

Knjižnica rms in nekaj napotkov za modeliranje efektov v regresijskih modelih

Nina Ružić Gorenjec

april 2021

1 Motivacija, podatki

- Podatki iz obsežne študije o odraslih, ki živijo doma, vendar v zadnjih petih letih niso obiskali svojega osebnega zdravnika.
- Izid (ang. *outcome*) pri tej analizi: nesamostojnost (da/ne)
- Članek na podlagi analize: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0245465>
- Določili smo naslednjih 9 pomembnih neodvisnih spremenljivk za nesamostojnost (0 = ne, 1 = da), za katere smo naredili model multiple logistične regresije:
 - spol: 1 = moški, 2 = ženski
 - starost: leta
 - itm: indeks telesne mase 4 kategorijami (underweight, normal, overweight, obese)
 - kronične bolezni: 0, 1, 2, 3 ali več
 - family function: 1 = dobra, 2 = slaba
 - povišano tveganje za padec: 0 = ne, 1 = da
 - splošna ocena trenutnega zdravja: 1-10
 - splošna ocena občutenja osamljenosti: 1-10
 - splošna ocena jakosti prisotne bolečine: 1-10
- Skupaj 1999 pacientov, od tega jih je imelo 1814 popolne podatke glede na zgornje spremenljivke.
- Tekom vaj obravnavamo podvzorec 1500 pacientov.

```
source("podatki.R")
```

Že narejeno:

- Čiščenje podatkov (ste obravnavali na predavanjih)
- Izbor neodvisnih spremenljivk:
 - Background knowledge - uporabili tu
 - Variable selection
 - * uporabili pri izidu tveganje za podhranjenost na podobnih podatkih: <https://www.sciencedirect.com/science/article/abs/pii/S0899900719302400>
 - * kontroverzna tematika, razvijajoča se, več o tem npr. v članku Heinze, Dunkler, *Five myths about variable selection*, <https://onlinelibrary.wiley.com/doi/full/10.1111/tri.12895>

Cilj: Prilagoditi model vsem 9 spremenljivkam, kjer moramo preveriti:

- Ali moramo številске spremenljivke vključiti nelinearno? Katere funkcije uporabiti?
- Ali je potrebno vključiti še interakcije med spremenljivkami?
- Ali je model dober?

Delo nam lahko olajša knjižnica rms priznanega statistika prof. dr. Franka Harella: <https://hbiostat.org/R/rms/>

Več o modeliranju v knjigi F. Harella *Regression Modeling Strategies*, izdana pri Springerju, skripta: <https://hbiostat.org/doc/rms.pdf>

Zanimiv njegov blog: <https://www.fharrell.com/>

2 Vrednotenje učinkov v regresijskih modelih - grafično in preko vrednosti p

Več o tem in še nekaj podrobnosti o knjižnici rms: [http://ims.mf.uni-lj.si/archive/25\(1-2\)e/03.pdf](http://ims.mf.uni-lj.si/archive/25(1-2)e/03.pdf)

V motivacijskih podatkih je izid binaren/dihotomen – logistična regresija.

Vsebinsko si pogledamo nelinearne efekte in interakcije raje na primeru linearne regresije – številski izid. Ob tem spoznamo osnove knjižnice rms.

```
library(rms) #ze vkljucuje ggplot2, lattice, plotly, idr.  
library(gridExtra) #za prikaz vec grafov na eni sliki z ggplot2
```

Uporabimo simulirane podatke, izide (več primerov) bomo generirali iz linearnega normalnega modela, pri tem bo vedno veljalo:

- velikost vzorca: $n = 200$
- napaka: $\epsilon \sim N(0, \sigma^2)$, $\sigma = 3$
- $X_1, X_2 \sim N(0, 1)$
- $Z_1 \sim \text{Bern}(0.2)$, $Z_2 \sim \text{Bern}(0.2) + 1$

```
# Generiramo neodvisne spremenljivke in napako normalnega modela  
set.seed(6)  
n <- 200  
x1 <- rnorm(n)  
x2 <- rnorm(n)  
z1 <- rbinom(n, size = 1, prob = 0.2)  
z2 <- rbinom(n, size = 1, prob = 0.2)+1  
epsilon <- rnorm(n, mean = 0, sd = 3)
```

Potrebno za delovanje ukazov iz rms:

```
# Prizmete vrednosti neodvisnih spremenljivk shranimo v datadist v options  
podatki <- data.frame(x1 = x1, x2 = x2, z1 = factor(z1), z2 = factor(z2))  
dd <- datadist(podatki)  
options(datadist = "dd")  
  
dd
```

```
##              x1          x2    z1    z2
## Low:effect   -0.71977454 -0.68386339 <NA> <NA>
## Adjust to    -0.05258683 -0.09596329    0    1
## High:effect    0.60810572  0.71639893 <NA> <NA>
## Low:prediction -1.61677862 -1.56601459    0    1
## High:prediction 1.62149559  1.48126280    1    2
## Low          -2.39744569 -3.03974046    0    1
## High         2.60809809  2.10345687    1    2
##
## Values:
##
## z1 : 0 1
## z2 : 1 2
```

Adjust to: na katero vrednost bodo nastavljene spremenljivke, ki bodo pri napovedi fiksne – funkcija `datadist` prednastavljeno uporabi modus za opisne spremenljivke (tiste, ki so shranjene kot `factor`) in mediano za številske.

Low:prediction, *High:prediction*: na katerem razponu bodo izračunane napovedi za številske spremenljivke, ki bo pri napovedi variirala – funkcija `datadist` avtomatsko nastavi 5. in 95. percentil pri velikosti vzorca 200 ali manj, za večje vzorce pa 10. najmanjšo in največjo vrednost.

