M7222 – 2. cvičení : **GLM02a**

(*Zotavení v závislosti na závažnosti nemoci a návštěvě nemocnice*)

Nejprve načteme vstupní data pomocí příkazu `read.csv2()` a podíváme se na jejich strukturu pomocí příkazu `str()`.

```
> fileDat <- paste(data.library, "InfectionSeverity.csv", sep = "")
> data <- read.csv2(fileDat, header = TRUE, sep = ";", dec = ".")
> str(data)
```

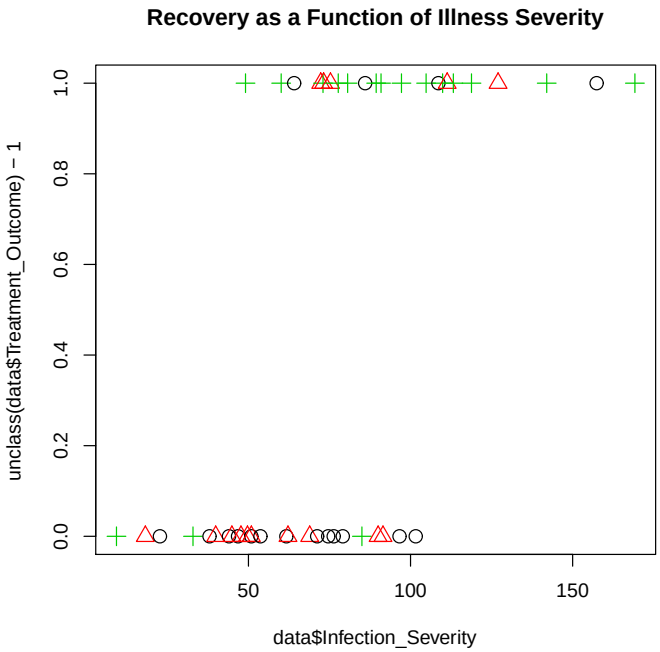
```
,data.frame,:   49 obs. of  3 variables:
 $ Infection_Severity: num  9.3 18.2 22.7 32.9 38 39.9 44 44.9 46.8 47.7 ...
 $ Treatment_Outcome : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Hospital          : int  3 2 1 3 1 2 1 2 1 2 ...
```

Z proměnných, které jsou kategoriální, utvoříme pomocí příkazu `factor()` proměnné typu faktor.

```
> data$Treatment_Outcome <- factor(data$Treatment_Outcome, labels = c("survived",
    "died"))
> data$Hospital <- factor(data$Hospital, labels = c("A", "B", "C"))
```

Data vykreslíme

```
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) - 1, type = "p",
    pch = unclass(data$Hospital), col = unclass(data$Hospital), cex = 1.5,
    main = "Recovery as a Function of Illness Severity")
```



Obrázek 1: Vykreslení dat pomocí příkazu `plot`.

Protože tento graf je málo srozumitelný, provedeme nejprve kategorizaci proměnné `Infection_Severity` do 10 subintervalů. Pak zjistíme počet osob, které přežily, popř. zemřely v jednotlivých subintervalech, na základě toho odpovídající relativní četnosti zemřelých.

```
> breaks_f_sev <- seq(0, 170, length.out = 11)
> f_sev <- cut(data$Infection_Severity, breaks = breaks_f_sev)
> (TabSurvDied <- table(f_sev, data$Treatment_Outcome))
```

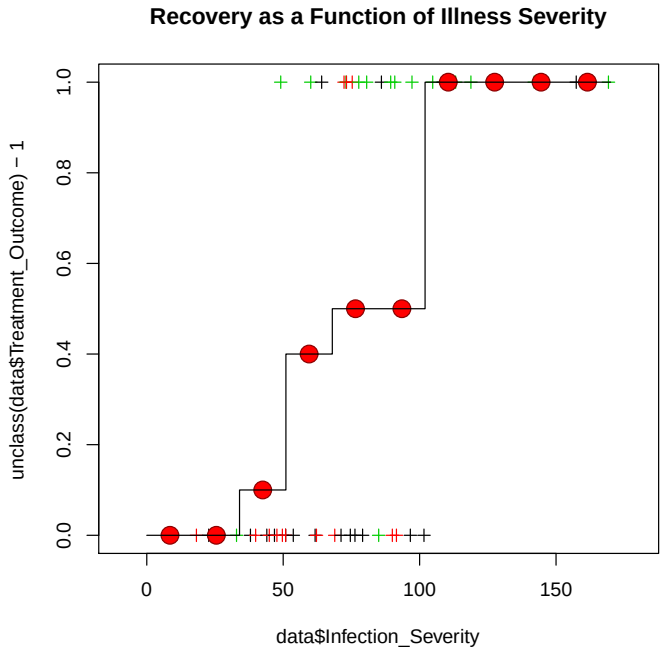
```
f_sev       survived died
  (0,17]           1    0
  (17,34]          3    0
  (34,51]          9    1
  (51,68]          3    2
  (68,85]          6    6
  (85,102]         4    4
  (102,119]        0    6
  (119,136]        0    1
  (136,153]        0    1
  (153,170]        0    2
```

```
> RelativDied <- TabSurvDied[, 2]/rowSums(TabSurvDied)
> (tab2 <- cbind(TabSurvDied, RelativDied))
```

```
          survived died RelativDied
(0,17]           1    0         0.0
(17,34]          3    0         0.0
(34,51]          9    1         0.1
(51,68]          3    2         0.4
(68,85]          6    6         0.5
(85,102]         4    4         0.5
(102,119]        0    6         1.0
(119,136]        0    1         1.0
(136,153]        0    1         1.0
(153,170]        0    2         1.0
```

Předchozí graf nyní budeme modifikovat tak, abychom doplnili relativní četnosti zemřelých v jednotlivých kategoriích

```
> delta2 <- 0.5 * diff(breaks_f_sev)[1]
> N <- length(breaks_f_sev)
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
      1, type = "p", pch = 3, xlim = c(-10, 180), col = unclass(data$Hospital),
      main = "Recovery as a Function of Illness Severity")
> points(breaks_f_sev[1:(N - 1)] + delta2, tab2[, 3],
      pch = 21, col = "darkred", cex = 2, bg = "red")
> lines(breaks_f_sev, c(0, tab2[, 3]), type = "S")
```



Obrázek 2: Vykreslení relativních četností zemřelých.

Model 1 - binární regresní model s jedinou spojitou kovariátou `Infection_Severity`

Za g volíme některou z linkovacích funkcí, takže dostáváme

$$\eta(\mathbf{x}) = g_1(\pi(\mathbf{x})) = \Phi^{-1}(\pi(\mathbf{x})) \qquad \text{probitový model} \qquad (1)$$

$$\eta(\mathbf{x}) = g_2(\pi(\mathbf{x})) = \log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) \qquad \text{logistický model} \qquad (2)$$

$$\eta(\mathbf{x}) = g_3(\pi(\mathbf{x})) = \log[-\log(1-\pi(\mathbf{x}))] \qquad \text{komplementární log-log model} \qquad (3)$$

Nejprve zvolíme kanonickou linkovací funkci, takže dostaneme logistický regresní model:

