.m2)
> for(i in 1:100){
+   y <- rpois(n, mu)       # Poissonowskie zmienne losowe
+   dev[i] <- glm(y~x, family = quasipoisson)$deviance
+ }
> c( Srednia=mean(dev), odchylenie = sd(dev))
   Srednia   odchylenie
 187.40839   18.74773
```

3.2 MLE dla regresji poissonowskiej

$$P(y;\mu) = e^{-\mu}\,\frac{\mu^y}{y!} = \exp\left\{-\mu + y\log\mu - \log y!\right\}.$$

Zatem

$$l(\underline{\mathbf{y}},\mu) = l(\underline{\beta}) = \sum_{i=1}^{n}[y_i\log\mu_i - \mu_i - \log y_i!] = \sum_{i=1}^{n}\left[y_i\underline{\mathbf{x}}_i^T\underline{\beta} - \exp(\underline{\mathbf{x}}_i^T\underline{\beta}) - \log y_i!\right].$$

Dalej

$$s(\underline{\beta}) = \frac{\partial l}{\partial\underline{\beta}}(\underline{\beta}) = X^T\underline{\mathbf{y}} - X^T\underline{\mu},$$

gdzie $\underline{\mu}^T = \underline{\mu}^T(\underline{\beta}) = [\exp(\underline{\mathbf{x}}_1^T\underline{\beta}), \exp(\underline{\mathbf{x}}_2^T\underline{\beta}), \ldots, \exp(\underline{\mathbf{x}}_n^T\underline{\beta})]$.

Szukamy $\hat{\underline{\beta}}$:

$$s(\hat{\underline{\beta}}) = 0 \qquad \hat{\underline{\beta}} \sim \mathbb{N}(\underline{\beta}, \mathcal{I}^{-1}(\underline{\beta}))$$

przy pewnych założeniach

$$s(\hat{\underline{\beta}}) = 0 \qquad \Leftrightarrow \qquad \sum_{i=1}^{n}\underline{\mathbf{x}}_i y_i = \sum_{i=1}^{n}\underline{\mathbf{x}}_i\exp(\underline{\mathbf{x}}_i^T\underline{\beta}).$$

To jest nieliniowe równanie, więc nie ma rozwiązań analitycznych – konieczność stosowania optymalizacji numerycznej.

Macierz informacji obserwowanej wynosi:

$$\mathcal{I}(\underline{\beta}, \underline{\mathbf{y}}) = \mathbb{E}\left[-\frac{\partial^2 l(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}^T}\right] = \sum_{i=1}^{n} \underline{\mathbf{x}}_i \underline{\mathbf{x}}_i^T \exp(\underline{\mathbf{x}}_i^T \underline{\beta}) = X^T W X,$$

gdzie $W = \operatorname{diag}(\exp(\underline{\mathbf{x}}_i^T \underline{\beta})) = W(\underline{\beta})$. Ponieważ, macierz informacji obserwowanej nie zależy od y, to macierz inf. Fishera $J(\underline{\beta}) = I(\underline{\beta}, \underline{\mathbf{y}})$. Zatem algorytm Fisher Scoring jest tym samym, co algorytm Newtona-Raphsona, i:

$$\underline{\beta}_{k+1} = \underline{\beta}_k - \mathcal{I}^{-1}(\underline{\beta}_k) \cdot s(\underline{\beta}_k)$$

$$\begin{aligned}\underline{\beta}_{k+1} &= \underline{\beta}_k + (\mathbf{X}^T W(\underline{\beta}_k)\mathbf{X})^{-1} \mathbf{X}^T (\underline{\mathbf{y}} - \underline{\hat{\mu}}(\underline{\beta}_k)) \\ &= (\mathbf{X}^T W(\underline{\beta}_k)\mathbf{X})^{-1} \mathbf{X}^T W(\underline{\beta}_k) \left[\mathbf{X}\underline{\beta}_k + W^{-1}(\underline{\beta}_k)(y - \underline{\hat{\mu}}(\underline{\beta}_k))\right].\end{aligned}$$

3.3 Dopasowanie modelu – ocena jakości

Podobnie jak w przypadku regresji logistycznej, do oceny jakości dopasowania modelu, będziemy korzystali z dewiancji oraz rezyduów Pearsona.

3.3.1 Dewiancja w modelu Poissonowskim

$$l(\underline{\beta}) = \sum_i (y_i \log \mu_i - \mu_i - \log(y_i!)).$$

Dla modelu wysyconego (tyle parametrów ile zmiennych)

$$f(x) = y \log x - x$$

$$f'(x) = \frac{y}{x} - 1 = 0 \Leftrightarrow y = x.$$

Stąd maksymalna wartość log-wiarogodności w modelu wysyconym, to

$$l_{\text{wys}}(\underline{\beta}) = \sum_i (y_i \log y_i - y_i - \log(y_i!)).$$

Dla modelu z $p < n$ parametrami:

$$l(\underline{\beta}) = \sum_i (y_i \log \hat{\mu}_i - \hat{\mu}_i - \log(y_i!)) = (y_i \log \hat{y}_i - \hat{y}_i - \log(y_i!)).$$

Ponieważ $\underline{\hat{\mathbf{y}}}_i = \hat{\mu}_i$, bo gdy $Y \sim \text{Pois}(\lambda)$, to $\mathbb{E}Y = \lambda$.

Stąd dewiancja wynosi

$$D = 2\left[\sum y_i \log \frac{y_i}{\hat{y}_i} - \sum (y_i - \hat{y}_i)\right].$$

Uwaga 2. *Tak jak w przypadku regresji logistycznej dewiancja jest asymptotycznie równoważna statystyce Pearsona:*

$$\chi^2 = \sum \frac{(o_i - e_i)^2}{e_i} = \sum \frac{(y_i - \hat{y}_i)^2}{\hat{y}_i}$$

oraz $D, \chi^2 \sim \chi^2(n - (p+1))$ *asymptotycznie.*

3.3.2 Rezydua dla regresji Poissonowskiej

Rezydua Pearsona:

$$r_p = \frac{o_i - e_i}{\sqrt{e_i}} = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i}}$$

Standaryzowane rezydua Pearsona:

$$r_{p,s} = \frac{r_p}{\sqrt{1 - h_{ii}}}$$

gdzie h_{ii} – $i$ty element macierzy daszkowej $\hat{H}$.

Rezydua dewiacyjne

$$r_{D_i} = \text{sign}(o_i - e_i)\sqrt{2(o_i \log \frac{o_i}{e_i} - (o_i - e_i))}$$

Standaryzowane

$$r_{D_i,s} = \frac{r_{D_i}}{\sqrt{1 - h_{ii}}}.$$

3.3.3 Porównywanie modeli zagnieżdżonych

Po pierwsze

$$\frac{D}{\phi} \sim \chi^2_{n-p}.$$

Problemów związanych z nadwyżką rozproszenia $\phi \neq 1$ można uniknąć poprzez zastosowanie testu F zamiast testu ilorazu wiarogodności.

Test F:

Niech $\mathbb{N} \ni q < p \in \mathbb{N}$, Ω – model z p parametrami, ω – model z q parametrami i załóżmy, że $\omega \subset \Omega$. Liczba stopni swobody: $df_\omega = n - q$, $df_\Omega = n - p$.

$$F := \frac{(D_\omega - D_\Omega)/(\phi(df_\omega - df_\Omega))}{D_\Omega/(\phi df_\Omega)} = \frac{(D_\omega - D_\Omega)/(df_\omega - df_\Omega)}{D_\Omega/df_\Omega}.$$

$(D_\omega - D_\Omega)/\phi$ ma rozkład χ^2_{p-q}, a $D_\Omega/\phi \sim \chi^2_{n-p}$. Pamiętamy, że

> Jeśli X i Y są niezależne i $X \sim \chi^2_{d1}$, $Y \sim \chi^2_{d2}$, to
>
> $$\frac{X/d1}{Y/d2} \sim F_{d1,d2}$$
>
> gdzie F jest rozkładem F-Snedecora.

Stąd, przy założeniu, że model ω jest właściwy, to $F \sim F_{[}p-q, n-p]$ asymptotycznie. Hipotezę zerową (H_0: ω jest modelem właściwym) odrzucamy, gdy $F > F^{(\alpha)}_{p-q,n-p}$.

Nadwyżka rozproszenia w modelu dwumianowym

Jeśli $\mathbb{V}\text{ar}(Y) = \phi np(1-p)$, $\frac{\chi^2}{\phi} \sim \chi^2$. Losowość możemy odkryć następująco: $p \sim$ Beta, $Y|p \sim b(n,p)$. Model ten nazywamy *beta-binomial model.*

Pomyśl: Jaki w tej sytuacji będzie rozkład Y?

3.4 Przykładowe zadania

1. Zbiór **discoveries** (opis - patrz: `?discoveries`) zawiera trajektorie szeregu czasowego z liczbą wielkich odkryć od 1860 do 1959 roku. Celem ćwiczenia jest stwierdzenie, czy średnia liczba odkryć w roku jest stała.

