Kazalo

1 NELINEARNOST

Linearnost odvisnosti odzivne spremenljivke od napovedne spremenljivke ob upoštevanju ostalih spremenljivk v modelu se v praksi velikokrat pokaže kot precej slaba aproksimacija dejanske odvisnosti. Obstaja več načinov modeliranja nelinearnosti v kontekstu linearnih modelov. Najpreprostejši sta polinomska regresija in regresija po odsekih (*step function regression* in *picewise regression*), kompleksnejše metode so regresija zlepkov (*regression splines*), glajeni zlepki (*smoothing splines*), lokalna regresija (*local regression*) in posplošeni aditivni modeli (*Generalized Additive Models*, GAM). V tem poglavju bomo predstavili polinomsko regresijo in regresijo zlepkov.

1.1 Polinomska regresija

Zgodovinsko gledano predstavlja polinomska regresija najstarejši način modeliranja nelinearne odvisnosti odzivne od napovedne spremenljivke. Osnovni linearni model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{1}$$

v tem primeru razširimo v polinomom stopnje p, $p \geq 2$:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_p x_i^p + \varepsilon_i. \tag{2}$$

Ob upoštevanju $x = x_1$, $x^2 = x_2$, ..., $x^p = x_p$, lahko izraz (2) zapišemo kot model, ki vključuje več številskih napovednih spremenljivk:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}.$$

Spremenljivke $x_1, ..., x_p$ so med seboj odvisne (multikolinearnost), kar pa ne predstavlja večjih težav, saj običajno na podlagi takega modela ne preverjamo ničelnih domnev o posameznih parametrih modela, bolj nas zanimajo napovedi. Parametri $\beta_0, \beta_1, ..., \beta_p$ so v linearnem odnosu z y, nimajo pa vsebinskega pomena.

Če želimo preveriti, ali je linearna odvisnost y od x upravičena, preverjamo sestavljeno ničelno domnevo $H_0 : \beta_2 = \beta_3 = ... = \beta_p = 0$. To naredimo z F-testom za gnezdene modele.

Pri modeliranju s polinomsko regresijo se v praksi skušamo omejiti na polinome nižjih stopenj, $p = 2$ do 4. Pri polinomih stopnje več kot 4 hitro pride do preprileganja podatkov, še posebej na robovih prostora napovedne spremenljivke. Namesto polinomov višjih stopenj je v določenih primerih bolje uporabiti nelinearne regresijske modele, pri katerih se parametri dajo vsebinsko interpretirati ali pa regresijo zlepkov.

1.1.1 Primer: KORUZA

Za rezultate bločnega poskusa s koruzo v letu 1990 (KORUZA.txt) analizirajmo, kako je pridelek koruze (kg/ha) odvisen od gostote setve. Zanima nas optimalen pridelek koruze, optimalna gostota setve in njuna 95 % intervala zaupanja. Poskus je bil zasnovan kot bločni poskus v 3 ponovitvah (blokih), v poskusu je bilo 15 različnih gostot setve. Znotraj vsakega bloka (dela njive) so bile enkrat ponovljene vse gostote setve. Ker tudi blok lahko vpliva na pridelek (npr. zaradi različnih rastnih pogojev), bomo vpliv gostote setve na pridelek modelirali ob upoštevanju vpliva bloka. V tem primeru blok vključimo v model kot opisno spremenljivko, njen vpliv pa je slučajen (o tem več v poglavju o mešanih linearnih modelih).

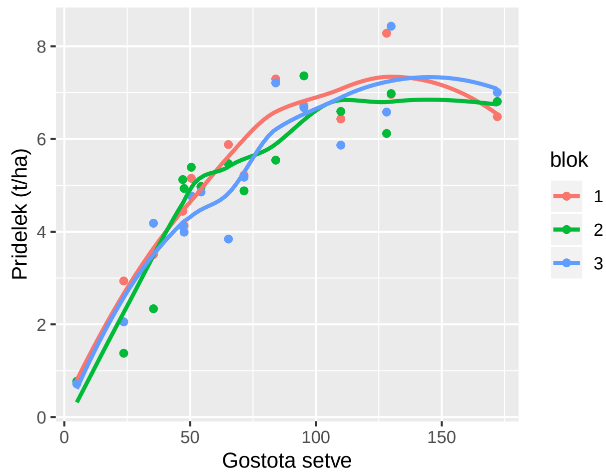
```
> koruza<-read.table(file="KORUZA.txt", header = TRUE)
> str(koruza)

'data.frame':        45 obs. of  5 variables:
 $ blok      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ gostsetve : num  65.2 23.6 129.9 50.5 47.6 ...
 $ gostvznika: num  51.9 23.5 123.8 49.5 41.3 ...
 $ prid.ha   : int  5880 2936 6962 5152 4129 720 5219 6481 3508 6719 ...
 $ prid.rast : num  0.09 0.124 0.054 0.102 0.087 0.145 0.073 0.038 0.101 0.071 ...

> summary(koruza[,c("gostsetve", "prid.ha")])

   gostsetve          prid.ha
 Min.   :  4.967   Min.   : 717
 1st Qu.: 47.080   1st Qu.:4129
 Median : 65.230   Median :5176
 Mean   : 74.632   Mean   :5095
 3rd Qu.:109.900   3rd Qu.:6595
 Max.   :172.100   Max.   :8433
```

Spremenljivka `prid.ha` je izražena v kg/ha, zaradi lažje interpretacije jo pretvorimo v t/ha, spremenljivko `blok` pa spremenimo v factor:

```
> koruza$prid1.ha<-koruza$prid.ha/1000
> koruza$blok <- factor(koruza$blok)
```

Slika 1 prikazuje odvisnost pridelka koruze od gostote setve in od bloka s pripadajočimi gladilniki.

```
> library(ggplot2)
> ggplot(data=koruza, aes(x=gostsetve, y=prid1.ha, col=blok)) +
+          geom_point() + geom_smooth(se=FALSE) +
+          ylab("Pridelek (t/ha)") +
+          xlab("Gostota setve")
```