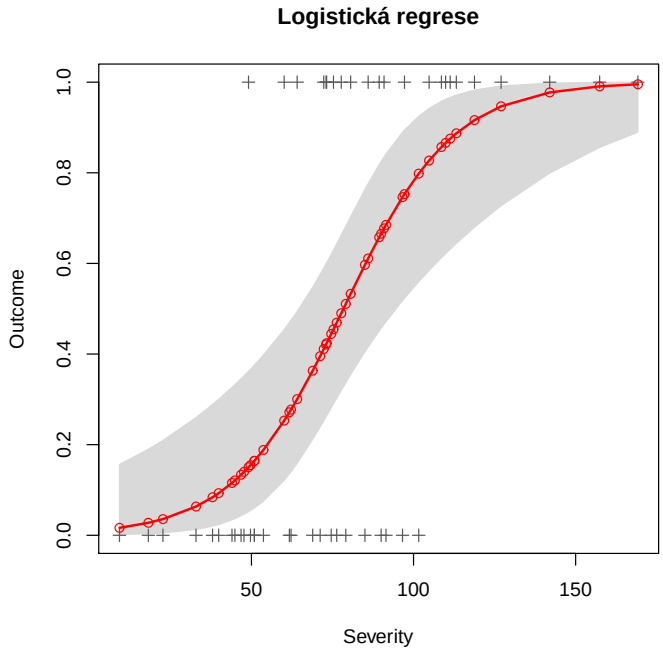
```
> m1.logit <- glm(Treatment_Outcome ~ Infection_Severity,
     family = binomial(logit), data = data)
> summary(m1.logit)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity, family = binomial(logit),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7891  -0.6459  -0.2365   0.7533   1.9474
```

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.64050    1.38335  -3.355 0.000795 ***
Infection_Severity  0.05921    0.01758   3.368 0.000756 ***
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 45.994  on 47  degrees of freedom
AIC: 49.994

Number of Fisher Scoring iterations: 5
```

Vidíme, že kovariáta `Infection_Severity` je v tomto modelu statisticky významná.

Provedeme vykreslení výsledné logistické křivky spolu s asymptotickými intervaly spolehlivosti:

```
> predicted.logit <- predict(m1.logit, type = "link",
     newdata = data, se = T)
> data$CI.lower.logit <- plogis(predicted.logit$fit -
     1.96 * predicted.logit$se.fit)
> data$fitted.logit <- plogis(predicted.logit$fit)
> data$CI.higher.logit <- plogis(predicted.logit$fit +
     1.96 * predicted.logit$se.fit)
> x <- c(data$Infection_Severity, rev(data$Infection_Severity))
> y <- c(data$CI.lower.logit, rev(data$CI.higher.logit))
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
     1, type = "n", pch = 3, ylab = "Outcome", xlab = "Severity",
     main = "Logistická regrese")
> polygon(x, y, col = "gray85", border = "gray85")
> points(data$Infection_Severity, unclass(data$Treatment_Outcome) -
     1, pch = 3, ylab = "Outcome", xlab = "Severity",
     col = "gray35")
> lines(data$Infection_Severity, data$fitted.logit, col = "red",
     lwd = 2)
> points(data$Infection_Severity, data$fitted.logit, col = "red")
```



Obrázek 3: Logistická regrese s intervaly spolehlivosti.

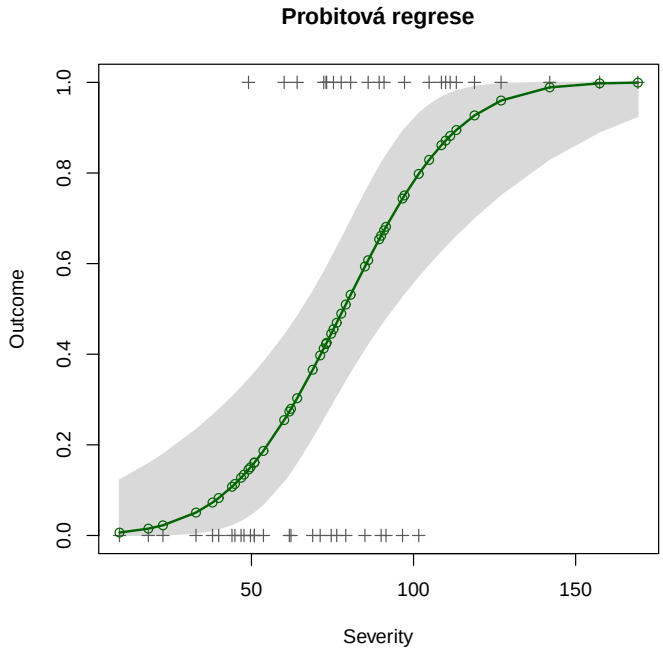
Podívejme se, jak dopadne **probitový regresní model** s jedinou kovariátou `Infection_Severity`, to znamená musíme zvolit místo kanonické linkovací funkce jinou, a to probitovou linkovací funkci.

```
> m1.probit <- glm(Treatment_Outcome ~ Infection_Severity,
     family = binomial(probit), data = data)
> summary(m1.probit)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity, family = binomial(probit),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7885  -0.6427  -0.1742   0.7577   1.9636

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.824141   0.763012  -3.701 0.000214 ***
Infection_Severity  0.036010   0.009699   3.713 0.000205 ***
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 45.597  on 47  degrees of freedom
AIC: 49.597

Number of Fisher Scoring iterations: 6
```

I v tomto modelu je kovariáta `Infection_Severity` statisticky významná.

Opět vykreslíme výslednou probitovou křivku spolu s asymptotickými intervaly spolehlivosti do grafu:

```
> predicted.probit <- predict(m1.probit, type = "link",
    newdata = data, se = T)
> data$CI.lower.probit <- pnorm(predicted.probit$fit -
    1.96 * predicted.probit$se.fit)
> data$fitted.probit <- pnorm(predicted.probit$fit)
> data$CI.higher.probit <- pnorm(predicted.probit$fit +
    1.96 * predicted.probit$se.fit)
> x <- c(data$Infection_Severity, rev(data$Infection_Severity))
> y <- c(data$CI.lower.probit, rev(data$CI.higher.probit))
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, type = "n", pch = 3, ylab = "Outcome", xlab = "Severity",
    main = "Probitová regrese")
> polygon(x, y, col = "gray85", border = "gray85")
> points(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, pch = 3, ylab = "Outcome", xlab = "Severity",
    col = "gray35")
> lines(data$Infection_Severity, data$fitted.probit, col = "darkgreen",
    lwd = 2)
> points(data$Infection_Severity, data$fitted.probit,
    col = "darkgreen")
```



Obrázek 4: Probitová regrese s intervaly spolehlivosti.

Nakonec uvažujme poslední možnost, a to komplementární log-log linkovací funkci. GLM model pro binární proměnnou budeme opět konstruovat pro jedinou kovariátu `Infection_Severity`.

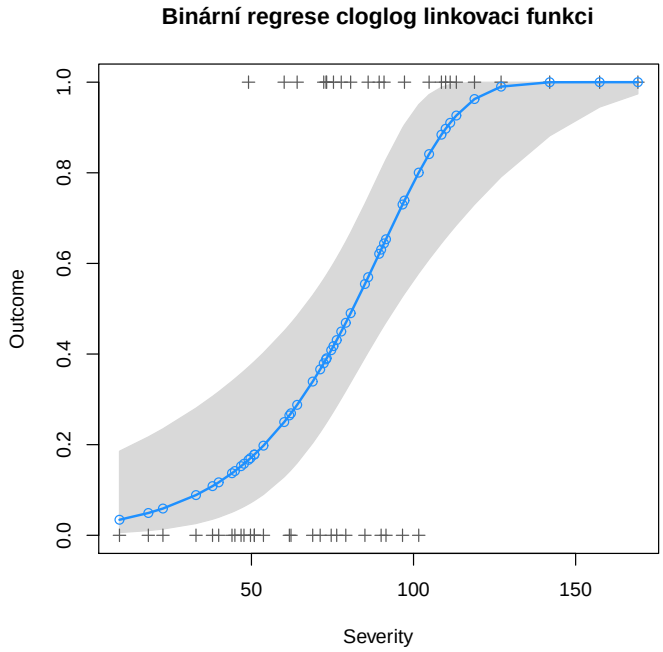
```
> m1.cloglog <- glm(Treatment_Outcome ~ Infection_Severity,
    family = binomial(cloglog), data = data)
> summary(m1.cloglog)
```