   1. Narysować wykres zależności liczby odkryć od czasu (discoveries są obiektem typu time series (ts), dlatego instrukcja plot(discoveries) daje na osi x zmienną o wartościach rzeczywistych).

   2. Zakładając, że liczba odkryć w roku ma rozkład Poissona i postulując model poissonowski: i) przeprowadzić test hipotezy o stałości średniej liczby odkryć postulując najprostszy możliwy model, sprawdzając uprzednio jego dopasowanie ii) metoda alternatywna: podobnie jak w postępowaniu z danymi ze zbioru **kyphosis**, dopasować do liczby odkryć trend kwadratowy względem czasu i stwierdzić, czy współczynniki odpowiadające członowi liniowemu i kwadratowemu są istotne.

2. Zbiór **gala** zawiera informacje o liczbie gatunków żółwi znalezionych na każdej z 30 wysp należących do archipelagu Galapagos oraz o liczbie gatunków stale występujących na danej wyspie (endemicznych). Dodatkowo zbiór zawiera pięć zmiennych geograficznych, które opisują każdą z wysp.

   1. Dopasować model liniowy `species~.` (oprócz zmiennej Endemics). Sporządzić wykres rezyduów (jako funkcji od wartości dopasowanych) i zauważyć wyraźną heteroskedastyczność (niestałość wariancji).

   2. Znaleźć (metodą Boxa–Coxa, funkcja boxcox w bibliotece MASS z opcją plotit=T i wybranym odpowiednio zakresem parametru λ) przekształcenie zmiennej Species poprawiające problem z poprzedniego punktu. Na podstawie analizy Boxa-Coxa wybrać najbardziej naturalną wartość λ. Dopasować nowy model i sporządzić wykres jego rezyduów.

   3. Dopasować model poissonowski. Stwierdzić, czy jest dopasowany.

   4. Obliczyć procent dewiacji objaśnianej przez model poissonowski i porównać go z wartością R^2 w modelu liniowym.

   5. Sprawdzić, czy ewentualne duże wartości odstające są przyczyną problemu ze słabym dopasowaniem modelu poissonowskiego.

   6. Sprawdzić, czy spełnione jest założenie dotyczące rozkładu Poissona o równości średniej i wariancji. W tym celu narysować wykres $(y - \hat{\mu})^2$ jako funkcji od $\hat{\mu}$.

Rozdział 4

Inne ważne zagadnienia

4.1 Ważenie obserwacji

Niekiedy do analizy zebranych danych potrzebne są **wagi** (ang. *weights*). Zdarza się tak, gdy posiadana próba nie reprezentuje (w sposób rzeczywiście losowy) całej populacji. Przykładami takich sytuacji są:

- Analiza danych z badań kompleksowych, np. próbek warstwowych (stratified samples). Wówczas prawodpodobieństwa włączenia do badania mogą być różne dla różnych warstw i stąd powinny mieć różną wagę.
- Brakujące dane (missing at random data) – (propensity score weighting).
- Obserwacje z różną precyzją – Inverse-variance weighting.
- **Zagregowane dane. Wóczas waga koduje liczbę orginalnych obserwacji.**
- Dane z ankiet, w których konkretne pytania mają konkretną wagę (ważność).

Ponadto metoda ważonych najmniejszych kwadratów może być stosowana, gdy nie jest spełnione założenie stałej wariancji błędów (czyli założenie heteroskedastyczności).

4.2 Offset

Po polsku słowo *offset* oznacza m.in. *wyrównanie, balansować, przesunięcie*. W kontekście uogólnionych modeli liniowych offset oznacza zmienną dla której β jest znana. Innymi słowy znane jest nachylenie prostej regresji w kierunku tej zmiennej. Wówczas nie ma sensu estymowanie tej β. Najłatwiej zrozumieć to zjawisko na przykładach.

w R

$\texttt{glm}(\texttt{y} \sim \texttt{offset}(\gamma) + \texttt{x}_1 + \ldots + \texttt{x}_\texttt{p}, \texttt{family} = \texttt{poisson}, \ldots)$

Przykład 1. *Obserwujemy autobusy przyjeżdżające na przystanek i zliczamy je, ale w różnych przedziałach czasowych (różnej długości). Zakładamy, jak często bywa, że przyjeżdżają one zgodnie z rozkładem Poissona o intensywności λ. Przy takich założeniach, liczba autobusów w przedziale Δ ma rozkład* $\mathrm{Pois}(\Delta\lambda)$. *Zatem średnia liczba zdarzeń, to $\mu = \Delta \cdot \lambda$ i składowa systematyczna regresji poissonowskiej wygląda następująco:*

$$\log(\mu_i) = \log(\Delta_i \lambda_i) = \log(\Delta_i) + \beta^T \underline{\mathbf{x}}.$$

Czynnik $\log(\Delta_i)$ *jest w tyn przypadku offsetem.*

Podobnie dzieje się, gdy agregujemy dane w regresji poissonowskiej.

Przykład 2. *Przypuśćmy, że zmienne objaśniające $\underline{\mathbf{x}}$ dzielą całą populację na $k < n$ poziomów. Niech $Y_{i,1}, \ldots, Y_{i,n_i}$ będą odpowiedziami zaobserwowanymi na itym poziomie, gdzie $i \leq n$. Możemy wówczas zagregować dane: $\tilde{y}_i = \sum_{j=1}^{n_i} y_{i,j} \sim \mathrm{Pois}(n_i \mu_i)$. Niech $\lambda_i := \mathbb{E}Y_i = n_i \mu_i$, wtedy*

$$\log(\lambda_i) = \log(n_i) + \beta^T \underline{\mathbf{x}}.$$

Postać ogólna GLM z offsetem:

$$h(\mu_i) = \gamma_i + \beta^T \underline{\mathbf{x}}_i.$$

Macierz informacji Fishera w modelu Poissona z offsetem:

$$F(\beta) = \sum_{i=1}^{n} \underline{\mathbf{x}}_i \underline{\mathbf{x}}_i^T \exp(\gamma_i + \beta^T \underline{\mathbf{x}}_i).$$

Zauważmy, że ponieważ $\mathbb{C}\mathrm{ov}(\hat{\beta}) = F^{-1}(\hat{\beta})$, to wzrost γ_i powoduje spadek błędów standardowych $\hat{\beta}$.

Rozdział 5

Odpowiedzi wielomianowe

W tym rozdziale powiemy sobie o modelu logitowym dla odpowiedzi wielomianowych. Czym jest odpowiedz wielomianowa? Jest to odpowiedz nominalna o skończonej liczbie możliwych wartości. Może być uporządkowana (porządkowa, ordynalna), lecz nie musi. Dla porządku o odpowiedzi wielomianowej mówimy, gdy zm. odpowiedzi przyjmuje więcej niż 2 wartości. Jest to naturalne uogólnienie regresji logistycznej.

Najpierw przywołajmy wiedzę potrzebną do omówienia tego modelu.

Rozkład wielomianowy

Wektor $\mathbf{Y} \in \mathbb{R}^k$ ma rozkład wielomianowy z parametrami $n \in \mathbb{N}$, $p_1 > 0, \ldots, p_k > 0$, $\sum_{i=1}^k p_i = 1$, jeśli

$$\mathbb{P}(\mathbf{Y} = \underline{\mathbf{y}}) = \frac{n!}{y_1! \cdots y_k!} p_1^{y_1} \cdots p_k^{y_k} \, I(y_1 + \ldots + y_k = n).$$

Oznaczamy $\mathbf{Y} \sim \mathrm{mn}(n, p_1, \ldots, p_k)$.

