Vaja 4: Polinomska regresija in zlepki

R paketi, ki jih bomo uporabili na vajah:

```
library(reshape2) # reshape data sets for ggplot (melt)
library(ggplot2) # nice plots (ggplot)
library(knitr)  # for markdown
library(ISLR) # datasets
library(splines) # spline basis functions
library(effects) # graphical effect displays
library(Hmisc) # data analysis, manipulation, and visualization
```

V današnjih vajah bomo obravnavali nekatere pristope, ki modelirajo nelinearnost. V splošnem ti pristopi temeljijo na tem, da spremenljivki X dodamo dodatne spremenljivke, ki so transformacije X. Ko so t.i. bazne funkcije določene, lahko ustrezno transformirano modelsko matriko modeliramo z linearnim modelom, saj je model v teh novo določenih spremenljivkah linearen. Za lažje razumevanje nekaterih od teh pristopov si lahko pomagamo z aplikacijo: https://clinicalbiometrics.shinyapps.io/Bendyourspline/

V podatkovnem okviru `Wage` v paketu `ISLR` so podatki o plačah 3000 moških delavcev v srednje atlantski regiji.

```
data("Wage")
str(Wage)
```

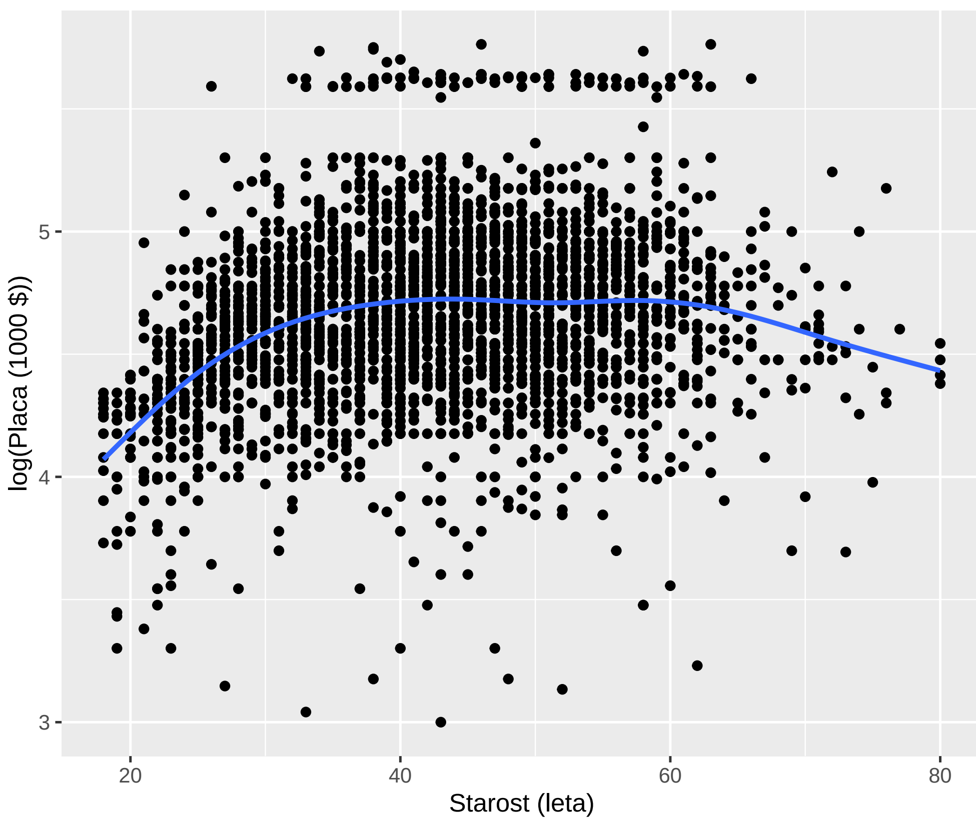
```
'data.frame':   3000 obs. of  11 variables:
 $ year      : int  2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 ...
 $ age       : int  18 24 45 43 50 54 44 30 41 52 ...
 $ maritl    : Factor w/ 5 levels "1. Never Married",..: 1 1 2 2 4 2 2 1 1 2 ...
 $ race      : Factor w/ 4 levels "1. White","2. Black",..: 1 1 1 3 1 1 4 3 2 1 ...
 $ education : Factor w/ 5 levels "1. < HS Grad",..: 1 4 3 4 2 4 3 3 3 2 ...
 $ region    : Factor w/ 9 levels "1. New England",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ jobclass  : Factor w/ 2 levels "1. Industrial",..: 1 2 1 2 2 2 1 2 2 2 ...
 $ health    : Factor w/ 2 levels "1. <=Good","2. >=Very Good": 1 2 1 2 1 2 2 1 2 2 ...
 $ health_ins: Factor w/ 2 levels "1. Yes","2. No": 2 2 1 1 1 1 1 1 1 1 ...
 $ logwage   : num  4.32 4.26 4.88 5.04 4.32 ...
 $ wage      : num  75 70.5 131 154.7 75 ...
```

```
#summary table
summary(Wage)
```

```
      year           age                     maritl           race     
 Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480  
 1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293  
 Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190  
 Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37  
 3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55                  
 Max.   :2009   Max.   :80.00                                          

              education                     region               jobclass   
 1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544  
 2. HS Grad        :971   1. New England       :   0   2. Information:1456  
 3. Some College   :650   3. East North Central:   0                        
 4. College Grad   :685   4. West North Central:   0                        
 5. Advanced Degree:426   5. South Atlantic    :   0                        
                          6. East South Central:   0                        
                          (Other)              :   0                        
            health      health_ins      logwage           wage       
```

```
 1. <=Good      : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09
 2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38
                                     Median :4.653   Median :104.92
                                     Mean   :4.654   Mean   :111.70
                                     3rd Qu.:4.857   3rd Qu.:128.68
                                     Max.   :5.763   Max.   :318.34
```

```r
ggplot(data=Wage, aes(x=age, y=log(wage))) +
  geom_point() +
  geom_smooth(se=FALSE) +
  #geom_smooth(method="lm", se=FALSE) +
  xlab("Starost (leta)") +
  ylab("log(Plača (1000 $))")
```



Slika 1: Odvisnost `logwage` od `age` v podatkovnem okviru *Wage*.

1. Polinomska regresija

V prejšnji vaji smo videli, da je zveza med logaritmom plače (`logwage`) in starostjo (`age`) nelinearna. Nelinearnost bomo najprej modelirali s polinomsko regresijo. Naš običajni linearni model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

bomo nadomestili z modelom:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_p x_i^p + \epsilon_i,$$

pri čemer je d stopnja polinoma. Ocene parametrov v modelu lahko preprosto ocenimo z metodo najmanjših kvadratov, saj gre za standardni linearni model z napovednimi spremenljivkami $x_i, x_{2i}, x_{3i}, \ldots, x_{pi}$. V praksi redko nastavimo p, ki je večji od 3 ali 4, saj se pri velikih stopnjah polinoma model hitro preprilega, še posebej na robovih spremenljivke X.