2.1 Nelinearen efekt številskih spremenljivk v regresijskih modelih

Naj bo izid Y je generiran iz linearnega modela

$$Y = 1 + X_1 + X_1^2 + X_2 + Z_1 + Z_2 + 3X_2Z_1 + 3X_2Z_2 + \epsilon.$$

```
# Generiramo izid prvega modela (Y)
y <- 1 + 1*x1 + 1*x1**2 + x2 + z1 + z2 + 3*x2*z1 + 3*x2*z2 + epsilon
```

Za analizo imamo v R na voljo (vsaj) dve varianti:

- Osnovna knjižnica **stats**:
 - linearni model: `lm`
 - izračun napovednih vrednosti: `predict`
 - vrednosti p celih spremenljivk: v nekaterih primerih potrebno “ročno” uporabiti `anova`
- Knjižnica **rms**:
 - linearni model: `ols`
 - izračun napovednih vrednosti: `Predict`, uporabi vedno raje `rms::Predict` zaradi istega ukaza v knjižnici `car`
 - vrednosti p celih spremenljivk: `anova` (za objekt `ols` je to natančneje `anova.rms`)

2.1.1 Preko osnovnih knjižnic:

```
fit_lm <- lm(y ~ pol(x1,2) + x2*z1 + x2*z2, data = podatki)
summary(fit_lm)

# Ali je značilna celotna spremenljivka x1?
anova(fit_lm)
```

```

# Ali je značilna celotna spremenljivka z1?
fit_brez_z1 <- lm(y ~ pol(x1,2) + x2*z2, data = podatki)
summary(fit_brez_z1)
anova(fit_lm, fit_brez_z1)

# Narišimo si napovedne vrednosti y glede na različne x1,
# kjer so preostale spremenljivke fiksirane na svojih medianah in modusih:
x1predict = seq(min(podatki$x1), max(podatki$x1), 0.01)
newdata <- data.frame(x1 = x1predict,
                      x2 = rep(quantile(podatki$x2, probs = 0.5), length(x1predict)),
                      z1 = rep("0", length(x1predict)),
                      z2 = rep("1", length(x1predict)))
stats_predict <- predict(fit_lm, newdata = newdata, interval = "confidence")

head(stats_predict)

stats_predict <- as.data.frame(cbind(x1predict, stats_predict))

ggplot(stats_predict, aes(x = x1predict, y = fit)) +
  geom_line() +
  geom_ribbon(aes(ymin = lwr, ymax = upr), alpha = 0.3)

```

2.1.2 Preko knjižnice rms:

```

fit <- ols(y ~ pol(x1,2) + x2*z1 + x2*z2, data = podatki)
fit

# Ali je značilna celotna spremenljivka x1?
# Ali je značilna celotna spremenljivka z1?
anova(fit)

# Narišimo si napovedne vrednosti y glede na različne x1,
# kjer so preostale spremenljivke fiksirane na svojih medianah in modusih:
predict1 = rms::Predict(fit, x1)
ggplot(predict1)

# Možna je tudi uporaba drugih knjižnic za graficne prikaze
plot(predict1) # knjižnica lattice

plotp(predict1) # knjižnica plotly

# Kaj pa, če želim preostale spremenljivke fiksirati na neko drugo vrednost?
predict2 <- rms::Predict(fit, x1, z1 = 1, z2 = 2)
ggplot(predict2)

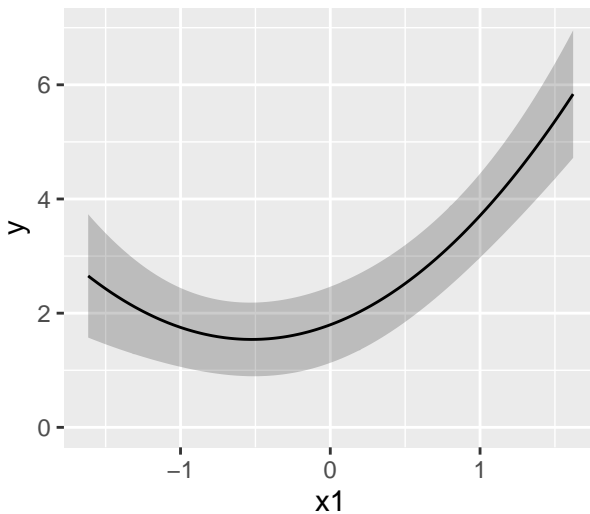
```

2.1.3 Nazaj k vsebini

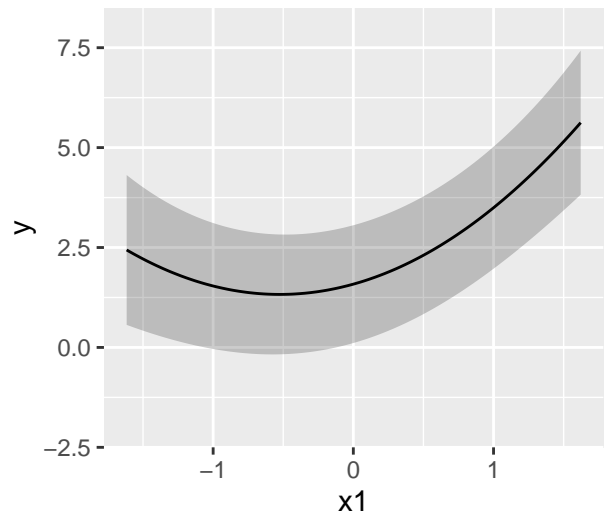
Ovrednotimo, ali je nelinearen efekt v obliki kvadratne funkcije potreben? (Ker vemo, kako smo podatke simulirali, pravi odgovor poznamo.)

- Preko vrednosti p : $p < 0.0001 \implies$ kvadraten efekt je statistično značilen (zelo)
- Grafično...

```
grid.arrange(ggplot(predict1), ggplot(predict2), ncol = 2)
```



Adjusted to: x2=-0.09596 z1=0 z2=1



Adjusted to: x2=-0.09596

Nauk: Pazi pri grafičnem vrednotenju nelinearnih efektov v multipli regresiji.