Przykład 3. *Krzyś urządza urodziny w stumilowym lesie. Ma n cukierków i k przyjaciół. Krzyś rzuca cukierki w stronę przyjaciół, a prawdopodobieństwo że osoba j, $j = 1, \ldots, k$, złapie cukierka wynosi p_j, $\sum_{j=1}^k p_j = 1$. Wówczas rozkład liczby cukierków, które złapią przyjaciele jest rozkładem wielomianowym* $\mathrm{mn}(n, p_1, \ldots, p_k)$*, tzn.* $\mathbf{Y} = (Y_1, \ldots, Y_k)$*, Y_i – liczba cukierków itego przyjaciela.*

Następny przykład jest ważny, ze względu na zastosowania.

Związek z rozkładem Poissona

Niech $Y_1, Y_2, \ldots, Y_J$ będą niezależnymi zmiennymi losowymi. $Y_j \sim \text{Pois}(\lambda_j)$, $\lambda_j > 0$. Niech $S = \sum_{j=1}^{J} Y_j$. Znajdźmy rozkład warunkowy wektora $(Y_1, \ldots, Y_J)$ pod warunkiem sumy S.

Dla $n \in N$ i $y_1, y_2, \ldots, y_J \in \mathbb{N}_0$:

$$\mathbb{P}(Y_1 = y_1, Y_2 = y_2, \ldots, Y_J = y_J | S = n) = \frac{P(Y_1=y_1, Y_2=y_2, \ldots, Y_J=y_J, S=n)}{\mathbb{P}(S=n)} \tag{5.1}$$

$$= \begin{cases} 0, & S \neq n \\ \star, & S = n \end{cases} \tag{5.2}$$

Po prostych rachunkach dochodzimy do

$$\star = \frac{n!}{y_1! \cdots y_J!} p_1^{y_1} \cdot \ldots \cdot p_J^{y_J},$$

gdzie $p_i = \frac{\lambda_i}{\sum_{j=1}^{J} \lambda_j}$. Zauważmy, że $\sum_{j=1}^{J} p_i = 1$.

5.1 Model odpowiedzi wielomianowej

Przypuśćmy, że zmienna odpowiedzi przyjmuje C różnych wartości (nominalnych). Mogą to być wartości dowolnego typu: nazwy, litery. Dowolne etykiety. W ogólności nie zakładamy o nich nic. Szczególnym przypadkiem, który omówimy później, są odpowiedzi o wartościach uporządkowanych liniowo. Niech dalej $\pi_{i,j} = \mathbb{P}(Y_i = j | \underline{\mathbf{x}}_i)$ dla $j \in \{1, \ldots, C\}$ będzie prawdopodobieństwem, że $i$ta obserwacja jest w $j$tej kategorii. Oczywiście

$$\sum_{j=1}^{C} \pi_{i,j} = 1.$$

Zauważmy, że wynika stąd, że nieznanych prawdopodobieństw jest tak naprawdę o jeden mniej dla każdego i. Liczbę poziomów dla zmiennej objaśniające oznaczmy przez N. Zatem $i \in \{1, \ldots, N\}$. **Kategoria referencyjna** to jedna wybrana kategoria, która może, ale nie musi, mieć znaczenie kategorii domyślnej. (Jeśli jedna z kategorii jest "domyślna", to warto przyjąć ją za referencyjną. Jeśli takiej nie ma, a wszystkie są dla nas tak samo ważne, to za referencyjną możemy przyjąć dowolną kategorię.) Przyjmijmy, że kategoria referencyjna jest ostatnia kategoria, kategoria C.

Postać modelu:

- Składowa systematyczna:

$$\log\left(\frac{\pi_{i,j}}{\pi_{i,C}}\right) = \underline{\mathbf{x}}_i' \underline{\beta}_j = \sum_{k=0}^{p} x_{i,k}\, \beta_{j,k}, \quad j = 1, \ldots, C-1,\ i = 1, \ldots, N.$$

- Składowa losowa:

$$\mathbb{P}(Y_i = j|\underline{\mathbf{x}}_i) = \pi_{i,j}$$

dla $j \in \{1, \ldots, C-1\}$ i

$$\mathbb{P}(Y_i = C|\underline{\mathbf{x}}_C) = 1 - \sum_{j=1}^{C-1} \pi_{i,j},\ i = 1, \ldots, N.$$

Z postaci modelu wynika, że

$$\pi_{i,C} = \left(1 + \sum_{j=1}^{C-1} \exp(\underline{\mathbf{x}}_i^{'}\underline{\beta}_j)\right)^{-1}$$

oraz

$$\pi_{i,j} = \exp(\underline{\mathbf{x}}_i^{'}\underline{\beta}_j)\left(1 + \sum_{j=1}^{C-1} \exp(\underline{\mathbf{x}}_i^{'}\underline{\beta}_j)\right)^{-1}.$$

Widać podobieństwo do modelu logistycznego, który jest modelem odpowiedzi wielomianowej dla dwóch kategorii.

Uwaga 3. *Kategoria referencyjna pełni ważną rolę. Zapewnia, że prawdopodobieństwa sumują się do* 1. *Ponadto, gdyby jej nie było, to* $\hat{\underline{\beta}}_j$ *nie byłyby jednoznaczne:*

$$\pi_{i,j} = \frac{\exp(\underline{\mathbf{x}}_i^{'}\underline{\beta}_j)}{\sum_{k=1}^{C} \exp(\underline{\mathbf{x}}_i^{'}\underline{\beta}_k)} = \frac{\exp(\underline{\mathbf{x}}_i^{'}(\underline{\beta}_j + c))}{\sum_{k=1}^{C} \exp(\underline{\mathbf{x}}_i^{'}(\underline{\beta}_k + c))}$$

dla dowolnego $c \in \mathbb{R}^{p+1}$.

5.1.1 Interpretacja wyników modelu referencyjnego w przypadku binarnej zmiennej objaśniającej

Rozważmy przypadek **binarnej zmiennej objaśniającej**, $\underline{\mathbf{x}}_i \in \{A, B\}$. Zmienna odpowiedzi Y w dalszym ciągu jest nominalna i przyjmuje C wartości. Dwuwartościowa zmienna objaśniająca obejmuje wiele ważnych przykładów (stary, młody; wykształcony, niewykształcony; duże miasto, małe miasto; kobieta, mężczyzna).

Ustalmy teraz kategorie $j_0 \in \{1, \ldots, C\}$ oraz $\underline{\mathbf{x}}$. Ilorazem szans kategorii j_0 nazywamy iloraz

$$\frac{\pi_{B,j_0}}{\pi_{A,j_0}}.$$

Prosty rachunek pokazuje, że

$$\frac{\pi_{B,j_0}}{\pi_{A,j_0}} = \frac{\pi_{B,C}}{\pi_{A,C}} \exp(\beta_{j_0,B}).$$

Natomiast tzw. **iloraz szans** (*odds ratio*), to

$$\mathrm{OR}_{j_0} = e^{\beta_{j_0,B}}.$$

Rozdział 6

Rozkłady z rodziny Tweedie

Ważnym, acz mniej klasycznym przykładem rozkładów z rodziny wykładniczej są rozkłady zwane *Tweedie*. Jak pokazano w ostatnich latach są one też cenne w pewnych kontekstach modelowania.

The Tweedie EDMs

O rozkładzie z rodziny wykładniczej (EDM) powiemy, że jest Tweedie (albo z rodziny Tweedie), jeśli jego funkcja wariancji jest potęgowa: $V(\mu) = \mu^{\xi}$ dla pewnej wartości rzeczywistej $\xi \notin (0, 1)$, która nazywana jest parametrem indeksującym (*Tweedie index parameter*). Rozkład oznaczamy $\mathrm{Tw}_{\xi}(\mu, \phi)$

Do rodziny Tweedie należą m.in.: rozkład normalny ($\xi = 0$), rozkład Poissona ($\xi = 1$), rozkład gamma ($\xi = 2$), rozkład odwrotny gaussowski ($\xi = 3$). Ponadto, można podzielić rozkłady z tej rodziny, ze względu na ξ:

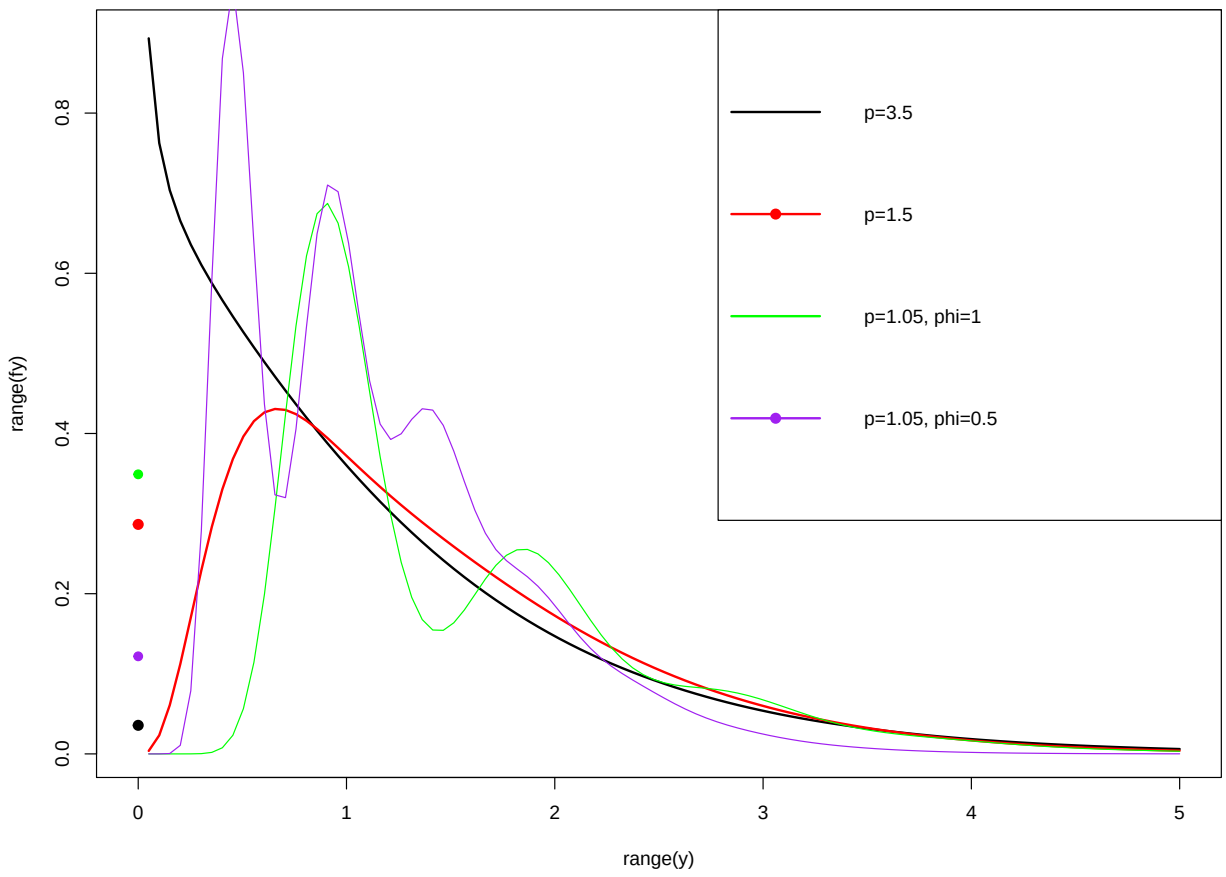
- $\xi \leq 0$ rozkład jest ciągły, a nośnikiem jest cała prosta rzeczywista. Dla $\xi < 0$ wiadomo, ze wartość oczekiwana jest dodatnia. Nie ma (na razie) ważnych zastosowań w tej podrodzinie.
- $\xi = 1$ odpowiada zmiennej dyskretnej o nośniku $\{0, \phi, 2\phi, 3\phi, \ldots\}$.
- $1 < \xi < 2$ –zmienna losowa nieujemna z jednym punktem nieciągłości w 0. Ważne w zastosowaniach, w których występują zera. Rozważmy ten przypadek na przykładzie. Na wykresie gęstości obserwować możemy lokalne maksima w wielokrotnościach ϕ, gdy ξ zbliża się do punktu 1.
- $\xi \geq 2$ – dodatnie, ciągłe zmienne losowe. Rozkład staje się bardziej skośny (w prawo), gdy ξ rośnie.

Podstawowym pakietem jest `tweedie`, który instalujemy w standardowy sposób i wczytujemy