```
Call:
glm(formula = Treatment_Outcome ~ Infection_Severity, family = binomial(cloglog),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7952  -0.6639  -0.3177   0.7782   1.8936

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.74248    1.00860  -3.711 0.000207 ***
Infection_Severity  0.04153    0.01159   3.582 0.000341 ***
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 45.964  on 47  degrees of freedom
AIC: 49.964

Number of Fisher Scoring iterations: 6
```

I v tomto modelu je kovariáta `Infection_Severity` statisticky významná.

Stejně jako v předchozích případech vykreslíme výslednou křivku spolu s asymptotickými intervaly spolehlivosti:

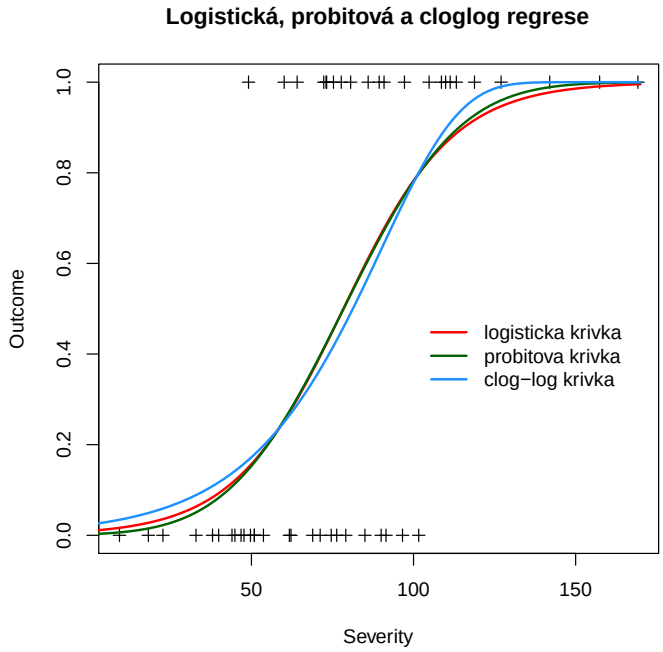
```
> Icloglog <- function(x) return(1 - exp(-exp(x)))
> predicted.cloglog <- predict(m1.cloglog, type = "link",
    newdata = data, se = T)
> data$CI.lower.cloglog <- Icloglog(predicted.cloglog$fit -
    1.96 * predicted.cloglog$se.fit)
> data$fitted.cloglog <- Icloglog(predicted.cloglog$fit)
> data$CI.higher.cloglog <- Icloglog(predicted.cloglog$fit +
    1.96 * predicted.cloglog$se.fit)
> x <- c(data$Infection_Severity, rev(data$Infection_Severity))
> y <- c(data$CI.lower.cloglog, rev(data$CI.higher.cloglog))
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, type = "n", pch = 3, ylab = "Outcome", xlab = "Severity",
    main = "Binární regrese cloglog linkovaci funkci")
> polygon(x, y, col = "gray85", border = "gray85")
> points(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, pch = 3, ylab = "Outcome", xlab = "Severity",
    col = "gray35")
> lines(data$Infection_Severity, data$fitted.cloglog,
    col = "dodgerblue", lwd = 2)
> points(data$Infection_Severity, data$fitted.cloglog,
    col = "dodgerblue")
```



Obrázek 5: Binární regrese s komplementární log-log linkovací funkcí s intervaly spolehlivosti.

Nyní zakreslíme všechny křivky do jediného grafu a aby byl výsledný graf kvalitnější, nepoužijeme předchozí odhady s 49 body, ale síť pro x–ové hodnoty zjemníme.

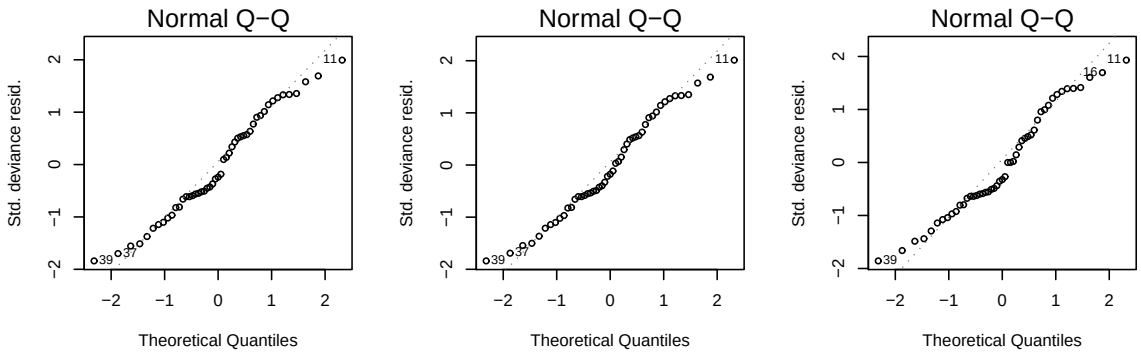
```
> xx <- seq(0, 170, length.out = 200)
> yy.logit <- predict(m1.logit, list(Infection_Severity = xx),
    type = "response")
> yy.probit <- predict(m1.probit, list(Infection_Severity = xx),
    type = "response")
> yy.cloglog <- predict(m1.cloglog, list(Infection_Severity = xx),
    type = "response")
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, pch = 3, ylab = "Outcome", xlab = "Severity",
    main = "Logistická, probitová a cloglog regrese")
> lines(xx, yy.logit, col = "red", lwd = 2)
> lines(xx, yy.probit, col = "darkgreen", lwd = 2)
> lines(xx, yy.cloglog, col = "dodgerblue", lwd = 2)
> legend(100, 0.5, bty = "n", col = c("red", "darkgreen",
    "dodgerblue"), lty = c(1, 1, 1), lwd = c(2, 2, 2),
    legend = c("logisticka krivka", "probitova krivka",
        "clog-log krivka"))
```



Obrázek 6: Porovnání všech binárních regresí.

Vhodný model se pokusíme vybrat na základě analýzy reziduí.