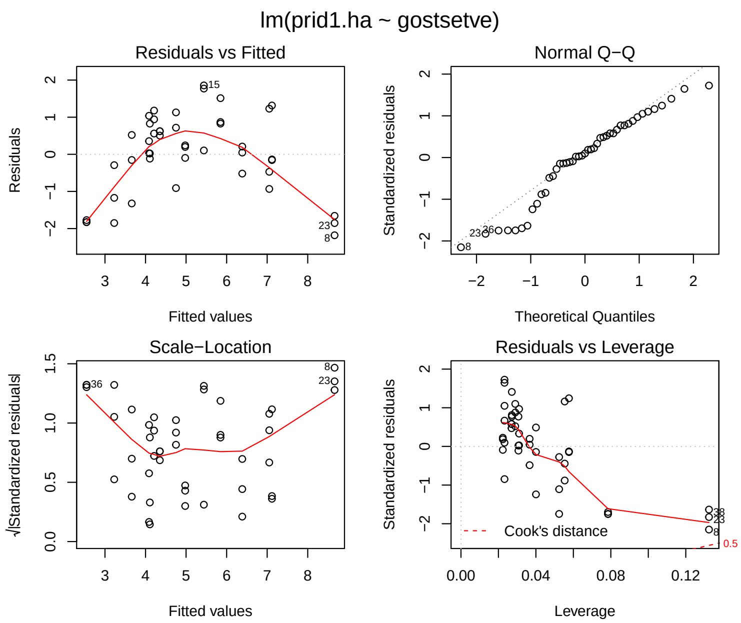


Slika 1: Odvisnost pridelka (t/ha) od gostote setve in od bloka z dodanim gladilnikom

Slika 1 ter vsebinski premislek nakazujejo, da pridelek ni linearno odvisen od gostote setve. Vidimo, da je odvisnost za vse tri bloke zelo podobna. Naredimo najprej neustrezeni linearni model in hkrati poglejmo, ali lahko vpliv blokov zanemarimo.

```
> model.lin <- lm(prid1.ha ~ gostsetve, data=koruza)
> model.lin.blok <- lm(prid1.ha ~ blok + gostsetve, data=koruza)
> anova(model.lin, model.lin.blok)

Analysis of Variance Table

Model 1: prid1.ha ~ gostsetve
Model 2: prid1.ha ~ blok + gostsetve
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     43 50.941
2     41 50.187  2   0.75458 0.3082 0.7364
```

Modela `model.lin` in `model.lin.blok` sta ekvivalentna, zato nadaljujemo s preprastejšim modelom, ki kot napovedno spremenljivko vključuje samo `gostsetve`.



Slika 2: Ostanki za `model.lin`

Ostanki kažejo, da `model.lin` ni sprejemljiv, na Grafu 1 je gladilnik v obliki parabole, zato model dopolnimo s kvadratnim členom:

```
> model.kvad <- lm(prid1.ha ~ gostsetve + I(gostsetve^2), data=koruza)
> # enak rezulat dobimo z uporabo funkcije poly
> # model.kvad.1 <- lm(prid1.ha ~ poly(gostsetve, degree=2, raw=TRUE), data=koruza)
```

V zapisu `lm` modela uporabimo funkcijo `I()`, ki zagotovi, da izraz $\texttt{gostsetve}^2$ določa regresor v linearnem modelu, znak za potenco, kot tudi ostali aritmetični operatorji (*,/,+,-), ima v formuli modela poseben pomen in s tem je ta pomen omejen na potenciranje `gostsetve`.

V `model.kvad` sta regresorja korelirana in posledično je VIF vrednost visoka, kar v tem primeru ignoriramo.

```
> library(car)
> vif(model.kvad)
```

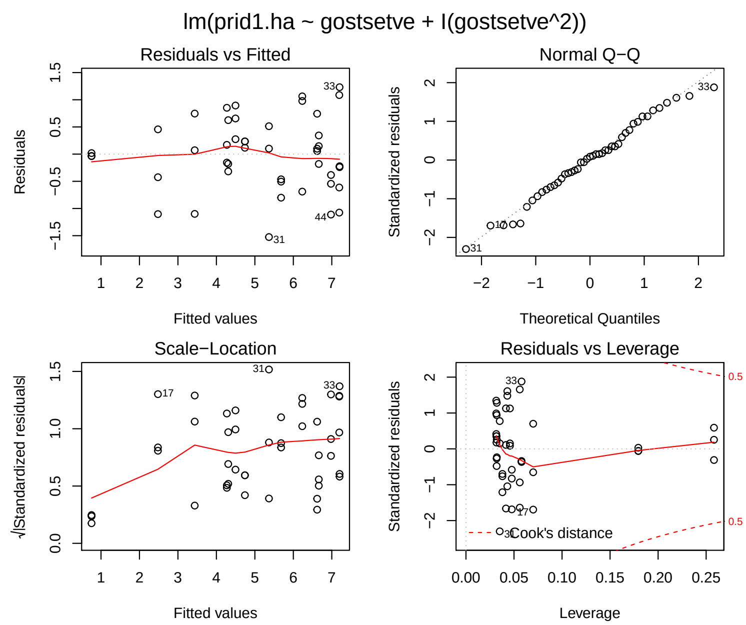
```
     gostsetve I(gostsetve^2)
      13.16808       13.16808
```

Naredimo primerjavo `model.lin` in `model.kvad`. Preverjamo ničelno domnevo, da sta modela ekvivalentna. F-test za dva gnezdena modela izvedemo s funkcijo `anova`

```
> anova(model.kvad)
```

```
Analysis of Variance Table

Response: prid1.ha
               Df  Sum Sq Mean Sq F value    Pr(>F)
gostsetve       1 115.495 115.495 253.488 < 2.2e-16 ***
I(gostsetve^2)  1  31.805  31.805  69.807 1.798e-10 ***
Residuals      42  19.136   0.456
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model z dodanim kvadratnim členom je statistično značilno boljši od modela brez kvadratnega člena ($F = 69.8, p < 0.0001$).



Slika 3: Ostanki za `model.kvad`

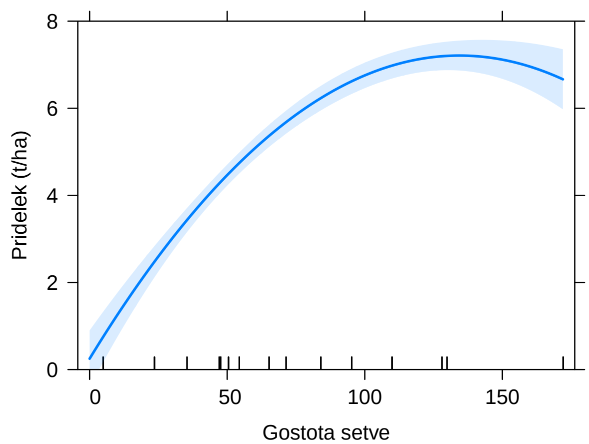
Naraščajoč gladilnik na Sliki 3 levo spodaj je posledica treh podatkov pri zelo nizki vrednosti napovedanega pridelka, če te točke odmislimo, težav z nekonstntno varianco ni videti in lahko rečemo, da je `model.kvad` sprejemljiv.