Komentar: Z modeliranjem nelinearnosti bomo na račun interpretabilnosti modela izboljšali napovedno kakovost modela (ocene parametrov ne bodo imele vsebinskega pomena, dobili pa bomo bolj natančne napovedi). V kolikor bi nas v praksi zanimale le napovedi za povprečno plačo na podlagi starosti, bi lahko v model vključili kar spremenljivko `wage` na originalni skali, četudi imamo prisotno heteroskedastičnost. Razmislite, zakaj.

```
model.stopnja1 <- lm(logwage ~ age, data = Wage)
model.stopnja2 <- lm(logwage ~ poly(age, 2), data = Wage)
model.stopnja3 <- lm(logwage ~ poly(age, 3), data = Wage)
model.stopnja4 <- lm(logwage ~ poly(age, 4), data = Wage)
model.stopnja5 <- lm(logwage ~ poly(age, 5), data = Wage)
model.stopnja6 <- lm(logwage ~ poly(age, 6), data = Wage)
```

Kako se spreminjajo stopinje prostosti modela z večanjem stopnje polinoma? Kakšno je število ocenjenih parametrov? V kakšnem razmerju je število ocenjenih parametrov in število stopinj prostosti modela?

Katera stopnja polinoma je ustrezna? Utemeljite odgovor.

```
anova(model.stopnja1, model.stopnja2,
      model.stopnja3, model.stopnja4,
      model.stopnja5, model.stopnja6)
```

```
Analysis of Variance Table

Model 1: logwage ~ age
Model 2: logwage ~ poly(age, 2)
Model 3: logwage ~ poly(age, 3)
Model 4: logwage ~ poly(age, 4)
Model 5: logwage ~ poly(age, 5)
Model 6: logwage ~ poly(age, 6)
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1   2998 353.45
2   2997 331.41  1   22.0377 201.6446 < 2.2e-16 ***
3   2996 328.71  1    2.7057  24.7572  6.87e-07 ***
4   2995 327.31  1    1.3913  12.7300 0.0003655 ***
5   2994 327.30  1    0.0177   0.1615 0.6877762
6   2993 327.10  1    0.1923   1.7591 0.1848370
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Linearni model, v katerem imamo le spremenljivko `age`, ocenjuje dva parametra: presečišče in naklon. Stopinje prostosti modela `model.stopnja1` so torej enake $n - 2 = 2998$. Z vsako stopnjo polinoma se oceni dodatni parameter, torej se porabljajo stopinje prostosti.

Pri izbiri stopnje polinoma bi si lahko pomagali tudi z drugimi metodami (npr. z navzkrižnim preverjanjem, AIC, prilagojeni R^2 ...). Več o tem v poglavju *Izbira modela*.

Ker so ortogonalni polinomi med seboj neodvisni, enake rezultate dobimo tudi na podlagi t-testov v povzetku modela:

```
summary(model.stopnja5)
```

```
Call:
```

```
lm(formula = logwage ~ poly(age, 5), data = Wage)

Residuals:
     Min       1Q   Median       3Q      Max
-1.71864 -0.19330  0.00813  0.18445  1.11760

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.653905   0.006036 770.962  < 2e-16 ***
poly(age, 5)1  4.197217   0.330632  12.695  < 2e-16 ***
poly(age, 5)2 -4.694434   0.330632 -14.198  < 2e-16 ***
poly(age, 5)3  1.644906   0.330632   4.975 6.89e-07 ***
poly(age, 5)4 -1.179518   0.330632  -3.567 0.000366 ***
poly(age, 5)5  0.132869   0.330632   0.402 0.687814
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3306 on 2994 degrees of freedom
Multiple R-squared:  0.118, Adjusted R-squared:  0.1165
F-statistic: 80.08 on 5 and 2994 DF,  p-value: < 2.2e-16
```

Na podlagi F-testa za primarjavo gnezdenih modelov (oz. t-testa) izberemo polinom četrte stopnje.

Poglejmo si, kako izgleda modelska matrika izbranega modela.

```r
head(model.matrix(model.stopnja4))
```

```
       (Intercept) poly(age, 4)1 poly(age, 4)2 poly(age, 4)3 poly(age, 4)4
231655           1 -0.0386247992   0.055908727 -0.0717405794   0.086729854
86582            1 -0.0291326034   0.026298066 -0.0145499511  -0.002599280
161300           1  0.0040900817  -0.014506548 -0.0001331835   0.014480093
155159           1  0.0009260164  -0.014831404  0.0045136682   0.012657507
11443            1  0.0120002448  -0.009815846 -0.0111366263   0.010211456
376662           1  0.0183283753  -0.002073906 -0.0166282799  -0.001314381
```

Poglejmo še ocene koeficientov v izbranem modelu. Kolikšen delež varibilnosti odzivne spremenljivke pojasni model?

```r
summary(model.stopnja4)
```

```
Call:
lm(formula = logwage ~ poly(age, 4), data = Wage)

Residuals:
     Min       1Q   Median       3Q      Max
-1.71991 -0.19342  0.00612  0.18386  1.12066

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.653905   0.006036 771.070  < 2e-16 ***
poly(age, 4)1  4.197217   0.330586  12.696  < 2e-16 ***
poly(age, 4)2 -4.694434   0.330586 -14.200  < 2e-16 ***
poly(age, 4)3  1.644906   0.330586   4.976 6.87e-07 ***
poly(age, 4)4 -1.179518   0.330586  -3.568 0.000365 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.3306 on 2995 degrees of freedom
Multiple R-squared:  0.1179,    Adjusted R-squared:  0.1167
F-statistic: 100.1 on 4 and 2995 DF,  p-value: < 2.2e-16
```

Kaj se zgodi, če v modelu nastavimo argument `raw = TRUE`?

```
model.stopnja4.raw <- lm(logwage ~ poly(age, 4, raw = TRUE), data = Wage)