```
> par(mfrow = c(1, 3))
> plot(m1.logit, which = 2, cex = 0.75)
> plot(m1.probit, which = 2, cex = 0.75)
> plot(m1.cloglog, which = 2, cex = 0.75)
```



Obrázek 7: Srovnání logistické, probitové a clolog regrese pomocí Q-Q grafů.

Na základě těchto grafů nejsme schopni rozhodnout, který model je nejvhodnější. Pro binární výstupy však máme k dispozici velmi účinný grafický nástroj, který se nazývá ROC křivky.

Poznámky k ROC analýze vztahující se k binární regresi

Klasická ROC křivka je definována pro **binární klasifikační pravidla**, tj. pro pravidla jejichž výstupem jsou pouze **dvě kategorie** (**třídy**, **populace**).

Zkratka ROC je odvozená od slov *Receiver Operating Characteristic*, neboť se původně využívala jako operační charakteristika radiolokátoru.

Binární klasifikační pravidlo je **předpis** určující, zda jedinec či objekt popsaný pomocí jednorozměrného nebo i vícerozměrného statistického znaku patří do **jedné ze dvou** rozlišitelných **tříd** nebo **populací**. Výstupem klasifikačního pravidla je tedy označení jisté třídy, populace či kategorie, např.

- zdravý, nemocný;
- prospěl, neprospěl.

Jde o kvalitativní proměnnou, která se obyčejně kóduje čísly, např. 0 a 1.

V případě **binární regrese** máme k dispozici (pro $i = 1, \ldots, n$):

Y_i binární proměnné nabývajících hodnot $\{0, 1\}$

$\mathbf{x}_i = (x_{i1}, \ldots, x_{im})^{\mathrm{T}}$ nezávisle proměnné - jednorozměrný či vícerozměrný znak charakterizující jedince

Pomocí GLM modelu s vhodnou linkovací funkcí g získáme odhady

$$\hat{\pi}_i = \hat{\pi}(\mathbf{x}_i) = g^{-1}(\mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}_{\mathrm{MLE}}) \qquad \text{pro } i = 1, \ldots, n,$$

které lze v prostředí R získat například příkazem `fitted(model)`.

Pak se jako **klasifikátor** používá následující klasifikační pravidlo:

$$\widehat{Y}_i = \begin{cases} 1 & \text{aposteriorní pravděpodobnost } \hat{\pi}(\mathbf{x}_i) > c, \text{ obvykle se volí } c = 0.5, \\ 0 & \text{jinak.} \end{cases}$$

Bod $\boxed{c}$ se nazývá **dělícím** či **kritickým bodem** (*decision limit*, *cutoff point*, *threshold*).

Při hodnocení **úspěšnosti** konkrétního binárního klasifikačního pravidla (které je spojeno s přesně daným dělícím bodem) se vychází z následující **kontingenční tabulky**, nazývané též *matice záměn* či *konfusní matice*.

Kontingenční tabulka

Skutečná kategorie	Klasifikovaná kategorie	
	0 (negativní)	1 (pozitivní)
0 (negativní)	*správně negativní* *specificita* $\boxed{1-\alpha}$	*nesprávně pozitivní* *chyba 1. druhu* $\boxed{\alpha}$
1 (pozitivní)	*nesprávně negativní* *chyba 2. druhu* $\boxed{\beta}$	*správně pozitivní* *senzitivita* $\boxed{1-\beta}$

Podle toho, která chyba má závažnější důsledky, je pak možné změnit dělící bod a mít pod kontrolou velikost vybrané chyby.

Binární klasifikační pravidlo (také se mu v diagnostice říká **diagnostický test** či **testové kritérium**) je náhodnou veličinou, která v našem případě nabývá spojitých hodnot mezi nulou a jedničkou. Označme ji například symbolem T.

Dále označme symbolem T_0 náhodnou veličinu T za podmínky, že jedinec ve skutečnosti patří do skupiny 0 (*undiseased population*), obdobně označme symbolem T_1 náhodnou veličinu T za podmínky, že jedinec ve skutečnosti patří do skupiny 1 (*diseased population*). Příslušné hustoty a distribuční funkce označme symboly f_0, F_0 a f_1, F_1.

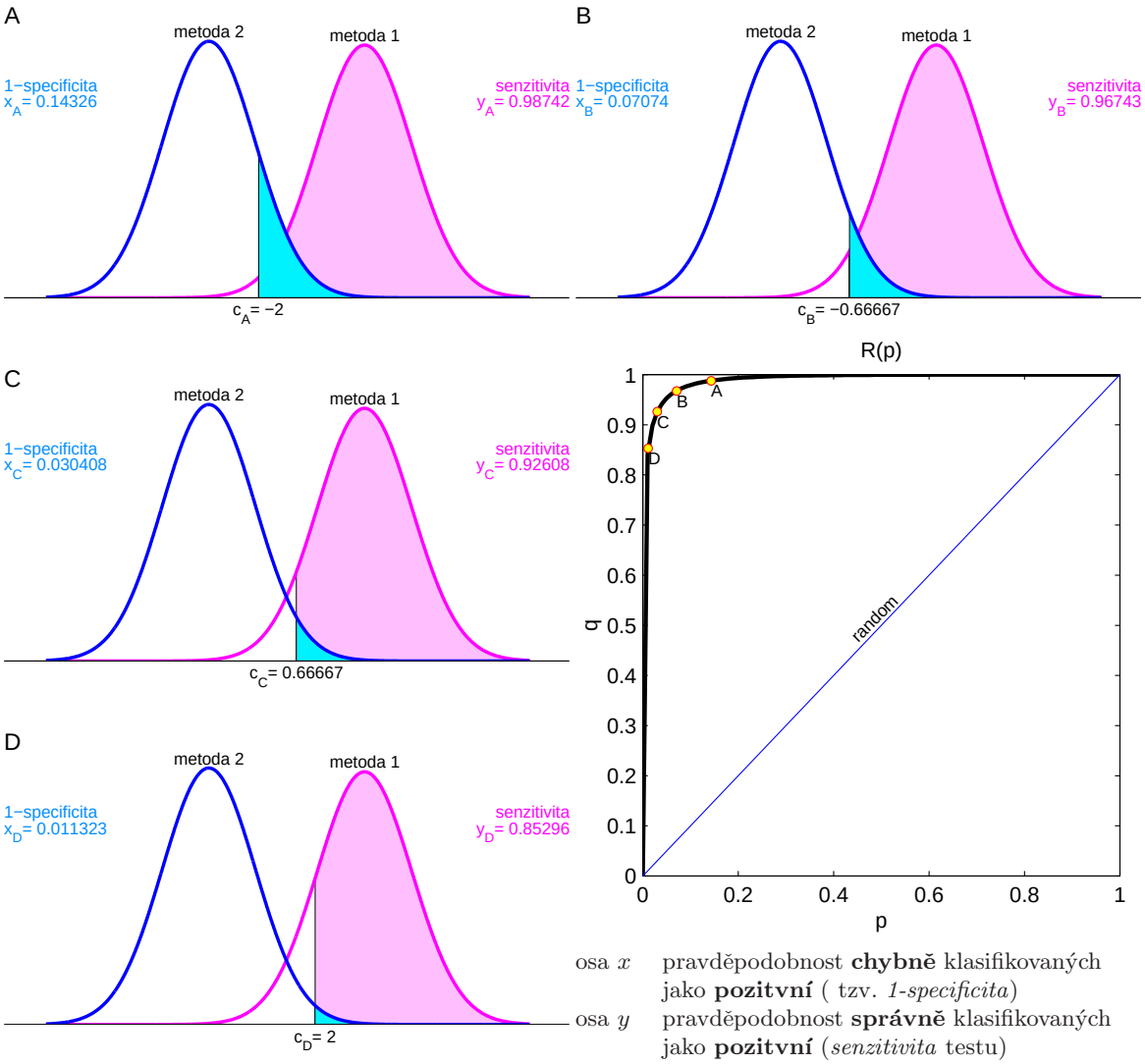
Dále budeme předpokládat, že vyšší hodnoty kritéria T vedou k vyšší pravděpodobnosti výskytu nějaké zkoumané nemoci (tj. k vyšší pravděpodobnosti, že jedinec patří do populace 1, tedy je pozitivní).

Pak pro $\forall\, c$ platí $FP(c)=P(T_0 > c) = 1 - F_0(c)$ (*false positive=1-specificity*)
$TP(c)=P(T_1 > c) = 1 - F_1(c)$ (*true positive=sensitivity*)

a *ROC* křivku tvoří dvojice bodů: $(FP(c), TP(c)) = (\underbrace{1 - F_0(c)}_{x}, \underbrace{1 - F_1(c)}_{y})$

nebo–li $ROC(p) = 1 - F_1(F_0^{-1}(1-p)) \qquad 0 \le p \le 1.$

Na následujících grafech vidíme názorně, jak se konstruuje *ROC* křivka pro měnící se dělící bod. V tomto případě metoda 2 představuje populaci s indexem 0.



osa x pravděpodobnost **chybně** klasifikovaných jako **pozitivní** (tzv. *1-specificita*)

osa y pravděpodobnost **správně** klasifikovaných jako **pozitivní** (*senzitivita* testu)

Klasifikační pravidlo je o to přesnější, čím více se *ROC* křivka přimyká k levého hornímu bodu $(0, 1)$.

Velmi důležitou charakteristikou je také plocha pod *ROC* křivkou

$$AUC = \int_0^1 ROC(p)\, dp \qquad (Area\ Under\ the\ ROC\ Curve).$$

Jestliže kovariáty charakterizované vektorem $\mathbf{x}$ nemají vliv na klasifikaci do dvou tříd, pak hodnota AUC bude 0.5. Čím vhodnější kovariáty byly zvoleny, tím více se hodnota AUC blíží k jedné.

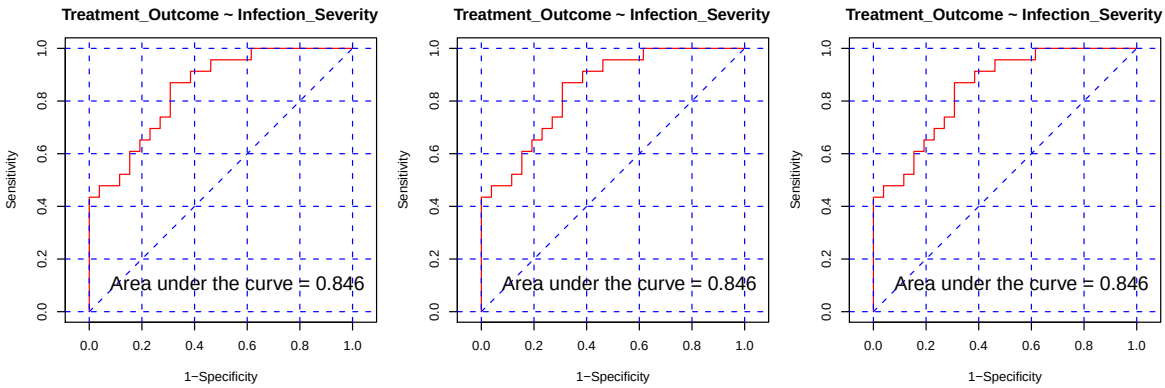
Protože skutečné rozdělení klasifikačního kritéria neznáme, musíme *ROC* křivku nějak odhadnout. Používá se celá řada přístupů:

- *parametrický* předpokládající například normalitu testového kritéria
- *neparametrický* neznámé podmíněné distribuční funkce se odhadují například pomocí jádrových odhadů.

Nejčastěji se však používá neparametrický přístup založený na **empirických distribučních funkcích**. V tom případě má odhadnutá *ROC* křivka schodovitý tvar.

V prostředí R existuje celá řada balíků, které dokáží vykreslit ROC křivku a vypočítat AUC hodnotu. Nejprve si ukážeme grafy získané z knihovy `epicalc`.