```
> coef(summary(model.kvad))

                    Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)     0.2491926865 3.218808e-01  0.774177 4.431625e-01
gostsetve       0.1035883074 8.341093e-03 12.419033 1.195171e-15
I(gostsetve^2) -0.0003853948 4.612724e-05 -8.355037 1.797520e-10

> summary(model.kvad)$r.squared

[1] 0.8850243
```

Napišimo enačbo parabole:

$$\hat{y} = 0.24919 + 0.10359x + (-0.00039)x^2.$$

```
> library(effects)
> plot(Effect(c("gostsetve"), model.kvad, xlevels=list(gostsetve=seq(0, 172, 2))),
+       ci.style="bands", xlab="Gostota setve",ylab="Pridelek (t/ha)",
+       main="", ylim=c(0,8))
```



Slika 4: Odvisnost pridelka (t/ha) od gostote setve in parabola izračunana z `model.kvad` ter 95 % intervali zaupanja za povprečne napovedi

Optimalna gostota setve

Izračunajmo optimumalno gostoto setve, optimalni pridelek in pripadajoči 95 % IZ. Optimum izračunamo z odvajanjem kvadratne enačbe po x:

```
> ocene<-summary(model.kvad)$coef[1:3]; ocene

[1]  0.2491926865  0.1035883074 -0.0003853948

> opt<--0.5*ocene[2]/ocene[3]
> cat("Optimalna gostota =", opt)

Optimalna gostota = 134.3925
```

```
> # Napoved in interval zaupanja za pridelek pri optimalni gostoti
> gostsetve.x<-data.frame(gostsetve=opt)
> povp.napoved.pridelek<-predict(model.kvad,gostsetve.x, interval="confidence")
> round(data.frame(cbind(gostsetve.x,povp.napoved.pridelek)), 1)

  gostsetve fit lwr upr
1     134.4 7.2 6.9 7.6
```

Interpretacija rezultatov: pri optimalni gostoti 134.39 je pričakovana vrednost pridelka 7.21 t/ha, pripadajoči 95 % IZ pa je (6.87 t/ha, 7.55 t/ha).

Intervalna ocena za optimalno gostoto setve

Optimum je izračunan kot razmerje dveh normalno porazdeljenih slučajnih spremenljivk in je tudi slučajna spremenljivka. Zanima nas njen interval zaupanja, za to rabimo pripadajočo varianco. Tega ne znamo dobiti analitično, uporabimo lahko eno izmed metod samovzorčenja.

Z **neparametričnim bootstrap pristopom** (Efron, 1979) iz osnovnega vzorca velikosti n tvorimo t. i. **bootstrap vzorce**. Vsak bootstrap vzorec ima n enot. Enote (s pripadajočimi vrednostmi odzivne in napovednih spremenljivk) vzorčimo z enostavnim slučajnim vzorčenjem s ponavljanjem. Takemu načinu samovzorčenja v kontekstu linearnih modelov pravimo **samovzorčenje primerov** (*case resampling*). Tvorimo R bootstrap vzorcev, R je veliko število (1000 in več). Za vsak bootstrap vzorec izračunamo vzorčno oceno iskane statistike. Na osnovi R bootstrap vzorcev dobimo njeno **bootstrap vzorčno porazdelitev**. 95 % **centilni bootstrap interval zaupanja** je določen z 2.5 in 97.5 centilom te porazdelitve.

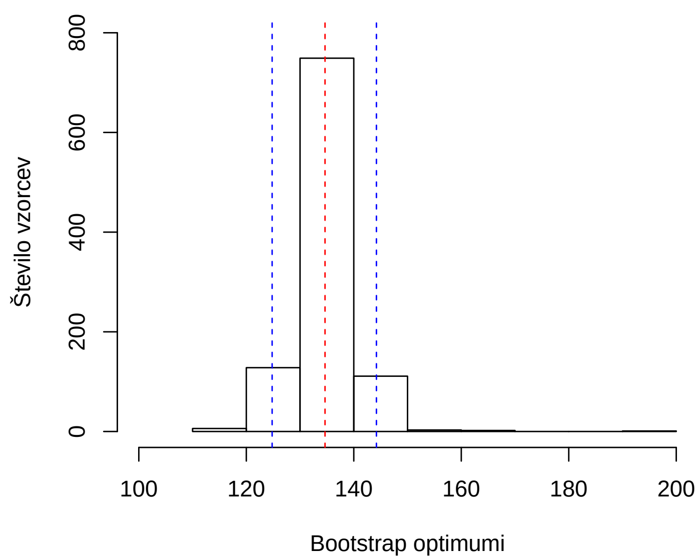
Za samovzorčenje primerov za `model.kvad` bomo uporabili funkcijo `Boot` iz paketa `car`. Ta funkcija z osnovnimi argumenti naredi samovzorčenje primerov za neparametrični bootstrap za modele vrste `lm`, `glm` in `nls`.