head(model.matrix(model.stopnja4.raw))
```

```
       (Intercept) poly(age, 4, raw = TRUE)1 poly(age, 4, raw = TRUE)2
231655           1                        18                       324
86582            1                        24                       576
161300           1                        45                      2025
155159           1                        43                      1849
11443            1                        50                      2500
376662           1                        54                      2916
       poly(age, 4, raw = TRUE)3 poly(age, 4, raw = TRUE)4
231655                      5832                    104976
86582                      13824                    331776
161300                     91125                   4100625
155159                     79507                   3418801
11443                     125000                   6250000
376662                    157464                   8503056
```

Če funkcijo `poly()` uporabimo brez privzete nastavitve argumenta `raw=FALSE`, štiri bazne funkcije ne predstavljajo ortogonalnih polinomov, temveč je vsaka bazna funkcija linearna kombinacija spremenljivk *age*, age^2, age^3 in age^4. Te funkcije med seboj niso neodvisne in rezultati testiranja ničelnih domnev o parametrih v povzetku linearnega modela niso enaki, kot pri F-testu za gnezdene modele. Vrednosti ocen parametrov so drugačne, v obeh primerih pa dobimo enake napovedane vrednosti in enak koeficient determinacije.

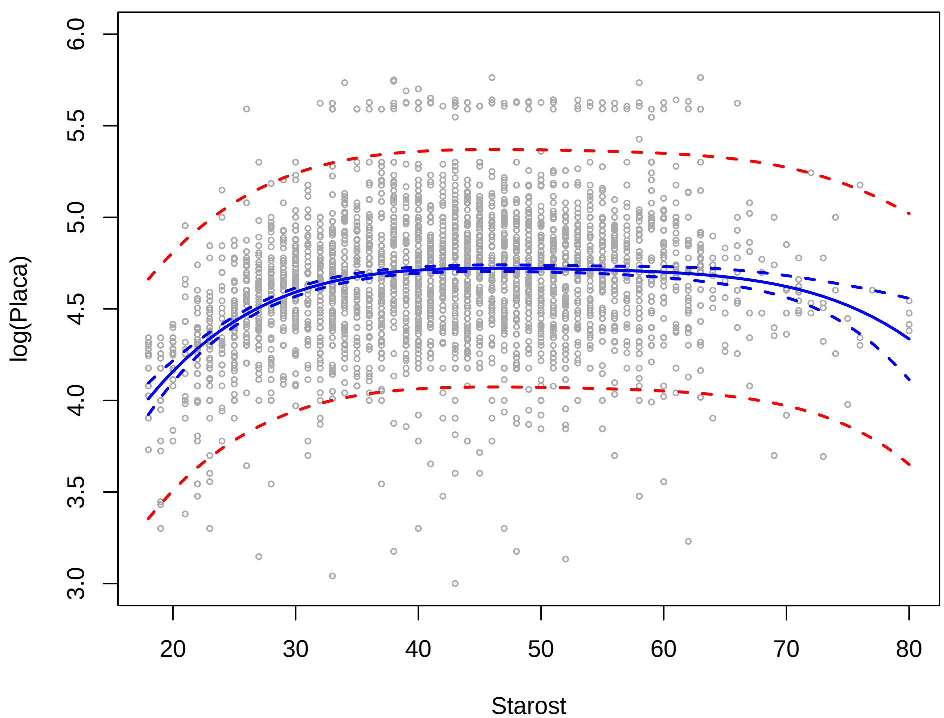
```
summary(model.stopnja4.raw)
```

```

Call:
lm(formula = logwage ~ poly(age, 4, raw = TRUE), data = Wage)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.71991 -0.19342  0.00612  0.18386  1.12066 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                9.698e-01  4.973e-01   1.950 0.051237 .  
poly(age, 4, raw = TRUE)1  2.832e-01  4.876e-02   5.808 6.97e-09 ***
poly(age, 4, raw = TRUE)2 -8.020e-03  1.707e-03  -4.698 2.74e-06 ***
poly(age, 4, raw = TRUE)3  1.014e-04  2.539e-05   3.992 6.70e-05 ***
poly(age, 4, raw = TRUE)4 -4.850e-07  1.359e-07  -3.568 0.000365 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3306 on 2995 degrees of freedom
Multiple R-squared:  0.1179,    Adjusted R-squared:  0.1167
F-statistic: 100.1 on 4 and 2995 DF,  p-value: < 2.2e-16
```

Grafično prikažimo napovedi modela polinoma stopnje 4 s 95% intervali zaupanja za povprečno in posamično vrednost s funkcijo predict().

```
age.nap <- seq(from = min(Wage$age), to = max(Wage$age))

napovedi.povp <- predict(model.stopnja4, newdata = data.frame(age = age.nap),
                         interval = "confidence")
napovedi.pos <- predict(model.stopnja4, newdata = data.frame(age=age.nap),
                           interval="prediction")
```

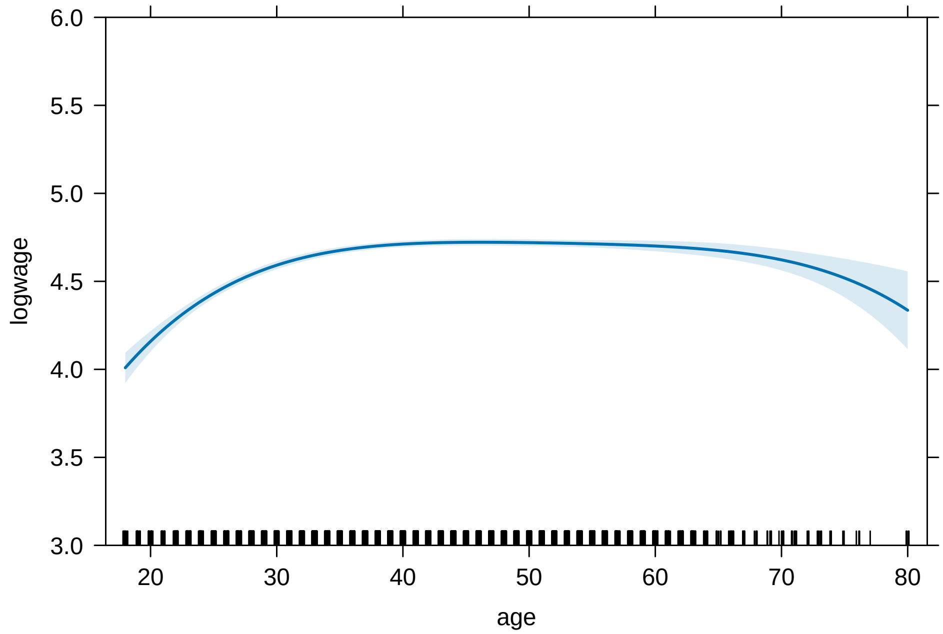
```
plot(Wage$age, Wage$logwage, xlim = range(Wage$age), cex = 0.5, col ="darkgrey",
     xlab="Starost", ylab="log(Plača)", ylim=c(3, 6))
lines(age.nap, napovedi.povp[,"fit"], lwd = 2, col=" blue")
matlines (age.nap, napovedi.povp[, c("lwr","upr")], lwd = 2, col = " blue", lty = 2)
matlines (age.nap, napovedi.pos[, c("lwr","upr")], lwd = 2, col = " red", lty = 2)
```



Slika 2: Napovedi logaritma plače (logwage) na podlagi modela model.stopnja4 s 95 % intervali zaupanja za povprečno (modra) oz. posamično napoved (rdeča).

Za prikaz povprečnih napovedi si lahko pomagamo tudi z ukazi iz paketa effects.

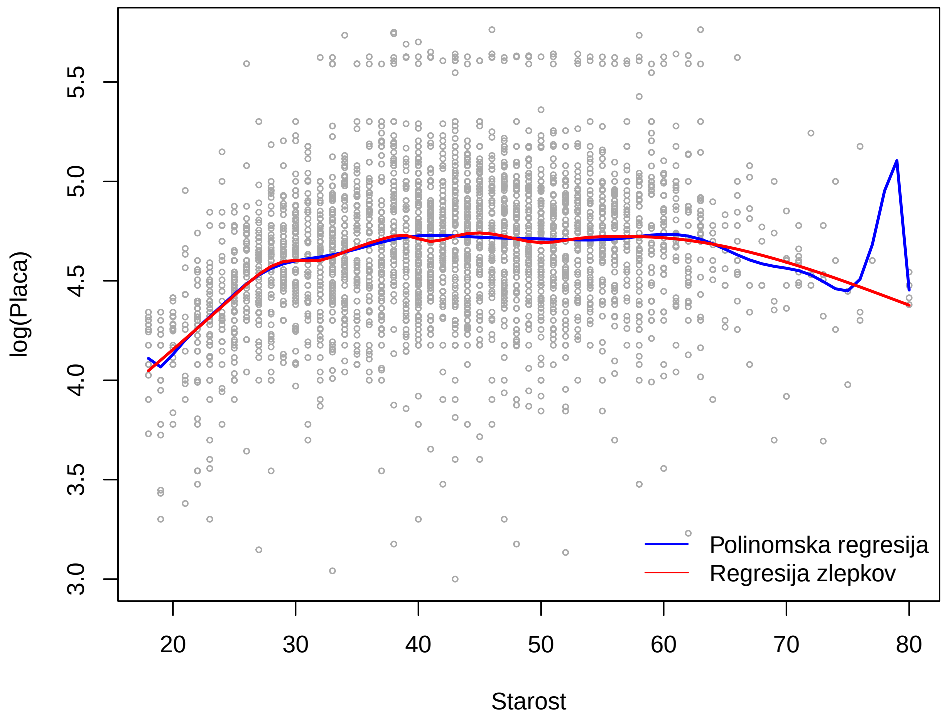
```
plot(predictorEffects(model.stopnja4), main = "", ylim=c(3, 6))
```



Slika 3: Povprečne napovedi logaritma plače (`logwage`) na podlagi modela `model.stopnja4` s 95 % intervali zaupanja.

V praksi se polinomska regresija uporablja le za nižje stopnje polinomov (nekje do 4). Primerjajmo polinomsko regresijo 15. stopnje z naravnim zlepkom s 15 stopinjami prostosti.