```
> library(epicalc)
> par(mfrow = c(1, 3), mar = c(5, 5, 3, 0) + 0.1)
> graf1 <- lroc(m1.logit, title = TRUE, auc.coords = c(0.05,
      0.1), cex = 1.5, cex.main = 1.25)
> graf2 <- lroc(m1.probit, title = TRUE, auc.coords = c(0.05,
      0.1), cex = 1.5, cex.main = 1.25)
> graf3 <- lroc(m1.cloglog, title = TRUE, auc.coords = c(0.05,
      0.1), cex = 1.5, cex.main = 1.25)
```

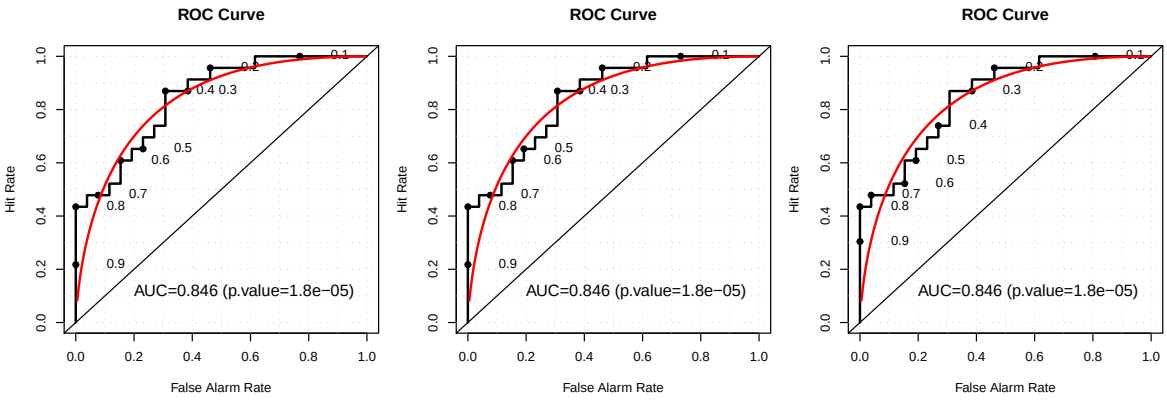


Obrázek 8: Porovnání ROC křivek a hodnot AUC pro logistickou, probitovou a cloglog binární regresi (pomocí příkazu `lroc` z knihovny `epicalc`).

Z výsledných grafů je patrné, že ani v ROC křivce, ani v AUC hodnotě se metody neliší.

Ještě si ukážeme, jaký typ grafu pro ROC křivku nabízí knihovna `verification`. V tomto případě vedle empirického odhadu lze získat také ROC křivku, která se z výchozích dat odhadne za předpokladu, že T_0 i T_1 mají normální rozdělení.

```
> library(verification)
> par(mfrow = c(1, 3), mar = c(5, 5, 3, 0) + 0.1)
> binvar <- unclass(data$Treatment_Outcome) - 1
> T <- fitted(m1.logit)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    6), ")", sep = "")
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
> T <- fitted(m1.probit)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    6), ")", sep = "")
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
> T <- fitted(m1.cloglog)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    6), ")", sep = "")
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
```



Obrázek 9: Porovnání ROC křivek a hodnot AUC pro logistickou, probitovou a cloglog binární regresi (pomocí příkazů `roc.area` a `roc.plot` z knihovny `verification`).

P–hodnota v závorce u AUC hodnoty se vztahuje k testování hypotézy $H_0 : AUC = 0.5$. Vidíme, že tuto hypotézu zamítáme, což značí že proměnná `Infection_Severity` má významný vliv na přežití.

Podívejme se, jak dopadly konfusní matice pro jednotlivé binární modely

```
> fitY.logit <- factor(fitted(m1.logit) > 0.5, labels = c("pred.survived",
    "pred.died"))
> table(data$Treatment_Outcome, fitY.logit)
```

```
          fitY.logit
           pred.survived pred.died
  survived            20         6
  died                 8        15
```

```
> fitY.probit <- factor(fitted(m1.probit) > 0.5, labels = c("pred.survived",
    "pred.died"))
> table(data$Treatment_Outcome, fitY.probit)
```

```
          fitY.probit
           pred.survived pred.died
  survived            20         6
  died                 8        15
```

```
> fitY.cloglog <- factor(fitted(m1.cloglog) > 0.5, labels = c("pred.survived",
    "pred.died"))
> table(data$Treatment_Outcome, fitY.cloglog)
```

```
          fitY.cloglog
           pred.survived pred.died
  survived            21         5
  died                 9        14
```

Vidíme, že konfusní matice jsou všechny stejné. Ukážeme si dále, jak lze místo absolutních četností získat relativní četnosti. Nejprve pro celou tabulku

```
> prop.table(table(data$Treatment_Outcome, fitY.logit))
```

```
          fitY.logit
           pred.survived pred.died
  survived     0.4081633 0.1224490
  died         0.1632653 0.3061224
```

Relativní četnosti podle řádků dostaneme příkazem

```
> prop.table(table(data$Treatment_Outcome, fitY.probit), 1)
```

```
          fitY.probit
           pred.survived pred.died
  survived     0.7692308 0.2307692
  died         0.3478261 0.6521739
```

A nakonec relativní četnosti podle sloupců dostaneme takto

```
> prop.table(table(data$Treatment_Outcome, fitY.cloglog), 2)
```

```
          fitY.cloglog
           pred.survived pred.died
  survived     0.7000000 0.2631579
  died         0.3000000 0.7368421
```

Mnohem více možností máme, pokud použijeme příkaz `CrossTable()` z knihovny `gmodels`.