```
> \library(tweedie)
```

Popatrzymy na wykresy gęstości, które można w tym pakiecie generować wyjątkowo wygodnie.

```
> y <- seq(0,5,length=100)
> d0 <-dtweedie( y, power=1.05, mu=1, phi=1)
> # w ten sposób generujemy gęstość
> # ale nie jest to niezbędne dla wykresu

> tweedie.plot( power=1.7, mu=1, phi=1, y=yy, lwd=2)
> tweedie.plot( power=1.2, mu=1, phi=1, y=yy, add=TRUE, lwd=2, col="red")
> tweedie.plot( power=1.05, mu = 1, phi =1 , y=yy, add=TRUE, lwd=1, col="green")
> tweedie.plot( power=1.05, mu = 1, phi =0.5 , y=yy, add=TRUE, lwd=1, col="purple")
> legend("topright",lwd=c(2,2), col=c("black","red"), pch=c(19,19),
       legend=c("p=1.7","p=1.2") )
```



6.1 Struktura Tweedie EDM

Rozkłady Tweedie są zdefiniowane dla pewnego $\xi \in \mathbb{R}$ z funkcją wariancji $V(\mu) = \mu^\xi$. Wiemy, że ta relacja jednoznacznie wyznacza gęstość w klasie rodzin wykładniczych. Ustalając stałe całkowania równe zero, otrzymujemy

$$\theta = \begin{cases} \frac{\mu^{1-\xi}}{1-\xi} \text{ dla } \xi \neq 1 \\ \log \mu \text{ dla } \xi = 1 \end{cases}$$

oraz

$$\kappa(\theta) = \begin{cases} \frac{\mu^{2-\xi}}{2-\xi} \text{ dla } \xi \neq 2 \\ \log \mu \text{ dla } \xi = 2 \end{cases}.$$

Przypomnijmy, że gęstość ma postać

$$P(y; \theta, \phi) = a(y, \phi) \exp\left\{\frac{y\theta - \kappa(\theta)}{\phi}\right\}$$

więc dla $\xi \notin \{1, 2\}$:

$$P(y; \xi, \phi) = a(y, \phi) \exp\left\{\frac{\mu^{1-\xi}}{\phi}\left(\frac{y}{1-\xi} - \frac{\mu}{2-\xi}\right)\right\}.$$

Zrób to sam!

Pokazać, że dla danego ξ wzory na θ i $\kappa(\theta)$ wyglądają tak jak powyżej.

Znaleźć wartość wartości oczekiwanej i wariancji dla rozkładu Tweedie z parametrami $\phi = 2$ i $\xi = 3/2$ i $\theta = -3/2$.

Zadanie

Pokazać, że dla $\xi \notin \{1, 2\}$ dewiancja jednostkowa

$$d(y, \mu) = 2\left[\frac{\max(y, 0)^{2-\xi}}{(1-\xi)(2-\xi)} - \frac{y\mu^{1-\xi}}{1-\xi} + \frac{\mu^{2-\xi}}{2-\xi}\right].$$

Aproksymacja rozkładem χ-kwadrat zachodzi, gdy $\phi \leq \min(y)^{2-\xi}/3$ dla $\xi \geq 1$. Zauważmy, że apkrosymacja może być słaba gdy 0 występuje w danych.

Parametr ξ też musimy wyestymować. Ponieważ logarytm wariancji zależy liniowo od logarytmu średniej (zobacz sam), to można znaleźć stałą tej zależności.

Przykład - ciąg dalszy