2.2 Interakcija med neodvisnimi spremenljivkami v regresijskih modelih

Imamo dva modela, eden z interakcijo:

$$Y_1 = 1 + X_2 + Z_1 + 2X_2Z_1 + \epsilon,$$

drugi pa brez:

$$Y_2 = 1 + X_2 + 5Z_2 + \epsilon.$$

Obema tako generiranima izidoma prilagodimo model z interakcijo.

```
y1 <- 1 + x2 + z1 + 2*x2*z1 + epsilon
y2 <- 1 + x2 + 5*z2 + epsilon
fit1 <- ols(y1 ~ x2*z1, data=podatki)
fit2 <- ols(y2 ~ x2*z2, data=podatki)
fit1 #statistično značilna interakcija - pričakovano
```

```
## Linear Regression Model
```

```
##
```

```
## ols(formula = y1 ~ x2 * z1, data = podatki)
```

```
##
```

```
## Model Likelihood Discrimination
```

```
## Ratio Test
```

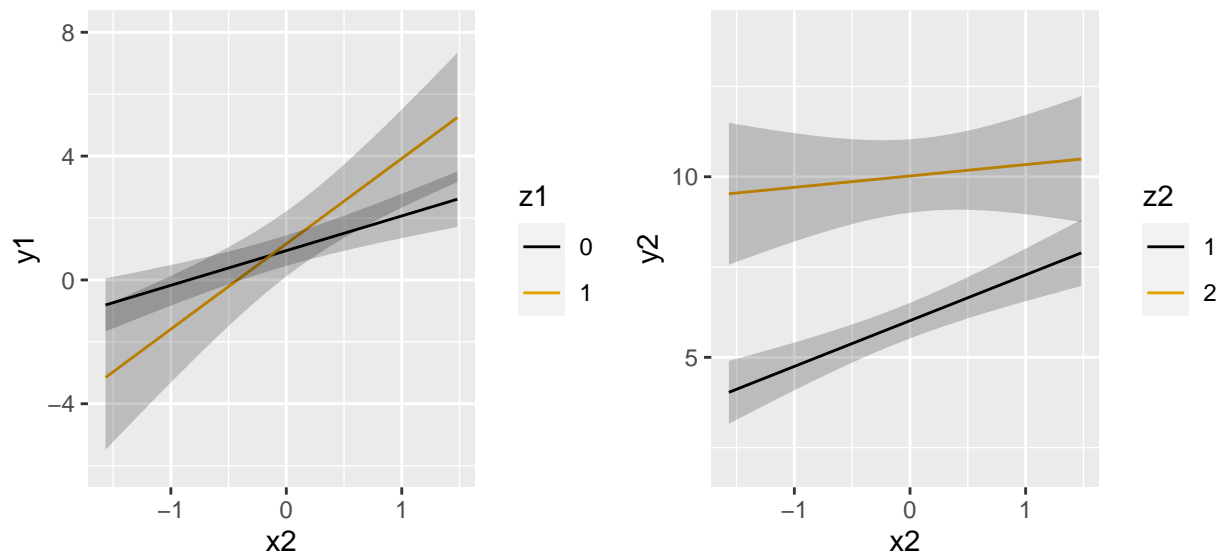
```
Indexes
```

```
## Obs      200    LR chi2      37.48    R2      0.171
## sigma3.1958    d.f.          3    R2 adj  0.158
## d.f.      196    Pr(> chi2) 0.0000    g      1.560
##
## Residuals
##
##      Min      1Q    Median      3Q      Max
## -14.8453 -2.3049  0.1142   2.0861  9.5987
##
##
##      Coef    S.E.    t    Pr(>|t|)
## Intercept  0.9465  0.2505  3.78 0.0002
## x2         1.1208  0.2417  4.64 <0.0001
## z1=1       0.2188  0.5902  0.37 0.7112
## x2 * z1=1  1.6349  0.6907  2.37 0.0189
##
```

```
fit2 #NI statistično značilna interakcija - pričakovano
```

```
## Linear Regression Model
##
## ols(formula = y2 ~ x2 * z2, data = podatki)
##
##      Model Likelihood    Discrimination
##      Ratio Test          Indexes
## Obs      200    LR chi2      68.02    R2      0.288
## sigma3.1642    d.f.          3    R2 adj  0.277
## d.f.      196    Pr(> chi2) 0.0000    g      2.242
##
## Residuals
##
##      Min      1Q    Median      3Q      Max
## -12.98191 -2.13039  0.06232   2.03151  9.32681
##
##
##      Coef    S.E.    t    Pr(>|t|)
## Intercept  6.0152  0.2498 24.08 <0.0001
## x2         1.2669  0.2491  5.09 <0.0001
## z2=2       4.0066  0.5727  7.00 <0.0001
## x2 * z2=2 -0.9518  0.5732 -1.66 0.0985
##
```

```
predict1 <- rms::Predict(fit1, x2, z1)
predict2 <- rms::Predict(fit2, x2, z2)
grid.arrange(ggplot(predict1), ggplot(predict2), ncol = 2)
```



Nauk: Risanje intervalov zaupanja pri grafičnih predstavitvah interakcij v splošnem NI priporočljivo.

3 O logistični regresiji

- Dihotomen/binaren izid Y z vrednostima 1 (dogodek) in 0 (ne dogodek).
- Verjetnost dogodka: $p = P(Y = 1)$
- Obeti (ang. *odds*) za dogodek: $p/(1 - p)$
- Model logistične regresije ($\log = \ln$):

$$\log \frac{p}{1 - p} = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

Ko ocenimo model, dobimo koeficient b_1 . Želimo ga interpretirati!

Imamo dva pacienta, drugi ima spremenljivko X_1 za 1 enoto večjo kakor prvi, oba imata X_2 enak.

b_1 je RAZLIKA med logaritmoma obetov za dogodek pri teh dveh pacientih:

$$b_1 = \log odds_2 - \log odds_1$$

Ni dobra vsebinska intpretacija.

$\exp(b_1)$ je RAZMERJE (kvocient) obetov za dogodek pri teh dveh pacientih:

$$\exp(b_1) = odds_2 / odds_1 =: OR \text{ odds ratio / razmerje obetov}$$

Poročamo v tabelah za logistično regresijo, skupaj z intervalom zaupanja.

Napovedne vrednosti:

- Ob nekih vrednostih spremenljivk torej preko linearnega prediktorja $b_0 + b_1 x_1 + b_2 x_2$ izračunam napoved ta logaritem obetov.
- Če na tem uporabim \exp , dobim napoved za obete $p/(1 - p)$.
- Če pa na linearnemu prediktorju uporabim t. i. logistično funkcijo (imenovano tudi *sigmoid function*) $x \mapsto 1/(1 + \exp(-x))$, dobim napoved za verjetnost dogodka p .

4 Analizirajmo podatke iz motivacije

```
source("podatki.R")
```

Cilji so bili: Prilagoditi model vsem 9 spremenljivkam, kjer moramo preveriti:

- Ali moramo številske spremenljivke vključiti nelinearno? Katere funkcije uporabiti?
- Ali je potrebno vključiti še interakcije med spremenljivkami?
- Ali je model dober?