```
> library(gmodels)
> CrossTable(table(data$Treatment_Outcome, fitY.cloglog), prop.r = T,
     prop.c = T, prop.t = T, prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  49


             | fitY.cloglog
             | pred.survived |     pred.died |     Row Total |
-------------|---------------|---------------|---------------|
    survived |            21 |             5 |            26 |
             |         0.808 |         0.192 |         0.531 |
             |         0.700 |         0.263 |               |
             |         0.429 |         0.102 |               |
-------------|---------------|---------------|---------------|
        died |             9 |            14 |            23 |
             |         0.391 |         0.609 |         0.469 |
             |         0.300 |         0.737 |               |
             |         0.184 |         0.286 |               |
-------------|---------------|---------------|---------------|
Column Total |            30 |            19 |            49 |
             |         0.612 |         0.388 |               |
-------------|---------------|---------------|---------------|
```

Nyní se vrátíme k binární regresi a zjistíme, zda se model nezlepší, jestliže přidáme další vysvětlující proměnnou, a to proměnnou `Hospital`.

Model 2 - binární regresní model s jedinou spojitou proměnnou `Infection_Severity` a kategoriální proměnnou `Hospital`

Začneme s modelem, ve kterém budeme uvažovat i interakce mezi proměnnými `Infection_Severity` a `Hospital`. Nejprve zvolíme kanonickou linkovací funkci.

```
> m2a.logit <- glm(Treatment_Outcome ~ Infection_Severity *
    Hospital, family = binomial(logit), data = data)
> summary(m2a.logit)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity * Hospital,
    family = binomial(logit), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2565  -0.4597  -0.1565   0.3539   2.0989

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -5.664355   2.803449  -2.020   0.0433 *
Infection_Severity          0.055830   0.032493   1.718   0.0858 .
HospitalB                  -0.012466   3.989376  -0.003   0.9975
HospitalC                   2.817310   3.600700   0.782   0.4340
Infection_Severity:HospitalB  0.012591   0.049197   0.256   0.7980
Infection_Severity:HospitalC  0.006657   0.046875   0.142   0.8871
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 34.676  on 43  degrees of freedom
AIC: 46.676

Number of Fisher Scoring iterations: 6
```

Vidíme, že významnost jednotlivých proměnných se výrazně zhoršila, proto uvažujme jednodušší model bez interakcí

```
> m2b.logit <- glm(Treatment_Outcome ~ Infection_Severity +
    Hospital, family = binomial(logit), data = data)
> summary(m2b.logit)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity + Hospital,
    family = binomial(logit), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2528  -0.4932  -0.1835   0.3643   2.1508

Coefficients:
```

```
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -6.18858    1.81807  -3.404 0.000664 ***
Infection_Severity  0.06209    0.01985   3.128 0.001760 **
HospitalB           0.98306    1.01251   0.971 0.331595
HospitalC           3.36626    1.20231   2.800 0.005113 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 34.742  on 45  degrees of freedom
AIC: 42.742

Number of Fisher Scoring iterations: 6
```

Ještě zkontrolujme, zda nedošlo k výraznému zhoršení tohoto modelu oproti předchozímu

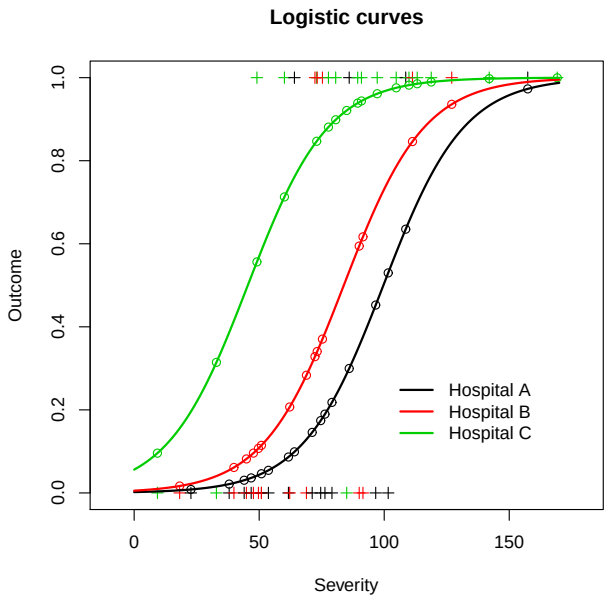
```
> anova(m2a.logit, m2b.logit, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: Treatment_Outcome ~ Infection_Severity * Hospital
Model 2: Treatment_Outcome ~ Infection_Severity + Hospital
  Resid. Df Resid. Dev Df  Deviance P(>|Chi|)
1        43     34.676
2        45     34.742 -2 -0.066078    0.9675
```

Protože P-hodnota není menší než 0.05, vypuštěním interakcí nedošlo k výraznému zhoršení modelu.

Provedeme vykreslení výsledných logistických křivek pro jednotlivé nemocnice.

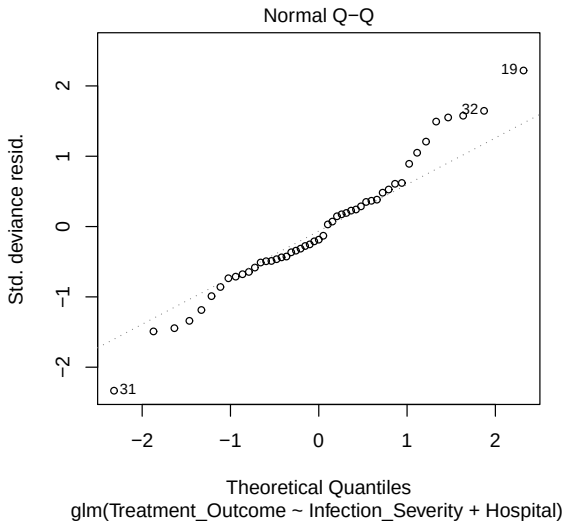
```
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
    1, type = "p", pch = 3, xlim = c(-10, 180), col = unclass(data$Hospital),
    ylab = "Outcome", xlab = "Severity", main = "Logistic curves")
> points(data$Infection_Severity, fitted(m2b.logit), col = unclass(data$Hospital))
> xx <- seq(0, 170, length.out = 200)
> yA.logit <- predict(m2b.logit, list(Infection_Severity = xx,
    Hospital = factor(rep("A", 200))), type = "response")
> lines(xx, yA.logit, col = 1, lwd = 2)
> yB.logit <- predict(m2b.logit, list(Infection_Severity = xx,
    Hospital = factor(rep("B", 200))), type = "response")
> lines(xx, yB.logit, col = 2, lwd = 2)
> yC.logit <- predict(m2b.logit, list(Infection_Severity = xx,
    Hospital = factor(rep("C", 200))), type = "response")
> lines(xx, yC.logit, col = 3, lwd = 2)
> legend(100, 0.3, bty = "n", col = c(1, 2, 3), lty = c(1,
    1, 1), lwd = c(2, 2, 2), legend = c("Hospital A", "Hospital B",
    "Hospital C"))
```



Obrázek 10: Logistické křivky pro jednotlivé nemocnice.

Pro tento model opět provedeme grafickou analýzu reziduí a vykreslíme ROC křivku.

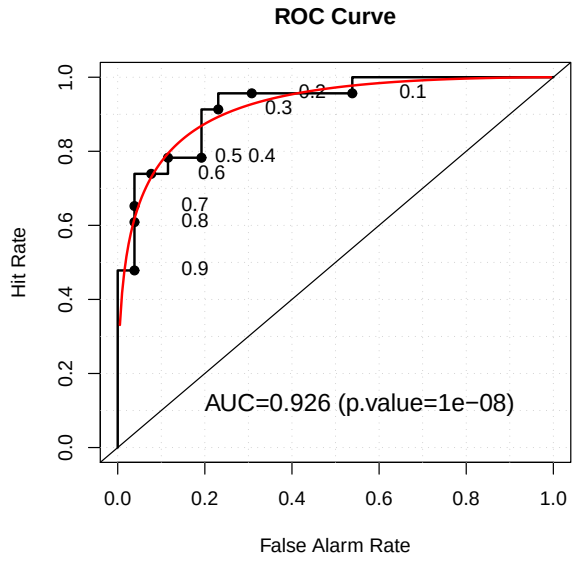
```
> plot(m2b.logit, which = 2, cex = 0.75)
```



Obrázek 11: Q-Q graf reziduí modelu 2b s logit linkovací funkcí.