```
> #Group by SOI phase
> mn <- with(quilpie, tapply(Rain, Phase, "mean"))
> vr <- with(quilpie, tapply(Rain, Phase, "var"))
> coef(lm(log(vr)~log(mn)))
> # Więc wyestymowane $\hat\xi = 1.553380$.
```

Lub inaczej

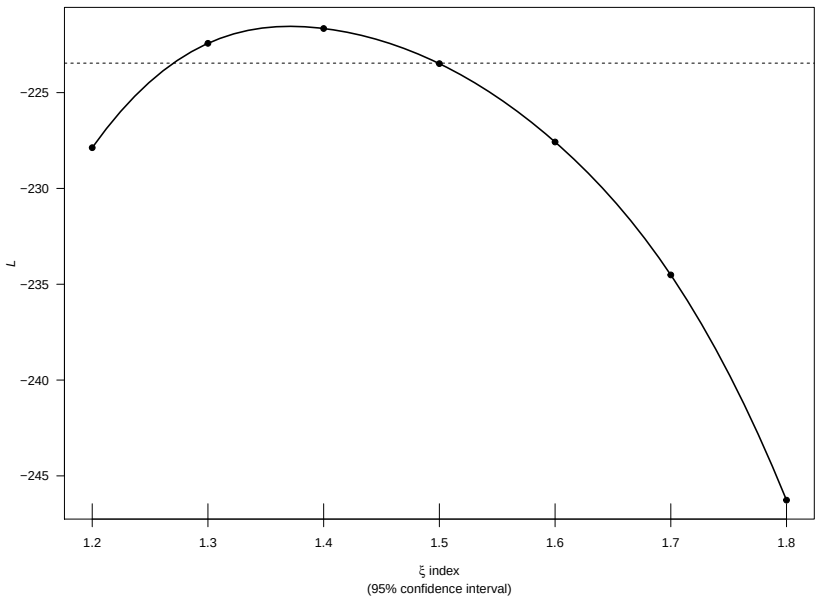
```
> # Group by Decade
> Decade <- cut(quilpie$Year, breaks=seq(1920, 1990, by=10))
> mn <- tapply(quilpie$Rain, Decade, "mean")
> vr <- tapply(quilpie$Rain, Decade, "var")
> coef(lm(log(vr)~log(mn)))
> # Tutaj wyestymowane $\hat\xi = 1.9459524$.
```

Jak widać wyniki są różne! Na szczęście obydwa mieszczą się w przedziale $(1, 2)$.

Ażeby uniknąć zależności od sposobu dzielenia na podgrupy należy przedsięwziąć działania. Jedną z możliwości jest znalezienie MLE dla ξ. Służy do tego funkcja `tweedie.profile` z pakietu `tweedie`.