```
plot(Wage$age, Wage$logwage, xlim = range(Wage$age), cex = 0.5, col ="darkgrey",
     xlab="Starost", ylab="log(Plača)")
lines(age.nap, predict(lm(logwage ~ poly(age, 15), data = Wage),
                        newdata = data.frame(age=age.nap)), lwd = 2, col="blue")
lines(age.nap, predict(lm(logwage ~ ns(age, df=15), data = Wage),
                        newdata = data.frame(age=age.nap)), lwd = 2, col="red")
legend("bottomright", c("Polinomska regresija", "Regresija zlepkov"),
       lty=c(1,1), col=c("blue", "red"), bty="n")
```



Slika 4: Povprečne napovedi logaritma plače (`logwage`) na podlagi polinomske regresije 15. stopnje ter regresije naravnih zlepkov s 15 stopinjami prostosti.

Prevelika stopnja polinoma privede do nezaželenega obnašanja v repih medtem, ko je pri regresiji zlepkov zveza med `logwage` in `age` še smiselna.

2. Kubični zlepki

Poglejmo torej še, kako bi nelinearno zvezo med `logwage` in `age` modelirali s kubičnimi zlepki. Kubični zlepek je funkcija, ki je zvezna in ima na vozliščih zvezne prve in druge odvode. Da se pokazati, da bazne funkcije (to so t.i. *truncated power basis*):

$$h_1(X) = 1, \qquad h_2(X) = X, \qquad h_3(X) = X^2,$$

$$h_4(X) = X^3, \qquad h_5(X) = (X - a_1)^3_+, \qquad h_6(X) = (X - a_2)^3_+,$$

pri čemer je $h_m(X)$ m-ta transformacija X, predstavljajo kubični zlepek z vozliščema pri a_1 in a_2. Kubični zlepki so izpeljani iz kubične polinomske regresije po odsekih, pri čemer je število parametrov za zgornji primer enako: (3 regije)×(4 parametri za vsako regijo)-(2 vozlišči)×(3 omejitve za vsako vozlišče)= 6.

Tako kot smo videli pri polinomski regresiji (ortogonalni/neortogonalni zlepki), obstajajo tudi drugačne bazne funkcije, ki tvorijo kubične zlepke, ki se v praksi pogosteje uporabljajo zaradi večje numerične stabilnosti.

Na splošno je torej število porabljenih stopinj prostosti pri kubičnih zlepkih enako $3 + K$ (+ presečišče), kjer je K število vozlišč. To je pomembno zato, ker fleksibilnost zlepka v programskih paketih pogosto določamo tako, da z argumentom `df` nastavimo število stopinj prostosti zlepka. Ukaz `bs(x,df=7)` generira matriko baznih funkcij s $7 - 3 = 4$ vozlišči na percentilih x (20., 40., 60. in 80.)

Na napovedi pa ne vpliva le število, pomemben je tudi položaj vozlišč. Zlepki se najbolje prilegajo na intervalih z veliko vozlišči, zato več vozlišč postavimo tja, kjer želimo, da se funkcija hitreje spreminja, in manj vozlišč tja, kjer je bolj stabilna. Velikokrat privzete nastavitve funkcij položaj vozlišč že dovolj smiselno določajo (običajno na podlagi percentilov tako, da nastavimo želeno število stopinj prostosti zlepka). Kdaj pa temu ni tako, zato je priporočljivo položaj vozlišč nastaviti ročno. Taki primeri so:

- Neenakomerna porazdelitev podatkov: privzeta postavitev vozlišč lahko povzroči preprileganje (*overfitting*) na intervalih z malo podatki in podprileganje (*underfitting*) na območjih z več podatki.
- Prisotnost znanih prelomnih točk ali pragov: če so prelomne točke, kjer se odnosi spreminjajo na nekem področju znani (npr. fiziološki pragi v medicini, spremembe politike v ekonomiji), je smiselno vozlišča postaviti na teh mestih.
- Nelinearno razmerje med napovedno in odzivno spremenljivko, ki se razlikuje glede na posamezni interval.
- Preprečevanje pre- oz. podprileganja: preveč vozlišč lahko povzroči preprileganje, zaradi česar zlepek preveč sledi posameznim enotam v podatkih. premalo vozlišč lahko vodi do preveč zglajene zveze.

Naredite model regresije kubičnih zlepkov z vozlišči na 33,75, 42 in 51 let in ga poimenujte `model.bs3`. Modelsko matriko za kubične zlepke lahko zgeneriramo s funkcijo `bs` (B-splines) iz paketa `splines`. B-zlepki so reparametrizirana polinomska regresija po odsekih. Po privzetih nastavitvah funkcija generira modelsko matriko kubičnih zlepkov (argument za stopnjo (degree) `d=3`).