```
> library(car)
> set.seed(3435)   ## zaradi ponovljivosti
> betahat.boot<-Boot(model.kvad, R=1000,  f = coef, method = "case")
> ## za vsak vzorec imamo v matriki betahat.boot ocene parametrov modela
> ## matriko spremenimo v data.frame, dodamo izračun optimuma za vsak vzorec
> summary(betahat.boot)

Number of bootstrap replications R = 1000
                 original    bootBias     bootSE     bootMed
(Intercept)     0.24919269  4.3947e-04 2.4854e-01  0.24800661
gostsetve       0.10358831  8.4170e-05 6.7941e-03  0.10353290
I(gostsetve^2) -0.00038539 -7.4505e-07 3.6259e-05 -0.00038468
```

```
> head(betahat.boot$t)

     (Intercept)  gostsetve I(gostsetve^2)
[1,]   0.2985552 0.10210209  -0.0003746396
[2,]   0.3248422 0.09717609  -0.0003509286
[3,]   0.1361163 0.10472212  -0.0003929578
[4,]   0.6434663 0.09357845  -0.0003394483
[5,]   0.5507324 0.09730414  -0.0003568802
[6,]   0.4113874 0.09751018  -0.0003508506

> # betahat.boot<-as.data.frame(betahat.boot$t)
> betahat.boot$t$opt<--0.5*betahat.boot$t[,2]/betahat.boot$t[,3]
> ## izračun IZ za optimalno gostoto
> bootIZ<-quantile(betahat.boot$t$opt,c(0.025,0.975))
> round(cbind(mean(betahat.boot$t$opt),t(bootIZ)), 2)

              2.5%  97.5%
[1,] 134.65 124.81 144.21
```

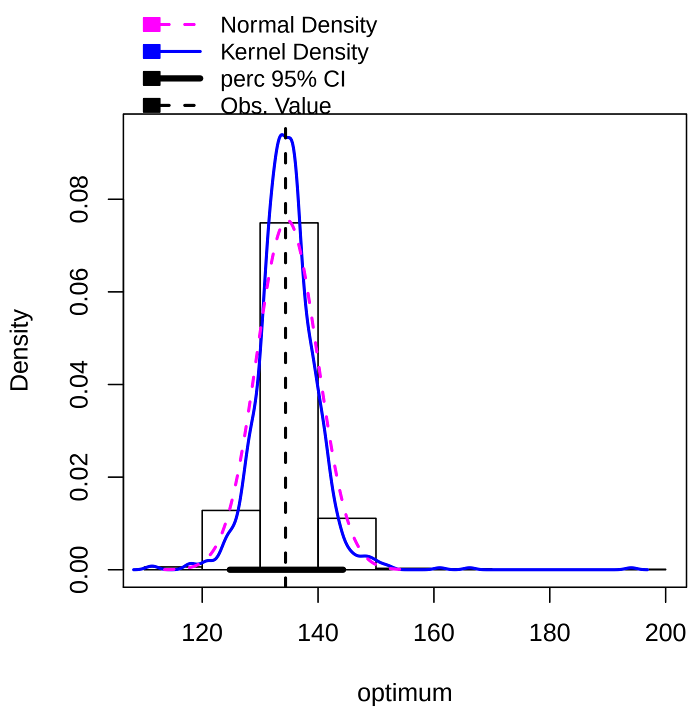
Optimalna gostota setve je 134.6, pripadajoči 95 % centilni bootstrap IZ je (124.8, 144.2). Grafični prikaz dobljene porazdelitve 1000 bootstrap optimumov je na Sliki 5.

```
> hist(betahat.boot$t$opt, ylim=c(0, 800), xlim=c(100, 200),
+      xlab="Bootstrap optimumi",  main="", ylab="Število vzorcev")
> abline(v=mean(betahat.boot$t$opt), lty=2, col="red"); abline(v=bootIZ, lty=2, col="blue")
```



Slika 5: Porazdelitev 1000 bootstrap optimumov za gostoto setve, vodoravne črte predstavljajo izračunani povprečni optimum (sredina) in 95 % centilni bootstrap interval zaupanja za optimum

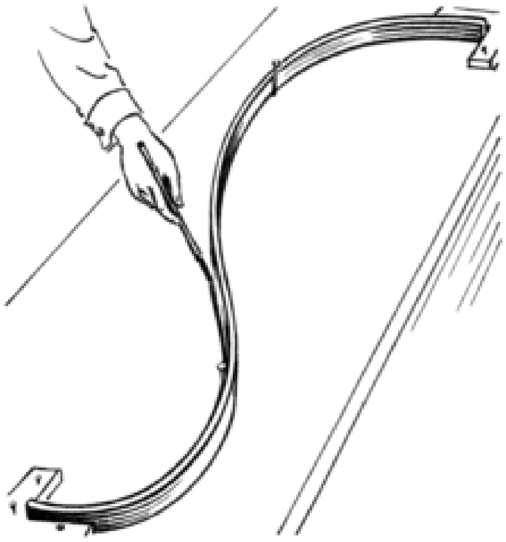
```
> # še druga možnost uporabe funkcije Boot za bootstrap optimume:
> set.seed(3435)
> betahat.boot.1<-Boot(object=model.kvad, R=1000, labels="optimum",
+                      f = function(object) -0.5*coef(object)[2]/coef(object)[3])
> summary(betahat.boot.1)

           R original bootBias bootSE bootMed
optimum 1000   134.39  0.25634 5.2841  134.52

> opt.bootstrap<-as.numeric(summary(betahat.boot.1)[2]+summary(betahat.boot.1)[3])
> round(opt.bootstrap,2)

[1] 134.65

> confint(betahat.boot.1, type="perc")