```
> quilpie$Phase <- factor(quilpie$Phase)
> plot(Rain~Phase, data=quilpie, ylab = "Total July rainfall", ylim=c(0,100), las=1)
> out <- tweedie.profile(Rain~Phase, do.plot= TRUE, > data=quilpie)
> names(out)
> xi.est <- out$xi.max
> x.est <- round(xi.est, 2); xi.est
[1] 1.37
```

Funkcja `tweedie.profile`, między innymi cennymi informacjami, patrz `names(out)`, zwraca wykres.



6.2 Tweedie GLM dla dodatnich ciągłych danych z zerami

Dla $1 < \xi < 2$ Tweedie GLMy są używane do modelowania danych ciągłych na $(0, \infty)$ ze skokiem w 0. Motywacją mogą być dane związane z ubezpieczeniami. Przypuśćmy, że N jest liczbą szkód u pewnego klienta w pewnym, ustalonym, okresie czasu. Niech $N \sim \text{Pois}(\lambda^\star)$. Oczywiście N może być równe 0 i będzie to odpowiadało brakowi szkód. Jeśli natomiast $N > 0$, tj. zaszła co najmniej jedna szkoda, to możemy patrzeć na łączny koszt wszystkich wypłaconych odszkodowań. Niech więc Z_i będzie wielkością $i$tego odszkodowania dla $i = 1, \ldots, N$ i zakładamy, że $Z_i \sim \mathcal{G}(\mu^\star, \phi^\star)$. Całkowita wypłacona kwota to:

$$Y = \sum_{i=1}^{N} Z_i.$$

(Suma równa jest zero, gdy $N = 0$).

Zadanie

Pokazać, że Y ma rozkład Tweedie z parametrem $\xi \in (1, 2)$.

Rozkład ten nazywany jest też czasem rozkładem gamma–Poissona.

Rozwiązanie

Jeśli Y jest zmienną o złożonym rozkładzie Poissona z $\text{Pois}(\lambda)$ i $\mathcal{G}(\alpha, \beta)$, to Y ma wartość oczekiwaną $\lambda\frac{\alpha}{\beta}$ i $\mathbb{V}\text{ar}(Y) = \lambda\alpha(\alpha+1)/\beta^2$. Z drugiej strony $\mathbb{E}Y = \mu$ i $\mathbb{V}\text{ar}Y = \psi\mu^p$. Wtedy jeśli $\lambda = \frac{\mu^{2-p}}{\psi(1-p)}$, $\alpha = \frac{2-p}{p-1}$, $1/\beta = \psi(p-1)\mu^{p-1}$. Oraz liczymy rozkład Y. Oczywiście $\mathbb{P}(Y = 0) = e^{-\lambda}$ oraz

$$f_Y(y) = e^{-\beta y}e^{-\lambda}\sum_{n=1}^{\infty}\frac{\beta^{n\alpha}}{\Gamma(n\alpha)}y^{n\alpha-1}\frac{\lambda^n}{n!}, \quad y > 0.$$

Ponieważ $\lambda\beta^\alpha$ nie zależy od μ, to cała suma będzie funkcją $a(y;\psi)$ dla $y > 0$. Wtedy $\theta = -\beta\psi = -\frac{1}{(p-1)\mu^{p-1}}$, $\mu(\theta) = (-\theta(p-1))^{1/(p-1)}$ i $b(\theta) = \lambda\psi = \frac{\mu^{2-p}}{2-p}$. Granicznie otrzymujemy też przypadki $p = 1$ (Poisson) i $p = 2$ (gamma).

6.2.1 IBNR

Ciekawy przykład zastosowania do estymacji IBNR w praktyce aktuarialnej przedstawia Rob Kaas w *COMPOUND POISSON DISTRIBUTIONS AND GLM'S — TWEEDIE'S DISTRIBUTION.*

6.2.2 Studium przypadku

Teraz możemy dopasować model za pomocą funkcji `glm()`. Żeby w parametrze `family` móc wpisać `tweedie` konieczne jest załadowanie pakietu `statmod`.

```
> library(statmod)
> # \mu^(link.power) (if link.power =0,then \log(\mu))
> m.quilpie <- glm(Rain~Phase, data = quilpie,
        family=tweedie(var.power=xi.est, link.power=0))
> printCoefmat(coef(summary(m.quilpie)))
```

Jeśli nie wpiszemy parametru `link.power`, to domyślnie funkcja łącząca będzie kanoniczna.

Spróbuj sam!

Uświadom sobie, że kanoniczna funkcja łączącą dla rozkładu $\mathrm{Tw}_\xi(\mu, \phi)$, $\xi \in (1, 2)$ ma postać:

$$\frac{\mu^{1-\xi}}{1-\xi}.$$

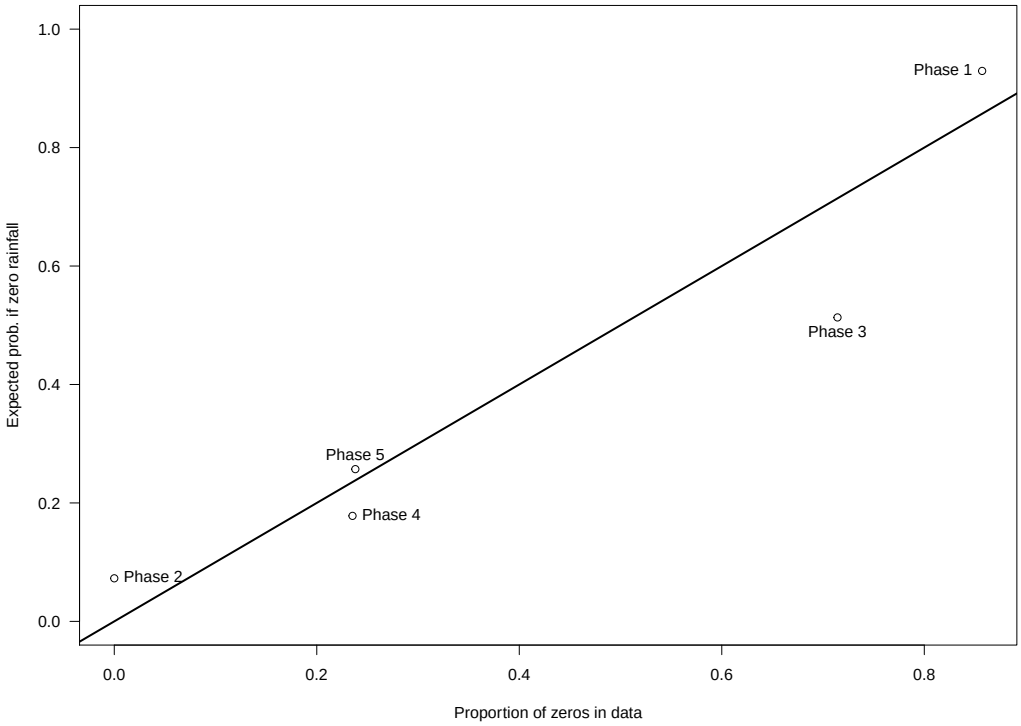
Zgodność dopasowania sprawdzimy wykresami kwantylowymi.

```
> #Residua Pearsona, diewiancyjne i kwantylowe
> dres <- resid(m.quilpie)
> pres <- resid(m.quilpie, type = "pearson")
> qres1 <- qresid(m.quilpie)
> qres2 <- qresid(m.quilpie)
> qqnorm(dres, main = "Deviance residuals", las=1); > qqline(dres)
> qqnorm(pres, main="Pearson residuals", las=1); qqline(pres)
> qqnorm(qres1, main = "QUantile residuals (set1)", las =1); qqline(qres1)
> qqnorm(qres2, main = "QUantile residuals (set2)", las =1); qqline(qres2)
```

Porównamjmy teraz przewidywaną liczbę miesiecy bez opadów z danymi.

```
> #Modelowane prawdopodobieństwo P(Y=0)
> new.phase <- factor(c(1,2,3,4,5))
> mu.phase <- predict(m.quilpie, newdata = data.frame(Phase = new.phase), type="response")
names(mu.phase) <- paste("Phase", 1:5)
> mu.phase
> phi.mle <- out$phi.max
> pi0 <- exp(-mu.phase^(2-xi.est)/(phi.mle*(2-xi.est)))
```

```
> # Obserwowane prawdopodobieństwo P(Y=0)
> prop0 <- tapply(quilpie$Rain, quilpie$Phase, function(x){sum(x==0)/length(x)})
> prop0

> #Plot
> plot(pi0~prop0, xlab="Proportion of zeros in data", ylim=c(0,1),
+                     ylab = "Expected prob. if zero rainfall", las=1)
> abline(0, 1, lwd=2) #linia równości
> text(prop0, pi0, #add labels to the points
+     labels = paste("Phase", levels(quilpie$Phase)),
+     pos=c(2,4,1,4,3) #positions of the lables
     )
```



Widzimy, że nie jest źle.

Pakiet `tweedie` oferuje też możliwość *rozbicia* rozkładu typu Tweedie na miksturę rozkładów Poissona i gamma, zgodnie z interpretacją na początku rozdziału. Służy do tego funkcja `tweedie.convert`.