```
model.bs3 <- lm(logwage ~ bs(age, knots = c(33.75, 42.00, 51.00)), data = Wage)
# ocenjujemo K + degree (p=3 za kubične) + 1 (intercept) parametrov
anova(model.bs3)
```

```
Analysis of Variance Table

Response: logwage
                                   Df Sum Sq Mean Sq F value    Pr(>F)
bs(age, knots = c(33.75, 42, 51))    6  43.87  7.3123   66.89 < 2.2e-16 ***
Residuals                         2993 327.19  0.1093
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Kako se spreminjajo stopinje prostosti modela z večanjem števila vozlišč? Kakšno je število ocenjenih parametrov? V kakšnem razmerju je število ocenjenih parametrov in število stopinj prostosti?

3. Naravni zlepki

Vendar pa tudi pri kubičnih zlepkih lahko pride do neželenega obnašanja na robovih funkcije. To omilijo naravni zlepki z dodatnimi omejitvami, da je funkcija izven zunanjih vozlišč (*boundary knots*) linearna. To na vsakem repu sprosti še dve stopinji prostosti, naravni zlepek porabi torej $K - 1$ (+ presečišče) stopinj prostosti, pri čemer je K vsota notranjih ter dveh zunanjih vozlišč.

Primerjajte napovedi modela regresije kubičnih zlepkov `model.bs3` z modelom regresije naravnih zlepkov z enako postavljenimi vozlišči ter z enakim številom porabljenih stopinj prostosti. Po privzetih nastavitvah funkcije v paketu `splines` so notranja vozlišča postavljena na vrednosti kvantilov, ki vrednosti napovedne spremenljivke razdelijo na številčno enake dele, zunanji vozlišči, v repih katerih je funkcija linearna, pa sta pri najmanjši oz. največji vrednosti spremenljivke X. Nekoliko drugačna priporočila da Harrell, glejte dokumetacijo `?rcspline.eval` v paketu `Hmisc`.

```
# enako postavljena vozlišča
model.ns5 <- lm(logwage ~ ns(age, df = 4), data = Wage)
# ocenjunemo K (notranja + 2 zunanji vozlišči) - 1 + 1 (intercept) parametrov
attr(ns(Wage$age, df = 4), "knots") # ukaz izpiše le notranja vozlišča
```

```
[1] 33.75 42.00 51.00
```

```r
attr(ns(Wage$age, df = 4), "Boundary.knots")
```

```
[1] 18 80
```

```r
# zunanji vozlišči v paketu splines sta min(age) in max(age)

# enake df
model.ns7 <- lm(logwage ~ ns(age, df = 6), data = Wage)
# ocenjunemo K (notranja + 2 zunanji vozlišči) - 1 + 1 (intercept) parametrov
attr(ns(Wage$age, df = 6), "knots") # ukaz izpiše le notranja vozlišča
```

```
[1] 30 37 42 48 54
```

```r
attr(ns(Wage$age, df = 6), "Boundary.knots")
```

```
[1] 18 80
```

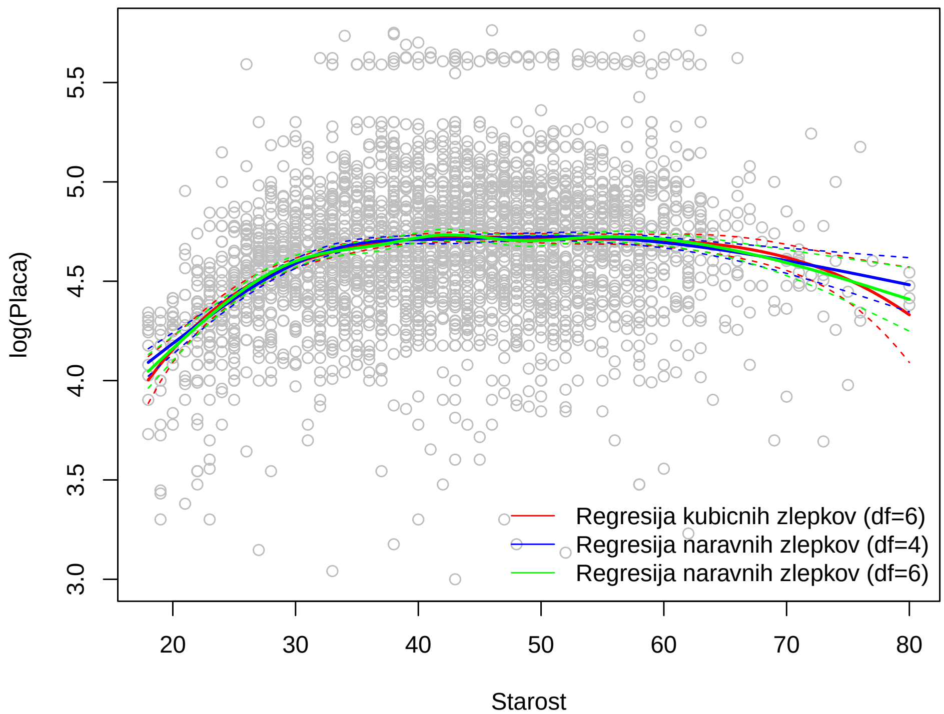
```r
# zunanji vozlišči v paketu splines sta min(age) in max(age)

head(model.matrix(model.ns5))
```

```
       (Intercept) ns(age, df = 4)1 ns(age, df = 4)2 ns(age, df = 4)3
231655           1       0.00000000       0.00000000       0.00000000
86582            1       0.01731602      -0.13795411       0.31872157
161300           1       0.75108560       0.16605047       0.09131609
155159           1       0.78017192       0.07222489       0.11002552
11443            1       0.52933205       0.38879374       0.13708340
376662           1       0.34484721       0.49710028       0.19449432
       ns(age, df = 4)4
231655       0.00000000
86582       -0.18076746
161300      -0.05061290
155159      -0.06235889
11443       -0.05540437
376662      -0.03644180
```

```r
age.nap <- seq(from = min(Wage$age), to = max(Wage$age))

napovedi.bs3 <- predict(model.bs3, newdata = data.frame(age = age.nap),
                        interval = "confidence")

napovedi.ns5 <- predict(model.ns5, newdata = data.frame(age = age.nap),
                        interval = "confidence")

napovedi.ns7 <- predict(model.ns7, newdata = data.frame(age = age.nap),
                        interval = "confidence")

#napovedi.stopnja4 <- predict(model.stopnja4, newdata = data.frame(age = age.nap),
     #                    interval = "confidence")
```

```r
plot(Wage$age, Wage$logwage, col ="gray", xlab="Starost", ylab="log(Plača)")

lines(age.nap, napovedi.bs3[,"fit"], col="red", lwd=2)
matlines(age.nap, napovedi.bs3[, c("lwr","upr")], lwd = 1, col = " red", lty = 2)

lines(age.nap, napovedi.ns5[,"fit"], col="blue", lwd=2)
```

```
matlines(age.nap, napovedi.ns5[, c("lwr","upr")], lwd = 1, col = " blue", lty = 2)

lines(age.nap, napovedi.ns7[,"fit"], col="green", lwd=2)
matlines(age.nap, napovedi.ns7[, c("lwr","upr")], lwd = 1, col = " green", lty = 2)

legend("bottomright",
       c("Regresija kubičnih zlepkov (df=6)",
         "Regresija naravnih zlepkov (df=4)",
         "Regresija naravnih zlepkov (df=6)"),
       lty=1, col=c("red", "blue", "green"), bty="n")
```