4.1 Vključevanje nelinearnih efektov

Koliko si lahko privoščimo? Pri logistični regresiji ni odvisno od velikosti vzorca (tako kot pri linearni) temveč od števila dogodkov!

```
table(podatki$nesamostojnost)
```

```
##  
##      0      1  
## 1138  362
```

Število stopinj prostosti (*degrees of freedom*, df) je lahko največ 10% od števila dogodkov, torej smo omejeni s 36 - bolje je imeti precej manj.

Preštajemo stopinje prostosti:

- 9 spremenljivk = 9 df
- 2 imata po 4 kategorije, vsaka torej dodatni 2 df, skupaj dodatne 4 df
- Brez vključevanja nelinearnih efektov za številske spremenljivke imamo 13 df.

Kakšne funkcije bi uporabili za modeliranje nelinearnega efekta?

Priporoča se zlepke (ang. *spline*) - med vozli (ang. *knots*) imamo polinom neke predpisane stopnje, ti polinomi so “zlepljeni” skupaj tako, da so prehodi gladki.

Natančneje, priporoča se *restricted cubic splines* (rcs):

- med vozli je polinom 3. stopnje (*cubic*)
- pred prvim vozlom in za zadnjim vozlom je linearna funkcija (*restricted*)

Če v modelu uporabimo številsko spremenljivko z *restricted cubic splinom* s k vozli, potem je njen df enak $k - 1$.

Prednosti rcs:

- porabijo malo df,
- zelo fleksibilni - predvsem s 4 ali 5 vozli, “kaj več je že pretiravanje”.

Imamo 4 številske spremenljivke. Če vse vključim preko

- rcs s 3 vozli: dodatne 4 df, skupaj 17 df ali 21.3 dogodkov na ocenjeni koeficient (*events per variable*, EV).
- rcs s 4 vozli: dodatne 8 df, skupaj 21 df ali 17.2 EV.
- rcs s 5 vozli: dodatne 12 df, skupaj 25 df ali 14.5 EV.

Zagotovo bi se odločili za rcs s 4 vozli, morda bi lahko imeli tudi 5 vozlov. Poskusimo.

Uporabili bomo **logistično regresijo**:

- Osnovna knjižnica **stats**: funkcija **glm**
- Knjižnica **rms**: funkcija **lrm**

Z osnovno knjižnico le prilagodimo model:

```
fitGlm = glm(nesamostojnost ~ Spol +
             rcs(starost,4) +
             itm +
             kronicneBolezni +
             druzinaAPGAR +
             tveganjePadec +
             rcs(Splosna.ocena.trenutnega.zdravja,4) +
             rcs(Splosna.ocena.obcutenja.osamljenosti,4) +
             rcs(Splosna.ocena.jakosti.prisotne.bolecine,4),
             data = podatki, family=binomial(link="logit"))
summary(fitGlm) #koeficienti b_i
round(exp(summary(fitGlm)$coef[,1]),2) #exp(b_i)

CIs = confint(fitGlm) #IZ za koeficiente b_i
```

Waiting for profiling to be done...

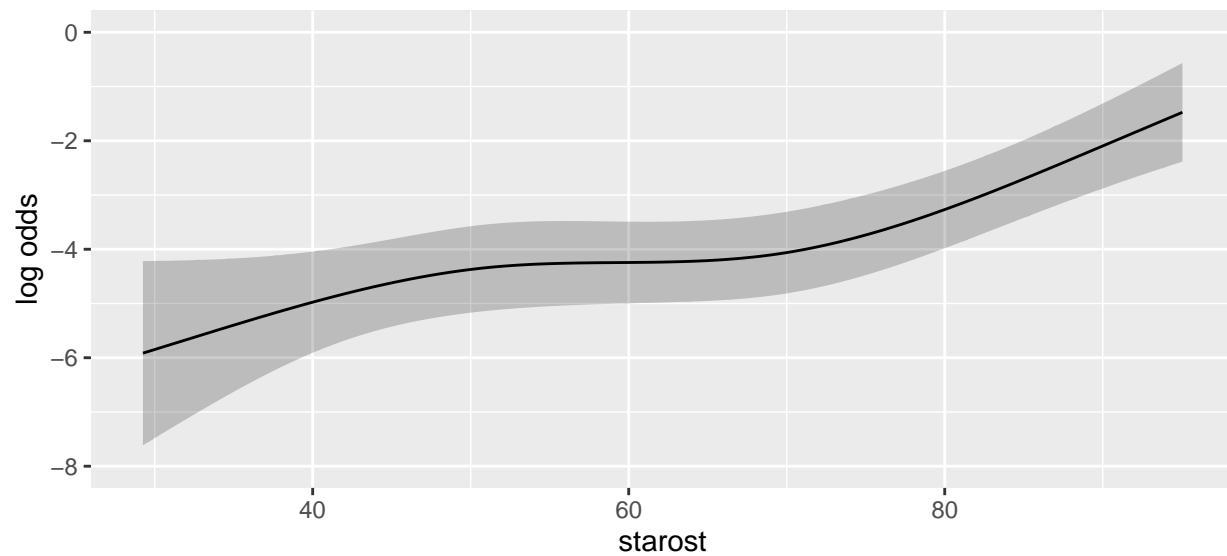
```
round(CIs, 2)
round(exp(CIs), 2) #IZ za exp(b_i)
```