```
> library(verification)
> par(mar = c(5, 5, 3, 0) + 0.1)
> binvar <- unclass(data$Treatment_Outcome) - 1
> T <- fitted(m2b.logit)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    8), ")", sep = "")
```

```
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
```



Obrázek 12: ROC křivka a hodnota AUC pro logistickou binární regresi – MODEL 2B (pomocí příkazů `roc.area` a `roc.plot` z knihovny `verification`).

Vidíme, že přidáním další proměnné se hodnota AUC z 0.846 zvedla na 0.926. Nezapomeneme také na konfusní matici:

```
> library(gmodels)
> fitY.logit <- factor(fitted(m2b.logit) > 0.5, labels = c("pred.survived",
     "pred.died"))
> CrossTable(table(data$Treatment_Outcome, fitY.cloglog), prop.r = T,
     prop.c = T, prop.t = T, prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  49


             | fitY.cloglog
             | pred.survived |     pred.died |     Row Total |
-------------|---------------|---------------|---------------|
    survived |            21 |             5 |            26 |
             |         0.808 |         0.192 |         0.531 |
             |         0.700 |         0.263 |               |
             |         0.429 |         0.102 |               |
-------------|---------------|---------------|---------------|
```

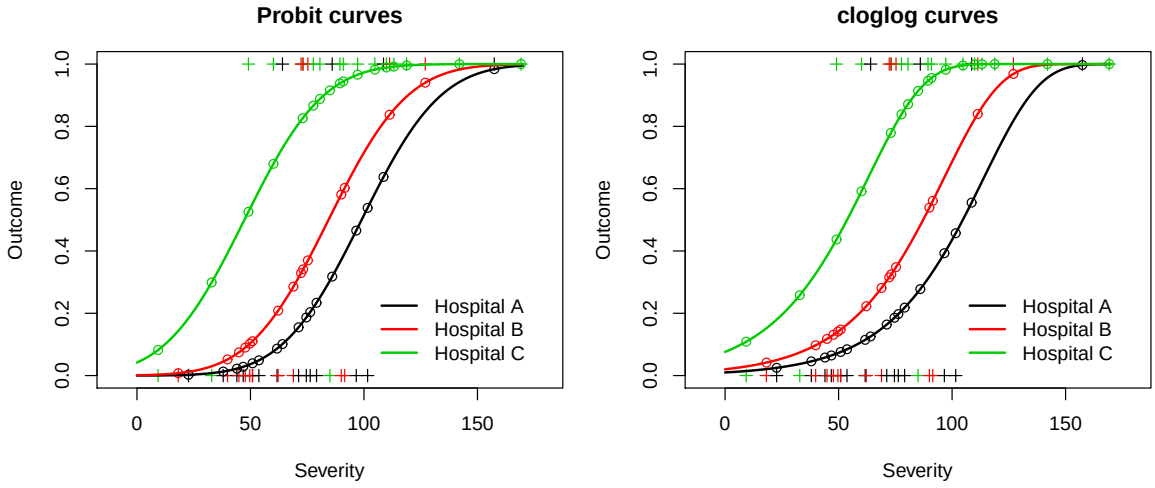
```
        died |             9 |            14 |            23 |
             |         0.391 |         0.609 |         0.469 |
             |         0.300 |         0.737 |               |
             |         0.184 |         0.286 |               |
-------------|---------------|---------------|---------------|
Column Total |            30 |            19 |            49 |
             |         0.612 |         0.388 |               |
-------------|---------------|---------------|---------------|
```

Pro úplnost spočítejme Model 2b pro zbývající dvě linkovací funkce. Vypočteme model, do jednoho grafu zakreslíme výsledné křivky pro všechny nemocnice, provedeme grafickou analýzu reziduí, vytvoříme ROC křivku a nakonec vypočteme konfusní matici.

```
> m2b.probit <- glm(Treatment_Outcome ~ Infection_Severity +
     Hospital, family = binomial(probit), data = data)
> summary(m2b.probit)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity + Hospital,
    family = binomial(probit), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2245  -0.4837  -0.1249   0.3579   2.1404

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.61785    0.95185  -3.801 0.000144 ***
Infection_Severity  0.03655    0.01058   3.454 0.000552 ***
HospitalB           0.53274    0.57472   0.927 0.353948
HospitalC           1.88787    0.64180   2.942 0.003266 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 34.511  on 45  degrees of freedom
AIC: 42.511

Number of Fisher Scoring iterations: 7


> m2b.cloglog <- glm(Treatment_Outcome ~ Infection_Severity +
     Hospital, family = binomial(cloglog), data = data)
> summary(m2b.cloglog)


Call:
glm(formula = Treatment_Outcome ~ Infection_Severity + Hospital,
    family = binomial(cloglog), data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2131  -0.5651  -0.2920   0.3321   2.0368
```

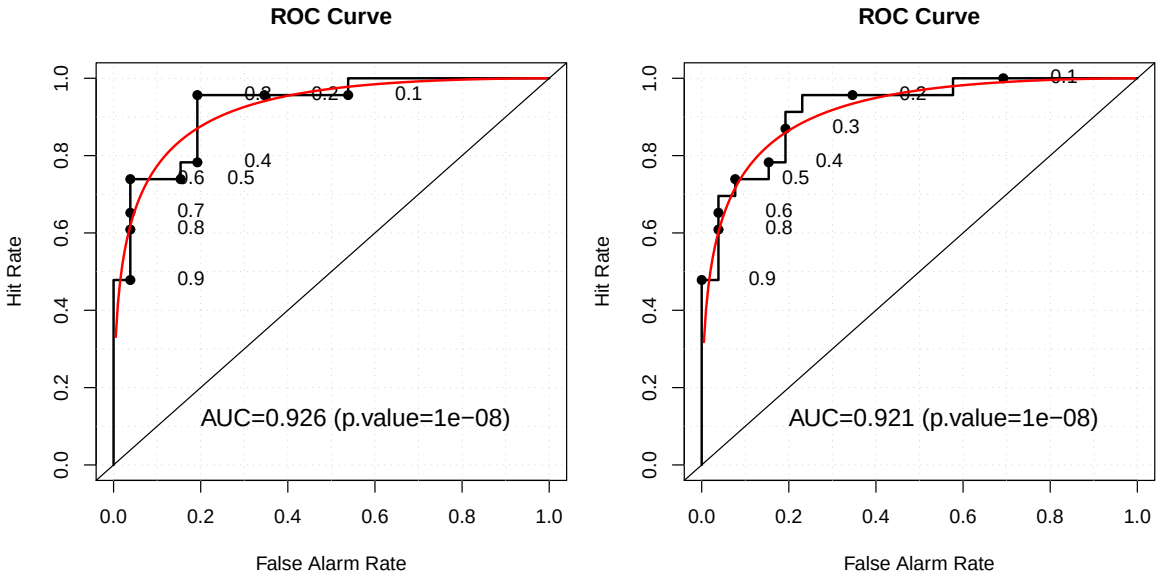
```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.59689    1.23198  -3.731 0.000191 ***
Infection_Severity  0.04039    0.01258   3.210 0.001329 **
HospitalB           0.70634    0.75040   0.941 0.346561
HospitalC           2.05929    0.73198   2.813 0.004903 **
---
Signif. codes:  0 ,***, 0.001 ,**, 0.01 ,*, 0.05 ,., 0.1 , , 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 67.745  on 48  degrees of freedom
Residual deviance: 35.482  on 45  degrees of freedom
AIC: 43.482