Bootstrap percent confidence intervals

          2.5 %   97.5 %
optimum 124.759 144.3246

> hist(betahat.boot.1, ci="perc")
```



Slika 6: Histogram za bootstrap optimume gostote setve na podlagi `model.kvad`

1.2 Regresija zlepkov

Zlepek (*spline*) v angleškem jeziku prestavlja dolg in tenek upogljiv kos lesa ali kovine, ki so ga načrtovalci/konstruktorji uporabljali za risanje krivulj skozi vnaprej določene točke (Slika 7). Zlepke v regresijski analizi uporabljamo za opis nelinearnega odnosa med odzivno in izbrano napovedno spremenljivko.



Slika 7: Zlepek z dvema vozliščema oziroma s tremi odseki (Vir: Wikipedia)

Pogosto se zgodi, da nelinearnosti ne moremo opisati s polinomsko regresijo dovolj nizke stopnje. V takem primeru lahko vrednosti napovedne spremenljivke razdelimo na odseke in na posameznem odseku uporabimo polinomsko regresijo nižje stopnje. Takemu načinu modeliranja pravimo **polinomska regresija po odsekih** (*piecewise polynomials*). Na primer, če vrednosti spremenljivke x razdelimo na dva odseka: $x < c$ in $x \geq c$ in izberemo polinom tretje stopnje, v modelu polinomske regresije na odsekih ocenjujemo osem parametrov modela:

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \varepsilon_i, & x_i < c, \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \varepsilon_i, & x_i \geq c. \end{cases} \tag{3}$$

Z modelom (3) podatkom prilogodimo dve polinomski funkciji, eno za podatke z $x_i < c$ in drugo za podatke z $x_i \geq c$. Vrednost spremenljivke $x = c$, kjer se vrednosti parametrov polinoma spremenijo, se imenuje **vozlišče** (*knot*). Več vozlišč omogoča bolj kompleksno nelinearno odvisnost. Če postavimo K vozlišč znotraj intervala vrednosti spremenljivke x, prilagodimo $K + 1$ polinomov izbrane stopnje p. V vozliščih je potrebno definirati, kako naj se polinoma stikata. Najbolj uporabno je, da se polinoma stikata zvezno in gladko, v takem primeru govorimo o **regresiji zlepkov**.

Zlepek je funkcija, ki opiše krivuljo na izbranih odsekih napovedne spremenljivke x. Odseki so določeni z vozlišči. Na posameznem odseku odvisnost odzivne spremenljivke od x opiše polinom p-te stopnje. V vozliščih se vrednosti sosednjih polinomov gladko stikajo, kar pomeni, da pri ocenjevanju parametrov polinomov postavimo še dodatne pogoje: v vozlišču morata imeti sosednja polinoma stopnje p isto vrednost in iste vrednosti odvodov reda od $1, ..., p - 1$. Ti dodatni pogoji zmanjšajo število parametrov, ki jih moramo oceniti v modelu.

Število vozlišč K izberemo vnaprej, prav tako njihove vrednosti; izbira je odvisna od števila podatkov, kompleksnosti nelinearnosti na posameznih odsekih in od predhodnega poznavanja procesa, ki ga modeliramo.

1.2.1 Bazne funkcije

Za razumevanje regresije zlepkov najprej definirajmo t. i. **bazne funkcije**. Polinomska regresija predstavlja poseben primer pristopa regresijskega modeliranja z baznimi funkcijami. Ideja baznih funkcij je v tem, da napovedno spremenljivko x v model vključimo v obliki različnih transformacij oziroma v obliki k baznih funkcij: $b_1(x), b_2(x), ..., b_k(x)$. V modelu bazne funkcije predstavljajo regresorje:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + ... + \beta_k b_k(x_i) + \varepsilon_i. \tag{4}$$

Bazne funkcije so vedno izbrane vnaprej. V kontekstu polinomske regresije stopnje p imamo $k = p$ baznih funkcij $b_j(x) = x^j$, $j = 1, ..., p$. Če gre za ortogonalne polinome, potem bazne funkcije predstavljajo linearno kombinacijo regresorjev x^j, $j = 1, ..., p$, ki ustreza pogoju, da so bazne funkcije med seboj neodvisne.

V primeru modeliranja z baznimi funkcijami parametre modela (4) ocenimo po metodi najmanjših kvadratov. Če so splošne predpostavke linearnega modela izpolnjene, je inferenca na ocenah parametrov enaka kot v primeru navadnih regresorjev.

Bazne funkcije lahko predstavljajo zelo različne funkcije napovedne spremenljivke, zelo pogosto so določene kot kombinacija polinomov nižjih stopenj (največ tretje).

1.2.2 Linearni zlepki

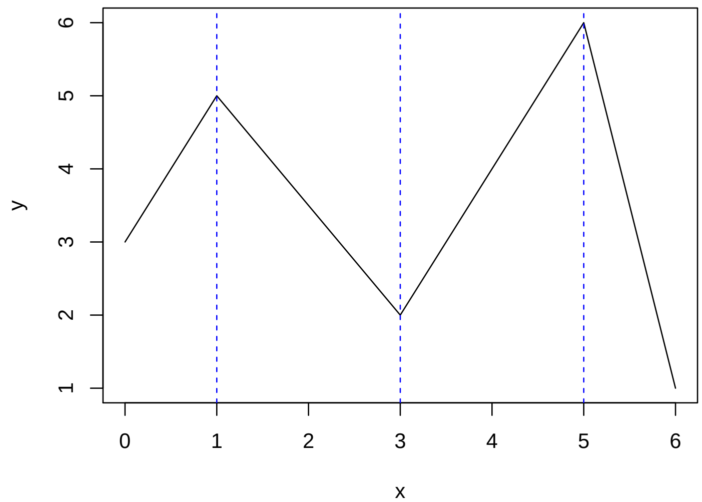
Linearni zlepki predstavljajo linearno funkcijo, ki se lomi v K vozliščih. Za primer poglejmo linearne zlepke s tremi vozlišči $K = 3$, torej so vrednosti napovedne spremenljivke x razdeljene na štiri odseke pri vozliščih $x = a_1$, $x = a_2$ in $x = a_3$. Model linearnega zlepka predstavlja funkcija:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - a_1)_+ + \beta_3 (x_i - a_2)_+ + \beta_4 (x_i - a_3)_+ + \varepsilon_i, \tag{5}$$

kjer velja $(u)_+ = u, u > 0$ in $(u)_+ = 0, u \leq 0$. V (5) je bazna funkcija x_i, $(x_i - a_1)_+$, $(x_i - a_2)_+$ in $(x_i -_{a\ 3})_+$ so t. i. **odrezane bazne funkcije** (*truncated basis*).

Enačbo (5) lahko zapišemo po odsekih napovedne spremenljivke x:

$$\begin{aligned}
y_i = &\beta_0 + \beta_1 x_i + \varepsilon_i, && x_i \leq a,\\
&\beta_0 + \beta_1 x_i + \beta_2 (x_i - a_1) + \varepsilon_i, && a_1 < x_i \leq a_2,\\
&\beta_0 + \beta_1 x_i + \beta_2 (x_i - a_1) + \beta_3 (x_i - a_2) + \varepsilon_i, && a_2 < x_i \leq a_3,\\
&\beta_0 + \beta_1 x_i + \beta_2 (x_i - a_1) + \beta_3 (x_i - a_2) + \beta_4 (x_i - a_3) + \varepsilon_i, && a_3 < x_i.
\end{aligned} \tag{6}$$



Slika 8: Linearni zlepek z vozlišči pri $a_1 = 1$, $a_2 = 3$ in $a_3 = 5$

Linearne zlepke v regresijski model s K vozlišči pri vrednostih $(a_1, a_2, ..., a_K)$ vključimo z baznimi funkcijami $b_1(x) = x$, $b_2(x) = (x - a_1)_+$ do $b_{K+1}(x) = (x - a_K)_+$, njihovo število je določeno s številom vozišč $K + 1$:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K+1} b_{K+1}(x_i) + \varepsilon_i. \tag{7}$$

Ocene $K + 1$ parametrov modela (7) izračunamo po metodi najmanjših kvadratov ob dodatnem pogoju, da se vrednosti $\hat{y}$ stikajo v vozliščih. Linearnost odvisnosti y od x testiramo z ničelno domnevo $H_0 : \beta_2 = \beta_3 = ... = \beta_{K+1} = 0$. Uporabimo F-test za dva gnezdena modela.