Od sedaj naprej uporabljamo knjižnico **rms**:

```
dd <- datadist(podatki)
options(datadist = "dd")

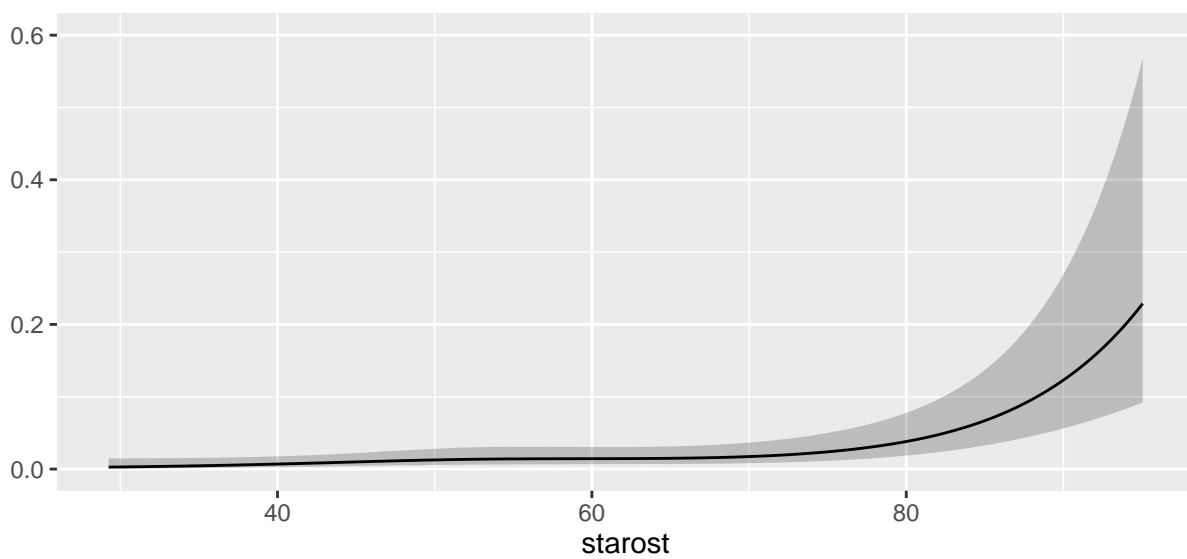
fit4knots = lrm(nesamostojnost ~ Spol +
               rcs(starost,4) +
               itm +
               kronicneBolezni +
               druzinaAPGAR +
               tveganjePadec +
               rcs(Splosna.ocena.trenutnega.zdravja,4) +
               rcs(Splosna.ocena.obcutenja.osamljenosti,4) +
               rcs(Splosna.ocena.jakosti.prisotne.bolecine,4),
               data = podatki)

ggplot(rms::Predict(fit4knots, starost),
       adj.subtitle=F) #odstranim zapis, na kaj so fiksirane ostale spremenljivke
```

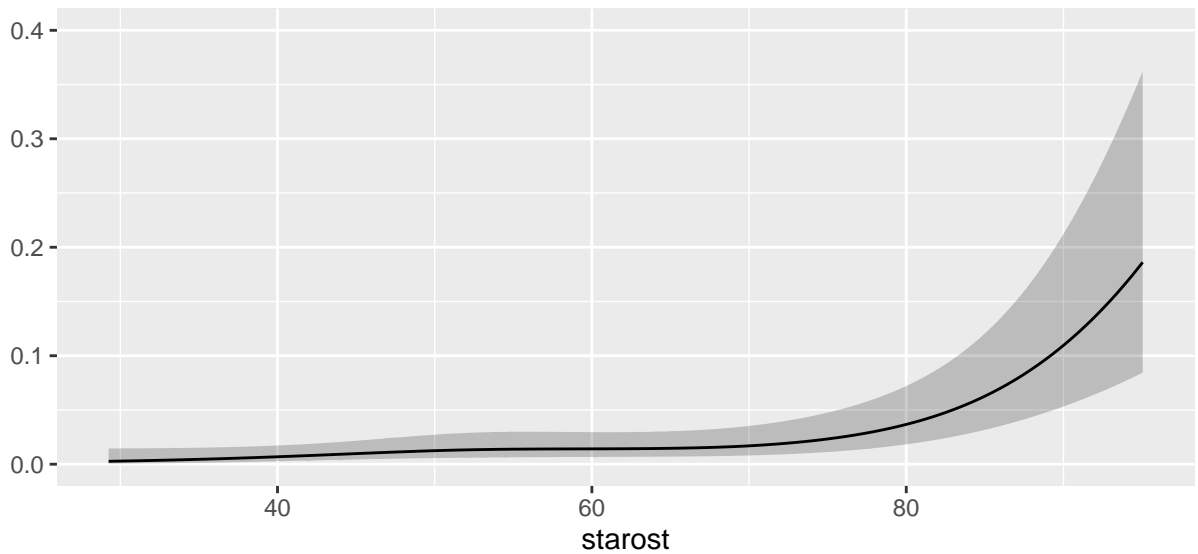


Kaj pa obeti? Ali pa verjetnost za dogodek?

```
ggplot(rms::Predict(fit4knots, starost, fun = exp),
  adj.subtitle=F)
```



```
ggplot(rms::Predict(fit4knots, starost, fun = plogis),
  adj.subtitle=F)
```