Number of Fisher Scoring iterations: 7
```

```
> par(mfrow = c(1, 2), mar = c(5, 5, 3, 0) + 0.1)
> xx <- seq(0, 170, length.out = 200)
> yA.probit <- predict(m2b.probit, list(Infection_Severity = xx,
     Hospital = factor(rep("A", 200))), type = "response")
> yB.probit <- predict(m2b.probit, list(Infection_Severity = xx,
     Hospital = factor(rep("B", 200))), type = "response")
> yC.probit <- predict(m2b.probit, list(Infection_Severity = xx,
     Hospital = factor(rep("C", 200))), type = "response")
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
     1, type = "p", pch = 3, xlim = c(-10, 180), col = unclass(data$Hospital),
     ylab = "Outcome", xlab = "Severity", main = "Probit curves")
> points(data$Infection_Severity, fitted(m2b.probit), col = unclass(data$Hospital))
> lines(xx, yA.probit, col = 1, lwd = 2)
> lines(xx, yB.probit, col = 2, lwd = 2)
> lines(xx, yC.probit, col = 3, lwd = 2)
> legend(100, 0.3, bty = "n", col = c(1, 2, 3), lty = c(1,
     1, 1), lwd = c(2, 2, 2), legend = c("Hospital A", "Hospital B",
     "Hospital C"))
> plot(data$Infection_Severity, unclass(data$Treatment_Outcome) -
     1, type = "p", pch = 3, xlim = c(-10, 180), col = unclass(data$Hospital),
     ylab = "Outcome", xlab = "Severity", main = "cloglog curves")
> points(data$Infection_Severity, fitted(m2b.cloglog), col = unclass(data$Hospital))
> yA.cloglog <- predict(m2b.cloglog, list(Infection_Severity = xx,
     Hospital = factor(rep("A", 200))), type = "response")
> yB.cloglog <- predict(m2b.cloglog, list(Infection_Severity = xx,
     Hospital = factor(rep("B", 200))), type = "response")
> yC.cloglog <- predict(m2b.cloglog, list(Infection_Severity = xx,
     Hospital = factor(rep("C", 200))), type = "response")
> lines(xx, yA.cloglog, col = 1, lwd = 2)
> lines(xx, yB.cloglog, col = 2, lwd = 2)
> lines(xx, yC.cloglog, col = 3, lwd = 2)
> legend(100, 0.3, bty = "n", col = c(1, 2, 3), lty = c(1,
     1, 1), lwd = c(2, 2, 2), legend = c("Hospital A", "Hospital B",
     "Hospital C"))
```



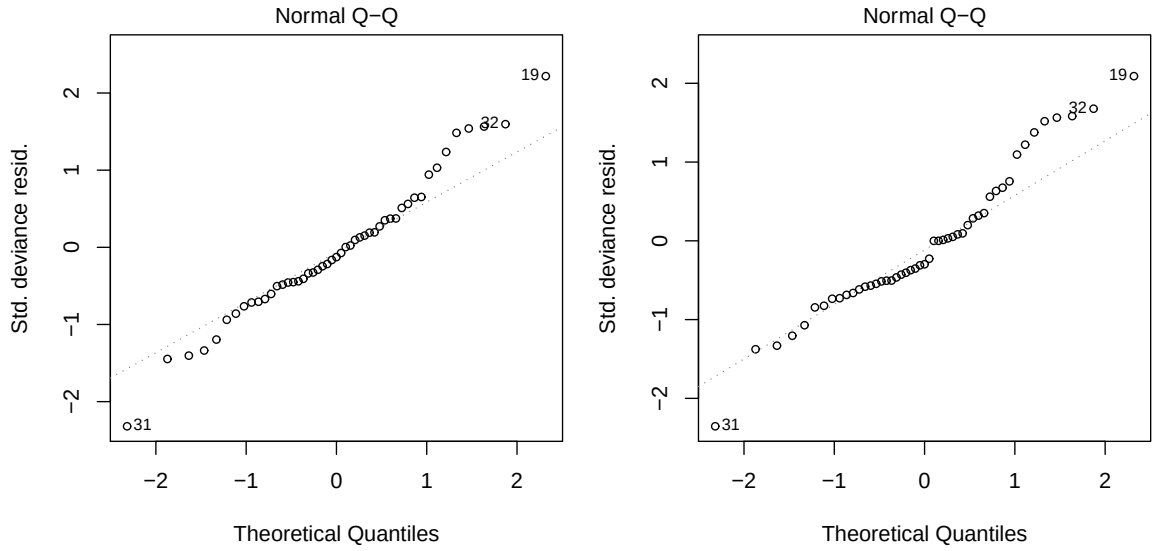
Obrázek 13: Výsledné křivky: probit a komplementární loglog

```
> par(mfrow = c(1, 2))
> binvar <- unclass(data$Treatment_Outcome) - 1
> T <- fitted(m2b.probit)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    8), ")", sep = "")
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
> T <- fitted(m2b.cloglog)
> AUC <- roc.area(binvar, T)
> auc.txt <- paste("AUC=", round(AUC$A, 3), " (p.value=", round(AUC$p.value,
    8), ")", sep = "")
> roc.plot(binvar, T, binormal = T, plot = "both")
> text(0.2, 0.1, auc.txt, adj = c(0, 0), cex = 1.25)
```



Obrázek 14: ROC křivky pro modely s probit a komplementární loglog linkovací funkcí

```
> par(mfrow = c(1, 2), mar = c(5, 5, 3, 0) + 0.1)
> plot(m2b.probit, which = 2, cex = 0.75)
> plot(m2b.cloglog, which = 2, cex = 0.75)
```



Obrázek 15: Q-Q grafy reziduí modelu 2b s linkovacími funkcemi probit a cloglog.

```
> library(gmodels)
> fitY.probit <- factor(fitted(m2b.probit) > 0.5, labels = c("pred.survived",
    "pred.died"))
> CrossTable(table(data$Treatment_Outcome, fitY.cloglog), prop.r = T,
    prop.c = T, prop.t = T, prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  49


             | fitY.cloglog
             | pred.survived |     pred.died |     Row Total |
-------------|---------------|---------------|---------------|
    survived |            21 |             5 |            26 |
             |         0.808 |         0.192 |         0.531 |
             |         0.700 |         0.263 |               |
             |         0.429 |         0.102 |               |
-------------|---------------|---------------|---------------|
```

```
        died |             9 |            14 |            23 |
             |         0.391 |         0.609 |         0.469 |
             |         0.300 |         0.737 |               |
             |         0.184 |         0.286 |               |
-------------|---------------|---------------|---------------|
Column Total |            30 |            19 |            49 |
             |         0.612 |         0.388 |               |
-------------|---------------|---------------|---------------|
```

```
> fitY.cloglog <- factor(fitted(m2b.cloglog) > 0.5, labels = c("pred.survived",
     "pred.died"))
> CrossTable(table(data$Treatment_Outcome, fitY.cloglog), prop.r = T,
     prop.c = T, prop.t = T, prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  49

             | fitY.cloglog
             | pred.survived |     pred.died |     Row Total |
-------------|---------------|---------------|---------------|
    survived |            23 |             3 |            26 |
             |         0.885 |         0.115 |         0.531 |
             |         0.793 |         0.150 |               |
             |         0.469 |         0.061 |               |
-------------|---------------|---------------|---------------|
        died |             6 |            17 |            23 |
             |         0.261 |         0.739 |         0.469 |
             |         0.207 |         0.850 |               |
             |         0.122 |         0.347 |               |
-------------|---------------|---------------|---------------|
Column Total |            29 |            20 |            49 |
             |         0.592 |         0.408 |               |
-------------|---------------|---------------|---------------|
```