Linearni zlepki so preprosti in z njimi lahko opišemo veliko odnosov, njihova slabost pa je, da se funkcija v vozliščih prelomi. Če želimo modelirati gladke krivulje, moramo za opis nelinearnosti na posameznih odsekih uporabiti polinome višjih stopenj.

1.2.3 Kubični zlepki

Praksa je pokazala, da imajo polinomi tretje stopnje (kubični polinomi) lepe lastnosti in sposobnost, da ob primerni izbiri števila vozlišč opišejo tudi zelo kompleksne nelinearne odvisnosti. Dva kubična polinoma se gladko stikata v vozlišču, če v vozlišču poleg vrednosti izenačimo tudi njun prvi in drugi odvod. Model **kubičnega zlepka** za tri vozlišča ($K = 3$) je:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 (x_i - a_1)_+^3 + \beta_5 (x_i - a_2)_+^3 + \beta_6 (x_i - a_3)_+^3 + \varepsilon_i, \tag{8}$$

kjer so bazne funkcije in odrezane potenčne bazne funkcije (*truncated power basis*)

$$\begin{aligned} &b_1(x) = x, && b_2(x) = x^2, && b_3(x) = x^3, \\ &b_4(x) = (x - a_1)_+^3, && b_5(x) = (x - a_2)_+^3, && b_6(x) = (x - a_3)_+^3. \end{aligned} \tag{9}$$

Model kubičnega zlepka za K-vozlišč je:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i, \tag{10}$$

kjer so prve tri bazne funkcije določene enako kot v (9), vse naslednje pa:

$$h(x, a_j) = (x - a_j)_+^3 = \begin{cases} (x - a_j)^3, & x > a_j \\ 0, & \text{drugače}, \end{cases} \tag{11}$$

a_j, $j = 1, ..., K$ so vozlišča.

Parametri modela regresijskih zlepkov ($\beta_0, ..., \beta_{K+3}$) so izračunani po metodi najmanjših kvadratov z upoštevanimi dodatnimi pogoji, ki zagotavljajo, da so njihovi stiki v vozliščih gladki. Če ima kubični zlepek K vozlišč, moramo v regresijskem modelu poleg presečišča oceniti $K+3$ parametrov.

1.2.4 Naravni zlepki

Praksa je pokazala, da se pri kubičnih regresijskih zlepkih pogosto zgodi, da se slabo obnesejo na prvem in zadnjem odseku (pred prvim in za zadnjim vozliščem). To težavo rešimo z uporabo t. i. **naravnih zlepkov** (*natural splines* ali *restricted cubic splines*). V tem primeru predpostavimo linearni odnos med y in x na prvem in zadnjem odseku. Posledično v modelu ocenjujemo poleg presečišča samo $K - 1$ parametrov: odpadeta parametra pri x^2 in x^3, zadnja dva parametra β_K in β_{K+1} se zapišeta kot linearna kombinacija predhodnih parametrov $\beta_2, ..., \beta_{K-1}$ (16). Regresijski model z naravnimi zlepki zapišemo:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - a_1)_+^3 + \beta_3 (x_i - a_2)_+^3 + ... + \beta_{K+1} (x_i - a_K)_+^3 + \varepsilon, \tag{12}$$

$a_1, ..., a_k$ so vozlišča. V procesu ocenjevanja parametrov modela naravnih zlepkov najprej ocenjujemo K parametrov na podlagi funkcije:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K-1} b_{K-1}(x_i), \tag{13}$$

kjer je $b_1(x_i) = x_i$, ostale bazne funkcije so za $j = 2, ..., K - 1$ izražene takole:

$$b_{j+1}(x_i) = (x - a_j)_+^3 - (x - a_{K-1})_+^3 (a_K - t_j)/(t_K - a_{K-1}) + (x - a_K)_+^3 (a_{K-1} - a_j)/(a_K - a_{K-1}). \tag{14}$$

Pokažemo lahko, da je bazna funkcija $b_{K+1}(x)$ na zadnjem odseku ($x \geq a_K$) linearna. Na podlagi ocen $\hat{\beta}_0, ..., \hat{\beta}_{K-1}$ se izračuna še oceni $\hat{\beta}_K$ in $\hat{\beta}_{K+1}$ iz (17):

$$\hat{\beta}_K = [\hat{\beta}_2 (a_1 - a_K) + \hat{\beta}_3 (a_2 - a_K) + ... + \hat{\beta}_{K-1} (a_{K-2} - a_K)]/(a_K - a_{K-1}), \tag{15}$$

$$\hat{\beta}_{K+1} = [\hat{\beta}_2 (a_1 - a_{K-1}) + \hat{\beta}_3 (a_2 - a_{K-1}) + ... + \hat{\beta}_{K-1} (a_{K-2} - a_{K-1})]/(a_{K-1} - a_K). \tag{16}$$

Pri modeliranju regresijskih zlepkov je število in položaj vozlišč določeno vnaprej. Položaj vozlišč lahko določimo na podlagi predhodnega poznavanja procesa, ki ga modeliramo; na primer, če vemo, da se naklon spremeni pri vrednosti $x = a$, vrednost a vnaprej izberemo za vozlišče. V splošnem smiselnih vrednosti za vozlišča ne poznamo. Analize so pokazale, da samo položaj vozlišč ni tako pomemben, bolj pomembno je število vozlišč. Položaj vozlišč po navadi določimo z vrednostmi enako razmaknjenih kvantilov napovedne spremenljivke x. S tem zagotovimo, da je število podatkov na vseh odsekih uravnoteženo. Pogosto sta prvi in zadnji odsek ob uporabi naravnih zlepkov manjša, priporočene kvantile kaže Tabela 1.

Tabela 1: Priporočeni kvantilni rangi za vozlišča naravnih zlepkov (Harrell F. E., 2015)

Število vozlišč k	Kvantilni rangi
3	.10, .5, .90
4	.05, .35, .65, .95
5	.05, .275, .5, .725, .95
6	.05, .23, .41, .59, .77, .95
7	.025, .1833, .3417, .5, .6583, .8167, .975

Če imamo malo podatkov ($n \leq 30$), običajno izberemo $K = 3$, sicer je najpogostejša primerna izbira $K = 4$ ali $K = 5$. Za velik n ($n \geq 100$), je običajno primerno število vozlišč $K = 7$, večje vrednosti za K so zelo redko potrebne.

Število potrebnih vozlišč v praksi pogosto določimo na podlagi navzkrižnega preverjanja modela (*cross-validation*). Ta postopek bomo spoznali v enem izmed poglavij, ki sledijo.

1.2.5 Primer: KORUZA (nadaljevanje)

Regresijo naravnih zlepkov bomo uporabili na primeru napovedi pridelka koruze v odvisnosti od gostote setve (datoteka KORUZA.txt). Za regresijo zlepkov potrebujemo paket `splines`. Za modeliranje zlepkov stopnje p s K vozlišči uporabimo funkcijo `bs`, za modeliranje naravnih zlepkov pa funkcijo `ns`.

Najprej poglejmo argumente funkcije `bs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE, Boundary.knots = range(x))`:

- prvi argument funkcije `bs` je vektor napovedne spremenljivke x, v našem primeru bo to gostota setve (`gostsetve`). Če je to edini argument, funkcija vrne tri bazne funkcije za polinom tretje stopnje, ker ima po prednastavitvi argument `degree` vrednost 3 ($p = 3$), argument `knots` pa `NULL` ($K = 0$);