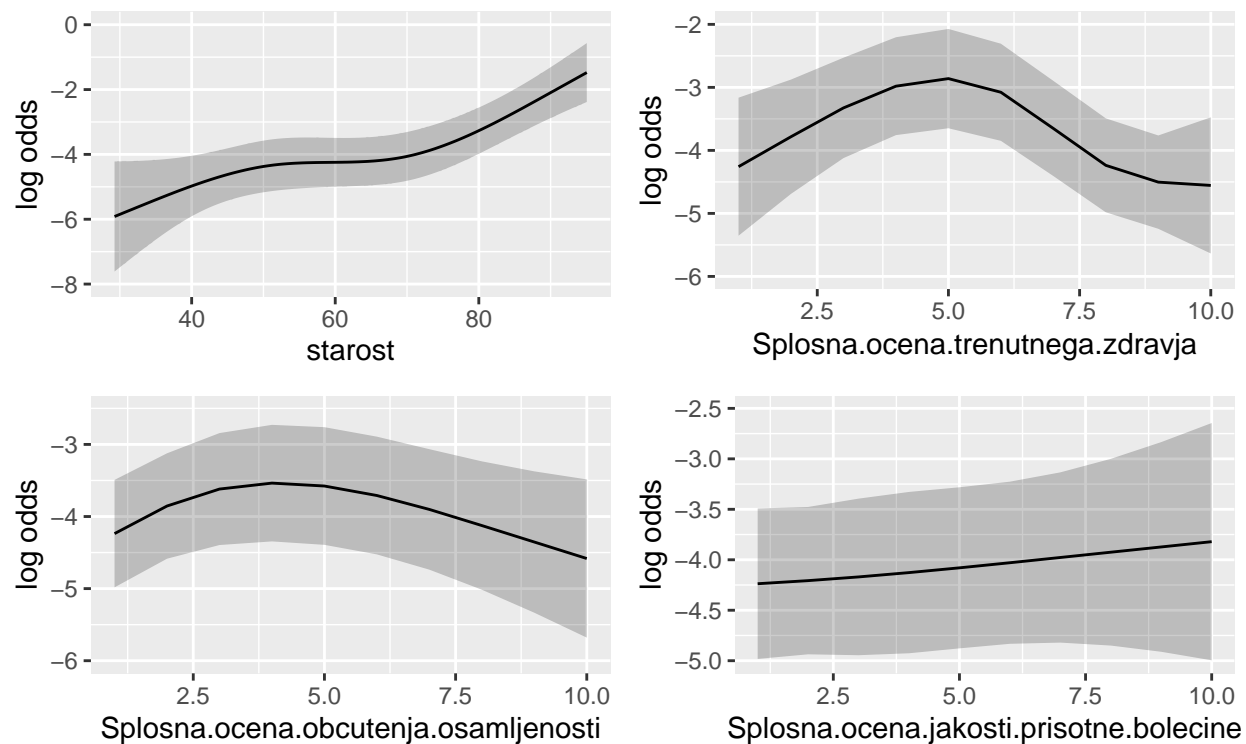


Na tak način lahko dodam katerokoli funkcijo, tudi npr. svojo vnaprej definirano funkcijo f .

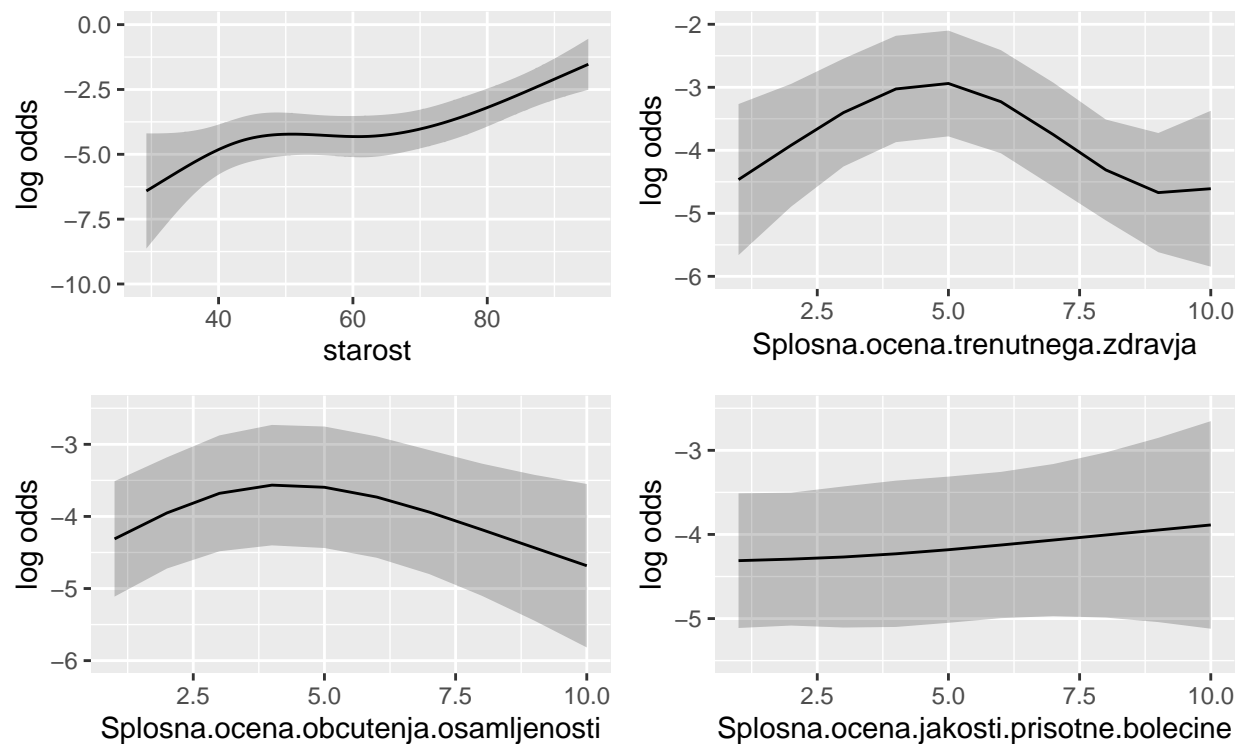
Uporabno, če naredim linearno regresijo na transformiranem številskem izidu (npr. zaradi porazdelitve residualov), potem pa želim prikazati napovedne vrednosti na originalni skali. Namesto Y (številski!) torej modeliram $f(Y)$, nato pa v `Predict` uporabim `fun = g`, kjer kot g definiram inverz funkcije f . **Pozor: ob transformiranem številskem izidu ne moremo interpretirati koeficientov, interpretiramo lahko le njihov predznak in vrednost p .**

Primerjajmo torej slike pri različnem številu vozlov.