Slika 5: Povprečne napovedi logaritma plače (`logwage`) glede na starost (`age`), modelirano z naravnimi zlepki (modra), kubičnimi zlepki (rdeča) ter polinomom 4. stopnje (zelena) s 95 % intervali zaupanja.

Kako se razlikujejo napovedi modelov? Zapišite ugotovitve.

S tem, ko smo pri naravnih zlepkih sprostili štiri stopnje prostosti (zaradi dveh omejitev na vsakem robu funkcije), ki jih lahko bolj smiselno porabimo tako, da raje dodamo vozlišča v notranjosti, smo pri istem številu stopinj prostosti modela uspeli zmanjšati varianco napak, kar se odraža v ožjih intervalih zaupanja za povprečno napoved predvsem na robovih funkcije.

Zdaj naredimo še modele naravnih zlepkov za naslednja števila vozlišč: $K = 3, 4, 5, 6, 7, 8$.

```
model.ns3 <- lm(logwage ~ ns(age, df = 2), data = Wage)
attr(ns(Wage$age, df = 2), "knots") # notranje vozlišče
```

```
[1] 42
```

```r
attr(ns(Wage$age, df = 2), "Boundary.knots") # zunanji vozlišči
```

```
[1] 18 80
```

```r
model.ns4 <- lm(logwage ~ ns(age, df = 3), data = Wage)
attr(ns(Wage$age, df = 3), "knots")
```

```
[1] 37 48
```

```r
model.ns5 <- lm(logwage ~ ns(age, df = 4), data = Wage)
attr(ns(Wage$age, df = 4), "knots")
```

```
[1] 33.75 42.00 51.00
```

```r
model.ns6 <- lm(logwage ~ ns(age, df = 5), data = Wage)
attr(ns(Wage$age, df = 5), "knots")
```

```
[1] 32 39 46 53
```

```r
model.ns7 <- lm(logwage ~ ns(age, df = 6), data = Wage)
attr(ns(Wage$age, df = 6), "knots")
```

```
[1] 30 37 42 48 54
```

```r
model.ns8 <- lm(logwage ~ ns(age, df = 7), data = Wage)
attr(ns(Wage$age, df = 7), "knots")
```

```
[1] 29 35 40 45 49 55
```

```r
model.ns9 <- lm(logwage ~ ns(age, df = 8), data = Wage)
attr(ns(Wage$age, df = 8), "knots")
```

```
[1] 28.000 33.750 38.000 42.000 46.375 51.000 56.000
```

```r
model.ns10 <- lm(logwage ~ ns(age, df = 9), data = Wage)
attr(ns(Wage$age, df = 9), "knots")
```

```
[1] 28 33 37 41 44 48 52 57
```

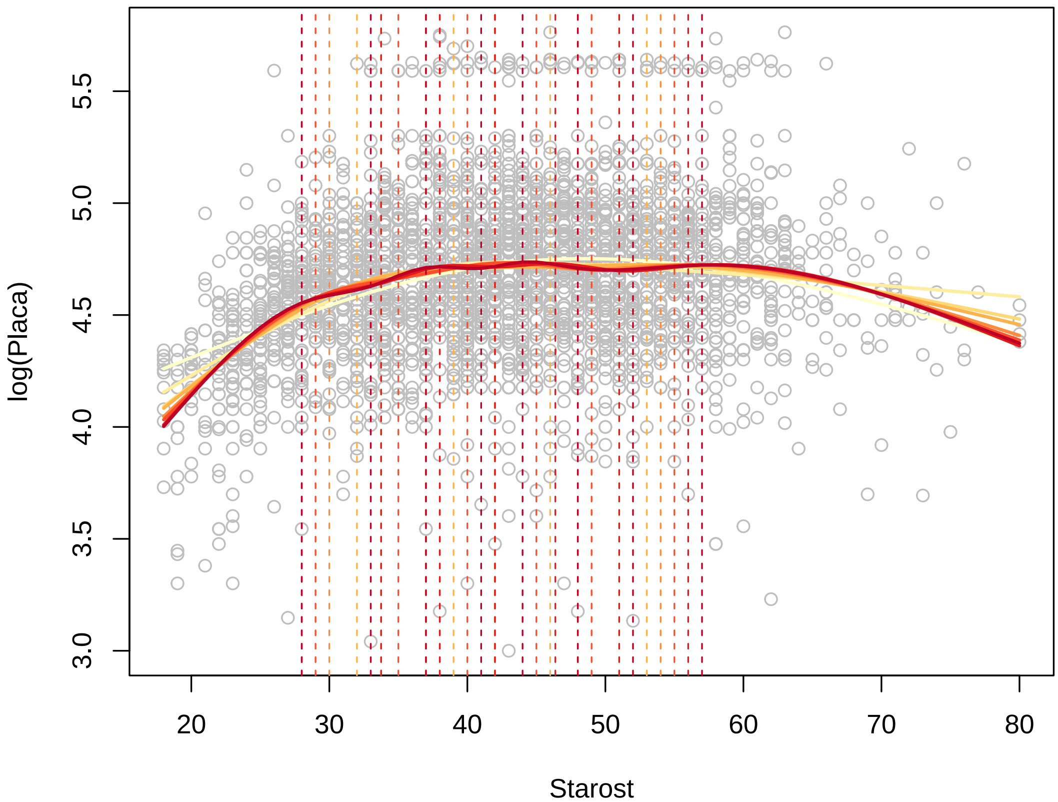
Kakšno je število ocenjenih parametrov v modelu z naravnimi zlepki glede na število vozlišč? V kakšnem razmerju je število ocenjenih parametrov, število vozlišč in število stopinj prostosti modela?