```
> # Interpretacja jako model Poisson-Gamma
> out <- tweedie.convert(xi=xi.est, mu=mu.phase, phi=phi.mle)
> downscale<-rbind("Poisson mean"     = out$poisson.lambda,
+                  "Gamma mean"       = out$gamma.mean,
+                  "Gamma dispersion" = out$gamma.phi)
> colnames(downscale) <- paste("Phase", 1:5)
> downscale

[1]                  Phase 1   Phase 2    Phase 3   Phase 4    Phase 5
Poisson mean     0.07320691  2.620963  0.6670653  1.725099  1.3584917
Gamma mean       0.16595357  1.374692  0.6123964  1.073665  0.9323044
Gamma dispersion 1.44012592 98.818480 19.6106718 60.278817 45.4509207
```

Dzięki takiemu rozbiciu możemy szybko zobaczyć jaka jest estymowana średnia liczba opadów i jaki jest średnia ilość pojedyńczego opadu.

```
> #Mean rainfall from data
> tapply(quilpie$Rain, quilpie$Phase, "mean")
         1          2          3          4          5
 0.1142857 33.8937500  3.8428571 17.4235294 11.9142857
> #Mean rainfall from model
> mu.phase
   Phase 1    Phase 2    Phase 3    Phase 4    Phase 5
 0.1142857 33.8937500  3.8428573 17.4235294 11.9142857

> tapply(quilpie$Rain, quilpie$Phase, min)
  1   2   3   4   5
0.0 3.6 0.0 0.0 0.0
> round(out$p0, 2)
[1] 0.93 0.07 0.51 0.18 0.26
```

Rozdział 7

GAMLSS

Skrót GAMLSS pochodzi od nazwy: *the generalized additive models for location, scale and shape* (uogólnione modele addytywne dla lokacji, skali i kształtu) – Statinopoulus i Rigby, 2007.

Niezależne obserwacje: y_i, $i = 1, \ldots, n$. Funkcja prawdopodobieństwa (gęstość) $f(y_i|\theta^i)$, gdzie $\theta^i = (\mu_i, \sigma_i, \nu_i, \tau_i)$. Każdy z czterech parametrów może być zależny od zmiennych objaśniających. Te cztery parametry nazywamy *parametrami rozkładu.* Można model stosować dla jeszcze większej liczby parametrów.

g_k – funkcja łącząca, o której zakładamy, że jest monotoniczna.

$$g_k(\theta_k) = \eta_k = X_k\beta_k + \sum_{j=1}^{J_k} Z_{jk}\gamma_{jk}$$

dla $k = 1, 2, 3, 4$.

Oznaczenia:

- $\beta_k^T = (\beta_{1k}, \beta_{2k}, \ldots, \beta_{J_k^{'}k})$,
- X_k – macierz eksperymentu (ustalona i znana), wymiaru $n \times J_k^{'}$,
- Z_{jk} – ustalona, znana design matrix wymiaru $n \times q_{jk}$,
- $\gamma_{jk} \sim N_{q_{jk}}(0, G_{jk}^{-1})$

Estymacja wektorów β_k i parametry 'efektu losowego' γ_{jk} dla $j = 1, 2, \ldots, J_k$ i $k = 1, 2, 3, 4$ polega na maksymalizacji funkcji logwiarogodności z karą:

$$l_p = l - \frac{1}{2}\sum_{k=1}^{p}\sum_{j=1}^{J_k}\lambda_{jk}\gamma_{jk}^T G_{jk}\gamma_{jk},$$

gdzie $l = \sum_{i=1}^{n} \log f(y_i|\theta^i)$.

Biblioteką w `R` jest `gamlss`, którą instalujemy i wczytujemy w standardowy sposób. Przykłady użycia są dostępne w pliku `gamlssintro.R`.

Rozdział 8

Modele graficzne

8.1 Wprowadzenie do pakietów gRbase, gRain i igraph

```
# Graphical models - wykład R
# installing/loading the package:
 if(!require(installr)) { install.packages("installr"); require(installr)} #load / install+l

# updateR(F, T, T, F, T, F, T)
setRepositories()
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(c("graph","Rgraphviz", "RBGL"))
install.packages("gRbase", dependencies=TRUE);
install.packages("gRain", dependencies=TRUE);
install.packages("gRim", dependencies=TRUE)
library(gRbase)

# Trzy równoważne sposoby tworzenia grafu (za kazdym razem wskazujemy kliki) nieskierowanego
# ug - undirected graph
ug0 <- ug(~a:b, ~b:c:d, ~e)              # po przecinku wskazujemy kolejne kliki
ug0
plot(ug0)
ug0 <- ug(~a:b + b:c:d + e)              # po znaku '+' wskazujemy kolejne kliki
ug0 <- ug(c("a","b"),c("b","c","d"),"e") # ...
ug0                                      # jedynie słowne opis
```

```
library(Rgraphviz)
plot(ug0)                       # teraz obrazek

ug01 <- ug(~a:b, ~b:c:d, ~e, result = "matrix") # ten sam graf, inna reprezentacja

ug01                            # graf w postaci macierzy sąsiedzctwa
plot(ug01)                      # niekoniecznie

nodes(ug0)                      # wierzchołki
edges(ug0)                      # krawędzie
edges(ug0)$a

nodes(ug01)                     # nie ma pakietów doskonałych! (error)

ug0i <- ug(~a:b, ~b:c:d, ~e, result = "igraph")  # ten sam graf, inna reprezentacja

ug0i
plot(ug0i)

library(igraph)
V(ug0i)                         # wierzchołki vertex
E(ug0i)                         # krawędzie

E(ug)                           # no cóż... błąd!
nodes(ug0i)                     # ...

# Teraz mała modyfikacja...
V(ug0i)$size <- c(10,20,30,40,50) # rozmiar wierzchołków
V(ug0i)$label.cex <- 5            # rozmiar czcionki etykiet
V(ug0i)                           # tutaj nie widać zmian...
plot(ug0i)

# Można też utworzyć igraph bezpośrednio:
g1 <- graph( edges=c(1,2, 3,4), n=6, directed=F )
plot(g1)
```

```
class(g1)
g1
# Now with 10 vertices, and directed by default:
g2 <- graph( edges=c(1,2, 2,3, 3, 1, 1, 3), n=10 )
plot(g2)

g3 <- graph( c("John", "Jim", "Jim", "Jill", "Jill", "John"))
# Jeśli wierzchołki są nazwane, to nie trzeba podawać ich liczby.
plot(g3)

g4 <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"),
             isolates=c("Jesse", "Janis", "Jennifer", "Justin") )
g5 <- graph( c("John", "Jim", "Jim", "Jack", "Jim", "Jack", "John", "John"), n=10 ) #błąd!

plot(g4, edge.arrow.size=.5, vertex.color="green", vertex.size=15,
     vertex.frame.color="gray", vertex.label.color="black",
     vertex.label.cex=0.8, vertex.label.dist=2, edge.curved=c(-1, -5, 5, 1))

######
g <- graph( c(0,1,1,0,1,2,1,3,1,3,1,3,
              2,3,2,3,2,3,2,3,0,1)+1 )

curve_multiple(g)

plot(graph_from_literal(a---b, b---c))
# Domyślnie graf jest uproszczony: pozbawiony pętli i wielokrotnych krawędzi
# parametr simplify = TRUE lub FALSE
plot(graph_from_literal(a:b:c---c:d:e))
plot(graph_from_literal(a:b:c---c:d:e, c--c), simplify=FALSE)
gl <- graph_from_literal(a-b-c-d-e-f, a-g-h-b, h-e:f:i, j)
plot(gl)
plot(graph_from_literal(a+-b, b+-+c))
```