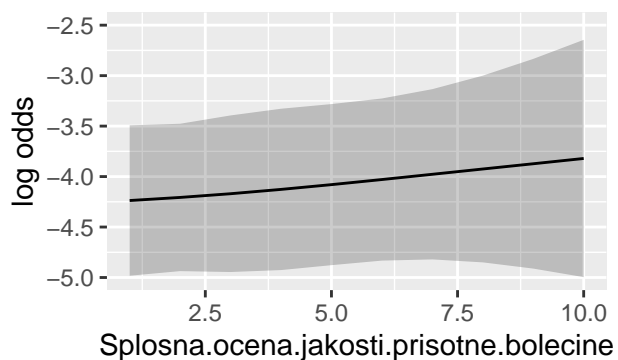
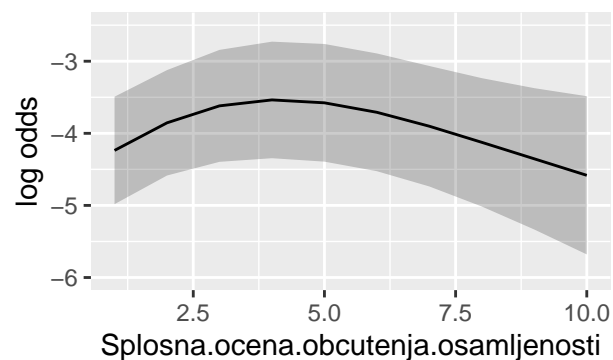
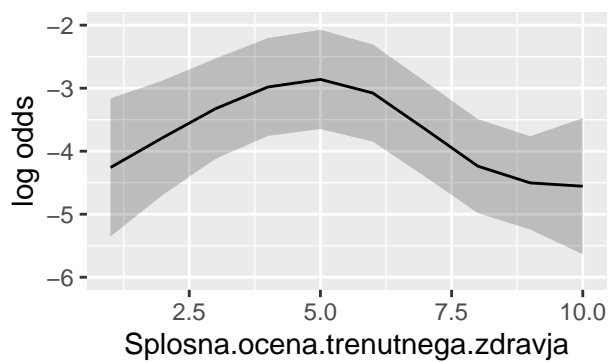
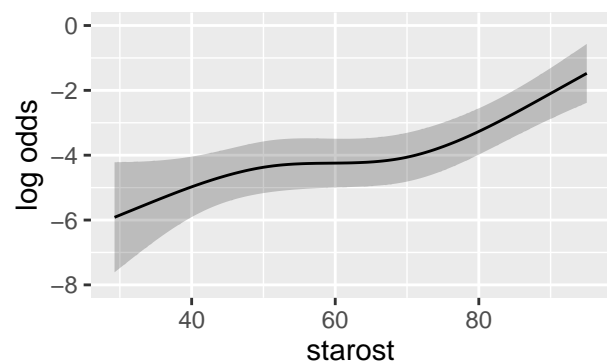
4 vozli



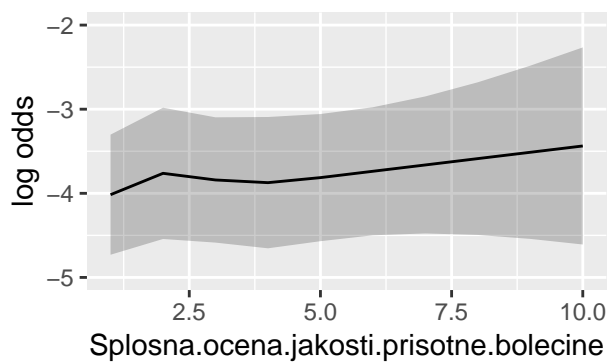
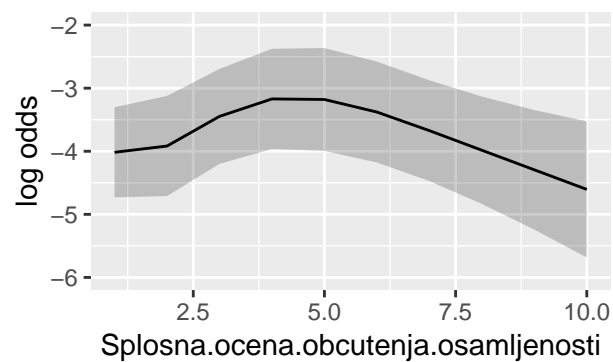
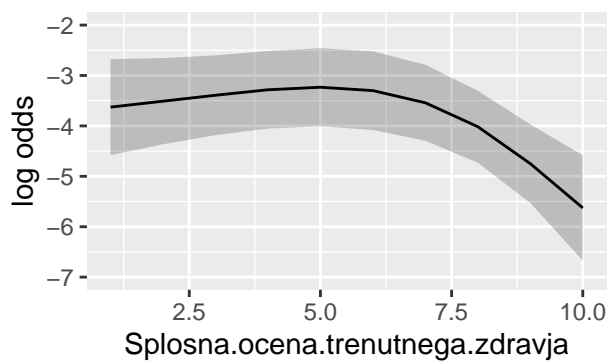
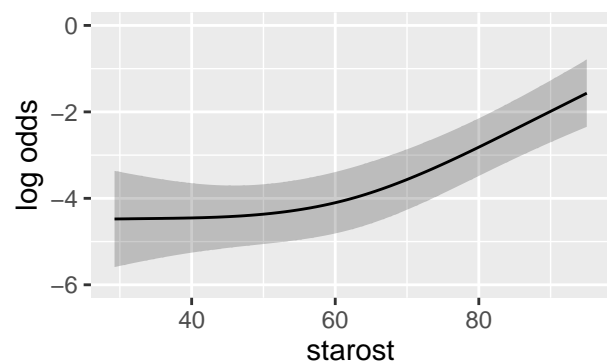
5 vozlov



4 vozli



3 vozli



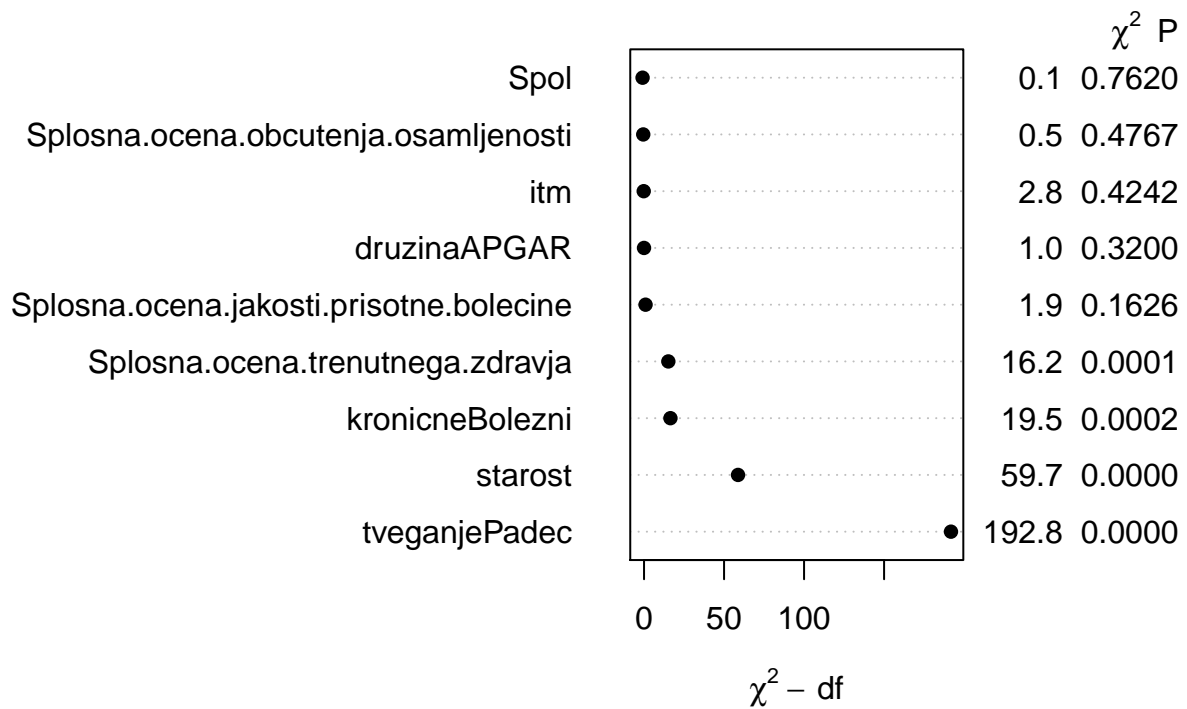
Pogledamo še značilnost nelinearnih delov.

```
anova(fit3knots) #le bolečina nima stat. znač. nelinearnega efekta, tudi cela sprem. ni stat. znač.  
anova(fit4knots) #enako  
anova(fit5knots) #enako
```