Narišite napovedi za `logwage` na podlagi vseh modelov z naravnimi zlepki. Opazujte, kakšne so razlike med napovedmi. Zapišite svoje ugotovitve.

```r
age.nap <- seq(from = min(Wage$age), to = max(Wage$age))

napovedi.ns3 <- predict(model.ns3, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns4 <- predict(model.ns4, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns5 <- predict(model.ns5, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns6 <- predict(model.ns6, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns7 <- predict(model.ns7, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns8 <- predict(model.ns8, newdata =data.frame(age=age.nap),
                        interval="confidence")
napovedi.ns9 <- predict(model.ns9, newdata =data.frame(age=age.nap),
```

```
                        interval="confidence")
napovedi.ns10 <- predict(model.ns10, newdata =data.frame(age=age.nap),
                        interval="confidence")
```

```
plot(Wage$age, Wage$logwage, col ="gray", xlab="Starost", ylab="log(Plača)")

lines(age.nap, napovedi.ns3[,"fit"], col="#FFFFCC", lwd=2)
abline(v=attr(ns(Wage$age, df = 2),"knots"), col="#FFFFCC", lty=2)

lines(age.nap, napovedi.ns4[,"fit"], col="#FFEDA0", lwd=2)
abline(v=attr(ns(Wage$age, df = 3),"knots"), col="#FFEDA0", lty=2)

lines(age.nap, napovedi.ns5[,"fit"], col="#FED976", lwd=2)
abline(v=attr(ns(Wage$age, df = 4),"knots"), col="#FED976", lty=2)

lines(age.nap, napovedi.ns6[,"fit"], col="#FEB24C", lwd=2)
abline(v=attr(ns(Wage$age, df = 5),"knots"), col="#FEB24C", lty=2)

lines(age.nap, napovedi.ns7[,"fit"], col="#FD8D3C", lwd=2)
abline(v=attr(ns(Wage$age, df = 6),"knots"), col="#FD8D3C", lty=2)

lines(age.nap, napovedi.ns8[,"fit"], col="#FC4E2A", lwd=2)
abline(v=attr(ns(Wage$age, df = 7),"knots"), col="#FC4E2A", lty=2)

lines(age.nap, napovedi.ns9[,"fit"], col="#E31A1C", lwd=2)
abline(v=attr(ns(Wage$age, df = 8),"knots"), col="#E31A1C", lty=2)

lines(age.nap, napovedi.ns10[,"fit"], col="#BD0026", lwd=2)
abline(v=attr(ns(Wage$age, df = 9),"knots"), col="#BD0026", lty=2)
```



Slika 6: Povprečne napovedi logaritma plače (`logwage`) glede na starost (`age`), modelirano z naravnimi zlepki z različnim številom notranjih vozlišč.

Koliko vozlišč je najprimerneje uporabiti? Svojo izbiro utemeljite.

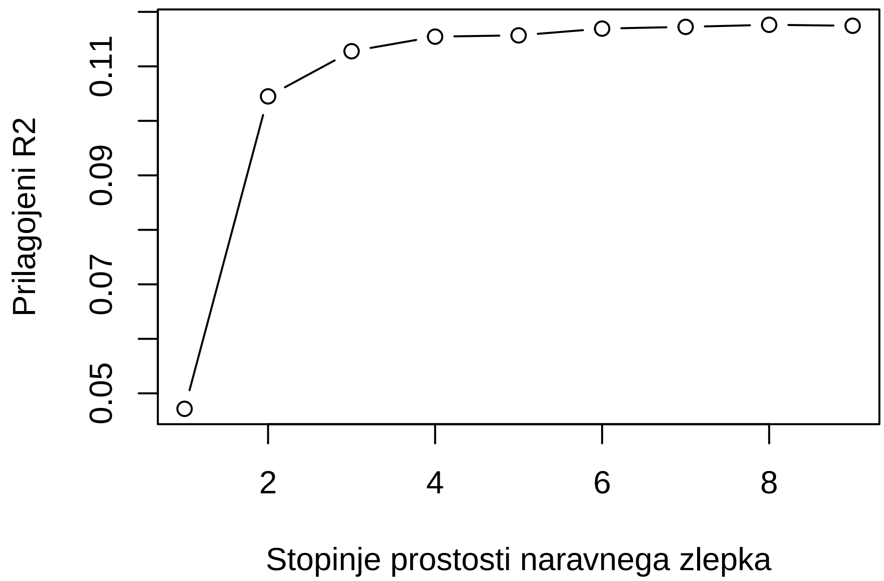
```
modeli <- list(model.stopnja1, model.ns3, model.ns4, model.ns5,
               model.ns6, model.ns7, model.ns8, model.ns9, model.ns10)
adj_r2 <- sapply(modeli, function(m) summary(m)$adj.r.squared)

which.max(adj_r2)
```

```
[1] 8
```

```
round(adj_r2, 5)
```

```
[1] 0.04716 0.10449 0.11278 0.11545 0.11569 0.11693 0.11724 0.11763 0.11745
```

```
plot(1:9, adj_r2, type="b",
     xlab="Stopinje prostosti naravnega zlepka",
     ylab="Prilagojeni R2")
```



Slika 7: Prilagojene vrednosti R^2 v odvisnosti od števila stopinj prostosti naravnega zlepka.

Slika jasno nakazuje izboljšanje v primerjavi z linearnih modelom. Izboljšanje je za modele z več kot 3 stopinjami prostosti zlepka minimalno, zato izberemo `model.ns4`.

Zakaj v tem primeru F-testa ne moremo uporabiti, kot smo ga pri polinomski regresiji?

F-test lahko uporabimo za primerjavo gnezdenih modelov. Kadar pa kompleksnost zlepka variiramo tako, da povečujemo število stopinj prostosti, bodo vsakič tudi položaji vozlišč drugačni in s tem bazne funkcije. Ne gre torej za gnezdene modele!

Poglejmo si izpis povzetka izbranega modela:

```r
summary(model.ns4)
```

```

Call:
lm(formula = logwage ~ ns(age, df = 3), data = Wage)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.73711 -0.19170  0.00649  0.18711  1.14965 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       4.15556    0.02913 142.653   <2e-16 ***
ns(age, df = 3)1  0.32811    0.02798  11.726   <2e-16 ***
ns(age, df = 3)2  1.01329    0.07534  13.449   <2e-16 ***
ns(age, df = 3)3  0.12074    0.05962   2.025   0.0429 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.3313 on 2996 degrees of freedom
Multiple R-squared:  0.1137,    Adjusted R-squared:  0.1128
F-statistic: 128.1 on 3 and 2996 DF,  p-value: < 2.2e-16
```

Kaj lahko zaključimo na podlagi tega izpisa? Kako bi model lahko interpretirali?