```
#macierz grafu
g4[]                          # znaczenie tych wartości?
g4[1,]                        # pierwszy wiersz
g4[,1]                        # pierWSZa kolumna
# Do sieci są przypisane atrybuty
V(g4)$name
# I można też dodawać nowe
V(g4)$plec <- c("m","k","m", "m","k","k","m")
E(g4)$type <- "email" # Atrybut krawędzi: przypisz "email" wszystkim krawędziom
E(g4)$weight <- 10    # Waga krawędzi: przypisz wszystkim wagę 10
edge_attr(g4)

vertex_attr(g4)

g4 <- set_graph_attr(g4, "name", "Email Network")
g4 <- set_graph_attr(g4, "something", "A thing")
graph_attr_names(g4)
graph_attr(g4)
g4 <- delete_graph_attr(g4, "something")
graph_attr(g4)

# Ładny przykład wykorzystania atrybutów
plot(g4, edge.arrow.size=.5, vertex.label.color="black", vertex.label.dist=1.5,
     vertex.color=c( "pink", "skyblue")[1+(V(g4)$plec=="k")] )

g4s <- simplify( g4, remove.multiple = T, remove.loops = F,
                 edge.attr.comb=c(weight="sum", type="ignore") )
plot(g4s, vertex.label.dist=1.5)
edge_attr(g4s)
g4s

# Opis obiektu typu igraph rozpoczyna się sekencją czterech lub mniej liter:
#1. D - skierowany lub U - nieskierowany,
#2. N - jeśli wierzchołki są nazwane,
```

```
#3. W - jeśli to krawędzi są przypisane wagi,
#4. B - jeśli wierzchołki mają przypisane typ.

#make_empty_graph(n), make_full_graph(n), make_star(n)
tr <- make_tree(41, children = 3, mode = "undirected")
plot(tr, vertex.size=10, vertex.label=NA)

rn <- make_ring(40)
plot(rn, vertex.size=10, vertex.label=NA)
er <- sample_gnm(n=100, m=40)
plot(er, vertex.size=6, vertex.label=NA)

plot(er %du% tr %du% rn, vertex.size=10, vertex.label = NA)
     # Kliki (maksymalne podgrafy pełne)
library(RBGL)
is.complete(ug0, c("b","c","d")) # sprawdzamy, czy jest podgrafem pełnym
is.complete(ug0, c("b","c"))   # ale nie, czy jest kliką maksymalną
maxClique(ug0)                 # a to wypisuje wszystkie klika maksymalne #RBGL
plot(ug0)
# Ścieżki i separatory
separates(a = "a", b = "d", S1 = c("b", "e"), ug0)  # czy a i b są separowane przez S1 w gra
separates(a = "a", b = "d", S1 = c("b", "e"), ug01) # błąd!
separates(a = "a", b = "d", S1 = c("b", "e"), ug0i) # błąd!

# podgrafy
ug1 <- subGraph(c("b","c","d","e"), ug0) # podgraf indukowany przez zadane wierzchołki
plot(ug1)
adj(object = ug0, "c")         # wszyscy sąsiedzi nb(c) bd(c)
closure("c", ug0)              # sąsiedzi razem z wierzchołkiem cl(c)

# Wierzchołek jest *simplicjalny* (prostolinijny!), jeśli jego sąsiedztwo tworzy graf pełny
# Można sprawdzić, czy wierzchołek jest simplicjalny tak:
is.complete(ug0, adj(object = ug0, "c")$c)
# albo tak
is.simplicial(set = "c", object = ug0)
```

```
simplicialNodes(object = ug0)  # wszystkie wierzchołki simplicjalne

# A tu składowe spójności
connectedComp(g = ug0)

# Graf *chordalny*, to taki w którym każdy cykl długości >= 4, ma przechątną (chord)
plot(ug0)
is.triangulated(ug0)           # obiekt typu graph
is.chordal(ug0i)               # obiekt typu igraph

# Niech (A; B; S) będzie trójką podzbiorów V.
# (A; B; S) jest dekompozycją G=(V,E), jeśli
#     i. (A; B; S) są rozłączne oraz  V = A \cup B \cup S
#     ii. S jest podgrafem pełnym
#    iii. S separuje A i B w G

is.decomposition(set = "a", set2 = "d", set3 = c("b","c"), ug0)
# bo nie jest spełniony warunek i.

ug1 <- subGraph(c("b","c","a","d"), ug0)
plot(ug1)
is.decomposition(set = "a", set2 = "d", set3 = c("b","c"), ug1)    # a teraz jest
is.decomposition(set = "a", set2 = c("d","b"), set3 = "c", ug1)
```

8.2 Sieć bayesowska

Omawiamy przykład tworzący sieć bayesowską na podstawie zadanych związków między zmiennymi.

```
>  install.packages("gRain")
>  library(gRain)
>  g<-list(~asia, ~tub | asia, ~smoke, ~lung | smoke, ~bronc | smoke,\\
> +   ~either | lung :   tub, ~xray | either, ~dysp | bronc : either)
>  chestdag<-dagList(g)
>  yn <- c("yes","no")
> a    <- cptable(~asia, values=c(1,99),levels=yn)
> t.a  <- cptable(~tub|asia, values=c(5,95,1,99),levels=yn)
> s    <- cptable(~smoke, values=c(5,5), levels=yn)
> l.s  <- cptable(~lung|smoke, values=c(1,9,1,99), levels=yn)
> b.s  <- cptable(~bronc|smoke, values=c(6,4,3,7), levels=yn)
> e.lt <- cptable(~either|lung:tub,values=c(1,0,1,0,1,0,0,1),levels=yn)
> x.e  <- cptable(~xray|either, values=c(98,2,5,95), levels=yn)
> d.be <- cptable(~dysp|bronc:either, values=c(9,1,7,3,8,2,1,9), levels=yn)
> plist <- compileCPT(list(a, t.a, s, l.s, b.s, e.lt, x.e, d.be))
> net1 <- grain(plist)
```

Żeby narysować graf potrzebujemy pakietu `Rgraphviz`.

```
> require(Rgraphviz)
> plot(net1)
> options("prompt"="> ","width"=85)
```

Tworzymy sieć bayesowską.

```
> plist <- compileCPT(list(a, t.a, s, l.s, b.s, e.lt, x.e, d.be))
> plist$tub
> plist$either ## Notice: a logical node
> net1 <- grain(plist)
> net1
# Zapytania
> querygrain(net1, nodes=c("lung","bronc"), type="marginal")
> querygrain(net1,nodes=c("lung","bronc"), type="joint")
```

Równoważnie

```
> et12  <- setEvidence(net1, evidence=list(asia="yes", dysp="yes"))
> net12  <- setEvidence(net1,
> +                     nodes=c("asia", "dysp"), states=c("yes", "yes"))
> pEvidence( net12 ) # prawdopodobieństwo warunku, który narzucamy
> querygrain( net12, nodes=c("lung","bronc") )
> querygrain( net12, nodes=c("lung","bronc"), type="joint" )
#warunek o prawdopodobieństwie 0
> net13 <- setEvidence(net1,nodes=c("either", "tub"),
> +                   states=c("no","yes"))
> pEvidence( net13 )
> querygrain( net13, nodes=c("lung","bronc"), type="joint" )
> tt <- querygrain( net1, type="joint")  #wszystkie scenariusze!
> sum(tt==0)/length(tt)
> sum(tableSlice(tt, c("either","tub"), c("no","yes")))
# widać, że prawdopodobieństwo żadanego warunku wynosi 0
```