Kaj se odločite? Kako boste vsako izmed številskih spremenljivk vključili v model?

Kako “preslepimo samega sebe”:

```
fitLin = lrm(nesamostojnost ~ Spol +
             starost +
             itm +
             kronicneBolezni +
             druzinaAPGAR +
             tveganjePadec +
             Splosna.ocena.trenutnega.zdravja +
             Splosna.ocena.obcutenja.osamljenosti +
             Splosna.ocena.jakosti.prisotne.bolecine,
             data = podatki)
plot(anova(fitLin))
```



4.2 Vključevanje interakcij

```
fitInter = lrm(nesamostojnost ~ Spol +
               rcs(starost,4) +
               itm +
               kronicneBolezni +
               druzinaAPGAR +
               tveganjePadec +
               rcs(Splosna.ocena.trenutnega.zdravja,4) +
               rcs(Splosna.ocena.obcutenja.osamljenosti,4) +
               rcs(Splosna.ocena.jakosti.prisotne.bolecine,4) +
               itm*druzinaAPGAR +
               itm*rcs(Splosna.ocena.jakosti.prisotne.bolecine,4),
               data = podatki)
anova(fitInter)

#izključim IZ v Predict:
ggplot(rms::Predict(fitInter, itm, druzinaAPGAR, Splosna.ocena.jakosti.prisotne.bolecine,
                    conf.int = F)) # NI OK

ggplot(rms::Predict(fitInter, Splosna.ocena.jakosti.prisotne.bolecine, itm, druzinaAPGAR,
                    conf.int = F)) # OK

ggplot(rms::Predict(fitInter, Splosna.ocena.jakosti.prisotne.bolecine, druzinaAPGAR, itm,
                    conf.int = F)) # NAJBOLJE

ggplot(rms::Predict(fitInter, Splosna.ocena.jakosti.prisotne.bolecine, druzinaAPGAR, itm))

table(podatki$itm)
table(podatki$itm, podatki$druzinaAPGAR)

ggplot(rms::Predict(fitInter, Splosna.ocena.jakosti.prisotne.bolecine, druzinaAPGAR, itm,
                    conf.int = F, fun=plogis),
       ylim.=c(0,1)) #ggplot.Predict ima nekaj dodatnih parametrov
```

Kaj bi se odločili?

Kako bi si interakcijo narisali z osnovno knjižnico?

4.3 Ali je model dober?

```
fit4knots = lrm(nesamostojnost ~ Spol +  
               rcs(starost,4) +  
               itm +  
               kronicneBolezni +  
               druzinaAPGAR +  
               tveganjePadec +  
               rcs(Splosna.ocena.trenutnega.zdravja,4) +  
               rcs(Splosna.ocena.obcutenja.osamljenosti,4) +  
               rcs(Splosna.ocena.jakosti.prisotne.bolecine,4),  
               data = podatki, x=TRUE, y=TRUE)
```

```
val <- validate(fit4knots)  
val
```

| | index.orig | training | test | optimism | index.corrected | n |
|--------------|------------|----------|---------|----------|-----------------|----|
| ## Dxy | 0.9081 | 0.9109 | 0.9026 | 0.0084 | 0.8997 | 40 |
| ## R2 | 0.7065 | 0.7142 | 0.6966 | 0.0176 | 0.6889 | 40 |
| ## Intercept | 0.0000 | 0.0000 | -0.0062 | 0.0062 | -0.0062 | 40 |
| ## Slope | 1.0000 | 1.0000 | 0.9487 | 0.0513 | 0.9487 | 40 |
| ## Emax | 0.0000 | 0.0000 | 0.0127 | 0.0127 | 0.0127 | 40 |
| ## D | 0.6390 | 0.6482 | 0.6266 | 0.0216 | 0.6174 | 40 |
| ## U | -0.0013 | -0.0013 | 0.0008 | -0.0022 | 0.0008 | 40 |
| ## Q | 0.6404 | 0.6495 | 0.6257 | 0.0238 | 0.6166 | 40 |
| ## B | 0.0697 | 0.0677 | 0.0715 | -0.0037 | 0.0734 | 40 |
| ## g | 3.0925 | 3.2954 | 3.1222 | 0.1732 | 2.9194 | 40 |
| ## gp | 0.3304 | 0.3312 | 0.3285 | 0.0027 | 0.3276 | 40 |

```
(1+val[1,])/2 #AUC (C index)
```

| | index.orig | training | test | optimism | index.corrected |
|----|------------|-----------|-----------|-----------|-----------------|
| ## | 0.9540291 | 0.9554743 | 0.9512901 | 0.5041842 | 0.9498449 |
| ## | n | | | | |
| ## | 20.5000000 | | | | |

```
val[2,] #pseudo R2 (Nagelkerke)
```

| | index.orig | training | test | optimism | index.corrected |
|----|-------------|------------|------------|------------|-----------------|
| ## | 0.70649250 | 0.71423203 | 0.69659759 | 0.01763444 | 0.68885806 |
| ## | n | | | | |
| ## | 40.00000000 | | | | |

In tako naprej...