Funkcija `ns` ločuje med zunanjimi in notranjimi vozlišči (nekateri `R`-ovi paketi, npr. `rcs` v `Hmisc` pa ne). V kolikor bi želeli model definirati na podlagi vozlišč, argument `knots` določa notranja vozlišča, argument `Boundary.knots` pa zunanji vozlišči. Primerjajmo napovedi modela `model.ns4`, kjer so vozlišča definirana glede na prizvete nastavitve, z modelom, kjer vozlišča določimo glede na Harrellova priporočila (dokumetacija `?rcspline.eval` v paketu `Hmisc`.)

```
st.vozlisc <- 4

#Harrell priporocila
zunanja <- if (st.vozlisc > 3) 0.05 else 0.1
if (st.vozlisc > 6) {
  zunanja <- 0.025
}

p <- seq(zunanja, 1 - zunanja, length = st.vozlisc)
#vozlisca Harrell
(vsi <- quantile(Wage$age, p, na.rm = TRUE))
```

```
 5% 35% 65% 95%
 24  38  47  61
```

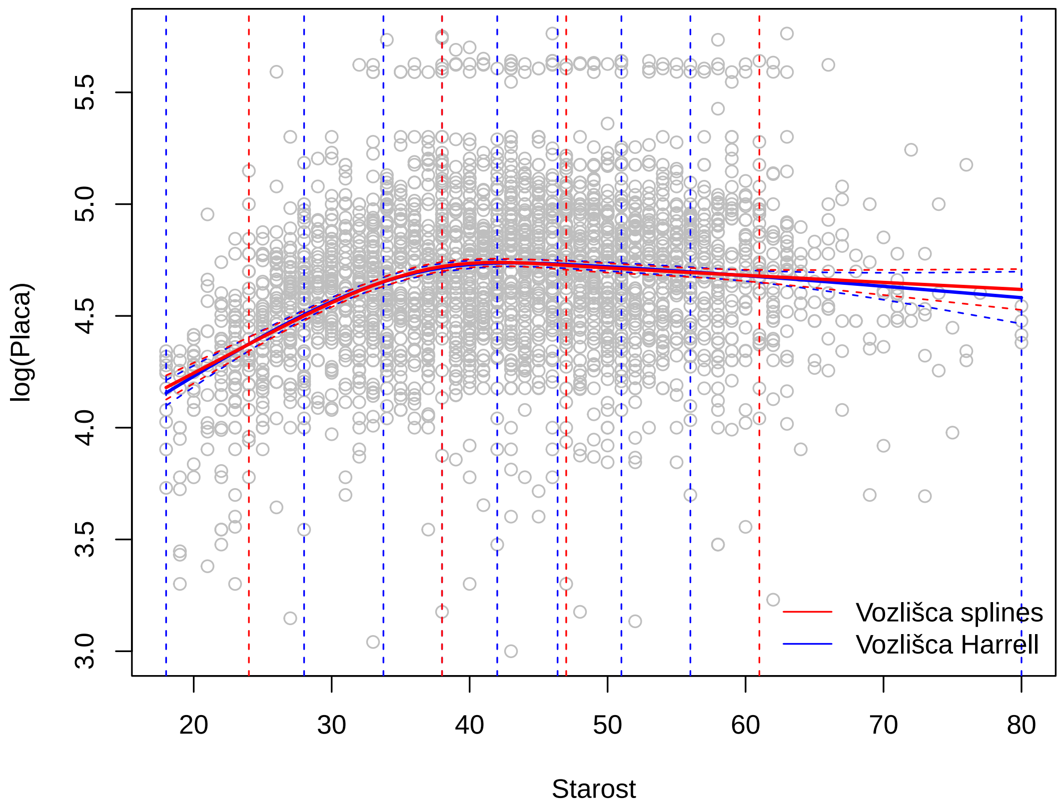
```
#attr(ns(Wage$age, df = 8), "knots")
#attr(ns(Wage$age, df = 8), "Boundary.knots")

Boundary.knots <- c(vsi[1], vsi[st.vozlisc])
knots <- vsi[-c(1, st.vozlisc)]

model.ns4.Harrell <- lm(logwage ~ ns(age, knots = knots,
                                     Boundary.knots =  Boundary.knots), data = Wage)

c(summary(model.ns4)$adj.r.squared, summary(model.ns4.Harrell)$adj.r.squared)
```

```
[1] 0.1127760 0.1111989
```

```
age.nap <- seq(from = min(Wage$age), to = max(Wage$age))

napovedi.ns4 <- predict(model.ns4, newdata =data.frame(age=age.nap),
                        interval="confidence")

napovedi.ns4.Harrell <- predict(model.ns4.Harrell, newdata =data.frame(age=age.nap),
                        interval="confidence")
```

```
plot(Wage$age, Wage$logwage, col ="gray",  xlab="Starost", ylab="log(Plača)")
lines(age.nap, napovedi.ns4[,"fit"], col="blue", lwd=2)
abline(v=c(min(Wage$age), attr(ns(Wage$age, df = 8),"knots"), max(Wage$age)), col="blue", lty=2)
matlines(age.nap, napovedi.ns4[, c("lwr","upr")], lwd = 1, col = "blue", lty = 2)

lines(age.nap, napovedi.ns4.Harrell[,"fit"], col="red", lwd=2)
abline(v=vsi, col="red", lty=2)
matlines(age.nap, napovedi.ns4.Harrell[, c("lwr","upr")], lwd = 1, col = " red", lty = 2)
```

```r
legend("bottomright", c("Vozlišča splines", "Vozlišča Harrell"),
       lty=c(1,1), col=c("red", "blue"), bty="n")
```



Slika 8: Povprečne napovedi logaritma plače (`logwage`) glede na starost (`age`), modelirano z naravnimi zlepki z različno postavljenimi vozlišči, s 95 % intervali zaupanja.

Rekli smo, da lahko dani parameter v linearnem modelu interpretiramo kot razliko v odvisni spremenljivki za dve osebi, ki se za 1 enoto razlikujeta v vrednostih dane napovedne spremenljivke, medtem ko so vrednosti ostalih napovednih spremenljivk konstante. Takšna interpretacija pri modelu, ki vsebuje polinome ali zlepke, ni mogoča, saj si ne moremo predstavljati, da bi recimo lahko spremenili vrednost spremenljivke X^2, medtem ko bi bila vrednost X nespremenjena.

Kadar v modelu sprostimo predpostavko linearnosti, žrtvujemo del interpretabilnosti modela za bolj fleksibilen model, na podlagi katerega lahko dobimo bolj natančne napovedi. Posameznih parametrov v takem modelu se ne da interpretirati. Pri interpretaciji pa si pomagamo z grafičnimi prikazi.

Interpretacija: Zveza med `log(wage)` in `age` je v modelu nelinearna. `log(wage)` narašča z `age` nekje do 40. leta, po tem letu pa se stabilizira oz. rahlo pade.

Domača naloga: Interpretacija modela z zlepki

1. Modelirajte `logwage` v odvisnosti od starosti (`age`) in izobrazbe (`education`), kot smo to naredili v prejšnji vaji, vendar v modelu ustrezno modelirajte nelinearnost. Pri diagnostiki si pomagaje z orodji, ki smo jih tekom predmeta obravnavali. Rezultate interpretirajte! Pri interpretaciji si pomagajte z ustreznimi grafičnimi prikazi.