  ```
  > library(splines)
  > #  df=NULL, knots=NULL
  > bs.0<-bs(koruza$gostsetve)
  > str(bs.0)
  ```

  ```
   'bs' num [1:45, 1:3] 0.442 0.264 0.143 0.433 0.425 ...
   - attr(*, "dimnames")=List of 2
  ```

```
  ..$ : NULL
  ..$ : chr [1:3] "1" "2" "3"
 - attr(*, "degree")= int 3
 - attr(*, "knots")= num(0)
 - attr(*, "Boundary.knots")= num [1:2] 4.97 172.1
 - attr(*, "intercept")= logi FALSE
```

```
> head(bs.0)
```

```
             1          2           3
[1,] 0.4422797 0.24939741 0.046877627
[2,] 0.2641465 0.03316373 0.001387908
[3,] 0.1429673 0.42325434 0.417681154
[4,] 0.4325936 0.16188647 0.020193880
[5,] 0.4247011 0.14552358 0.016621190
[6,] 0.0000000 0.00000000 0.000000000
```

- argument `df` predstavlja število stopinj prostosti regresijskega modela `degree+K`; s tem argumentom lahko posredno nastavimo število vozlišč K=`df-degree`; po prednastavitvi ima vrednost `NULL`, kar pomeni, da je $K = 0$;

```
> #  df=4, knots=NULL
> bs.1<-bs(koruza$gostsetve, df=4)
> str(bs.1)
```

```
 'bs' num [1:45, 1:4] 0.4089 0.5816 0.0252 0.5797 0.6071 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:4] "1" "2" "3" "4"
 - attr(*, "degree")= int 3
 - attr(*, "knots")= Named num 65.2
  ..- attr(*, "names")= chr "50%"
 - attr(*, "Boundary.knots")= num [1:2] 4.97 172.1
 - attr(*, "intercept")= logi FALSE
```

```
> head(bs.1)
```

```
              1          2           3         4
[1,] 0.40887184 0.46111806 0.130010096 0.0000000
[2,] 0.58157783 0.08514991 0.003849216 0.0000000
[3,] 0.02517428 0.20938968 0.543850631 0.2215854
[4,] 0.57967645 0.34965495 0.056005570 0.0000000
[5,] 0.60710389 0.32184584 0.046097098 0.0000000
[6,] 0.00000000 0.00000000 0.000000000 0.0000000
```

- z argumentom `knots` nastavimo položaje vozlišč, vrednosti vozlišč zapišemo v vektor. Če ima vrednost `NULL` in je argument `df` različen od `NULL`, se položaji vozlišč določijo na podlagi kvantilnih rangov, ki vrednosti za x razdelijo na enake dele glede na število vozlišč. Na primer če je `df=5` in `degree=3`, vozlišča predstavljata 33.3 centil in 66.7. centil vrednosti x;

- argument `intercept` ima po prednastavitvi vrednost `FALSE`, kar pomeni, da presečišče ni vključeno pri računanju baznih funkcij zlepka, to je priročno za uporabo funkcije `bs` v formuli modela `lm`;
- argument `Boundary.knots` ima po prednastavitvi vrednosti `min(x)` in `max(x)` ter določa razpon vrednosti spremenljivke `x`, na katerem se računajo bazne funkcije zlepkov.

Ilustracija baznih funkcij kubičnih zlepkov s tremi vozlišči določenimi s kvartili `gostsetve`:

```
> # določimo vrednosti gostsetve za vozlišča
> vozl<-quantile(koruza$gostsetve, probs = c(0.25, 0.5, 0.75), na.rm=T)
> bs.3<-bs(koruza$gostsetve, knots=vozl, degree=3)
> # enakovreden ukaz je
> # bs.3<-bs(koruza$gostsetve,df=6)
> str(bs.3)
```

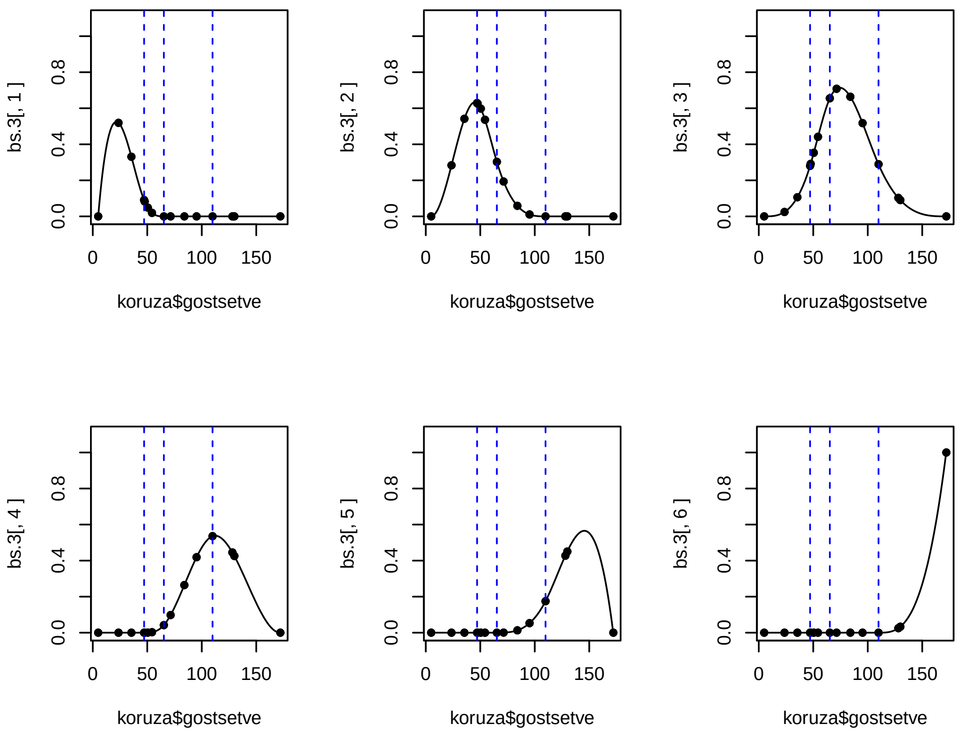
```
 'bs' num [1:45, 1:6] 0 0.519 0 0.0487 0.0829 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:6] "1" "2" "3" "4" ...
 - attr(*, "degree")= int 3
 - attr(*, "knots")= Named num [1:3] 47.1 65.2 109.9
  ..- attr(*, "names")= chr [1:3] "25%" "50%" "75%"
 - attr(*, "Boundary.knots")= num [1:2] 4.97 172.1
 - attr(*, "intercept")= logi FALSE
```

```
> # vrednosti baznih funkcij zlepkov polinomov tretje stopnje s tremi vozlišči
> # za dane vrednosti gostsetve
> head(bs.3)
```

```
              1         2          3            4         5          6
[1,] 0.00000000 0.3027066 0.65534883 4.194458e-02 0.0000000 0.00000000
[2,] 0.51904398 0.2835271 0.02433147 0.000000e+00 0.0000000 0.00000000
[3,] 0.00000000 0.0000000 0.09042982 4.257325e-01 0.4505933 0.03324443
[4,] 0.04868531 0.5981136 0.35292537 2.757292e-04 0.0000000 0.00000000
[5,] 0.08285142 0.6257652 0.29138232 1.104656e-06 0.0000000 0.00000000
[6,] 0.00000000 0.0000000 0.00000000 0.000000e+00 0.0000000 0.00000000
```

Za ilustracijo Slika 9 prikazuje vrednosti baznih funkcij za regresijo kubičnih zlepkov s tremi vozlišči.

```
> par(mfrow=c(2,dim(bs.3)[2]/2))
> x<-seq(min(koruza$gostsetve), max(koruza$gostsetve),1 )
> for (i in 1:dim(bs.3)[2])
+  {plot(koruza$gostsetve,bs.3[,i], pch=16, ylim=c(0,1.1),
+        ylab=paste("bs.3[,", i,"]"))
+   lines(x, bs(x, knots=vozl)[,i])
+   abline(v=vozl, col="blue", lty=2)}
```



Slika 9: Grafična predstavitev baznih funkcij kubičnega zlepka s tremi vozlišči, $K = 3$, ki predstavljajo kvartile `gostsetve`

Modelirajmo odvisnost pridelka koruze od gostote setve z uporabo zlepkov. Najprej bomo naredili model, ki je enakovreden modelu polinomske regresije reda 2. Funkcija `bs` ima argument `degree` $= 2$, število vozlišč je 0 (`knots=NULL`, prednastavitev):

```
> # model regresije kvadratnih zlepkov brez vozlišč
> model.bs2.0<-lm(pridl.ha ~ bs(gostsetve, degree=2), data=koruza)
> coef(summary(model.bs2.0))

                           Estimate Std. Error   t value     Pr(>|t|)
(Intercept)               0.7542077  0.2858920  2.638086 1.164340e-02
bs(gostsetve, degree = 2)1 8.3365769  0.6603105 12.625238 6.903840e-16
bs(gostsetve, degree = 2)2 5.9077512  0.3849820 15.345524 7.976758e-19
```

```
> # za primerjavo izpišimo ocene parametrov polinomske regresije
> coef(summary(model.kvad))

                    Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)     0.2491926865 3.218808e-01  0.774177 4.431625e-01
gostsetve       0.1035883074 8.341093e-03 12.419033 1.195171e-15
I(gostsetve^2) -0.0003853948 4.612724e-05 -8.355037 1.797520e-10

> # še polinomska regresija z baznimi funkcijami,
> # ki predstavljajo ortogonalne kvadratne polinome regresorjev, degree=2, raw=FALSE
> model.kvad.1 <-lm(prid1.ha ~ poly(gostsetve, 2), data=koruza)
> coef(summary(model.kvad.1))

                     Estimate Std. Error   t value     Pr(>|t|)
(Intercept)          5.095022  0.1006226 50.634946 2.781172e-39
poly(gostsetve, 2)1 10.746839  0.6749972 15.921309 2.119810e-19
poly(gostsetve, 2)2 -5.639627  0.6749972 -8.355037 1.797520e-10

> # koeficienti determinacije za vse tri modele
> summary(model.kvad)$r.squared

[1] 0.8850243

> summary(model.kvad.1)$r.squared

[1] 0.8850243

> summary(model.bs2.0)$r.squared

[1] 0.8850243
```

Ocene parametrov za `model.kvad`, `model.kvad.1` in `model.bs2.0` so različne, ker so modelske matrike različne, koeficienti determinacije pa so isti in skoraj identične so tudi napovedi modelov (Slika 10).

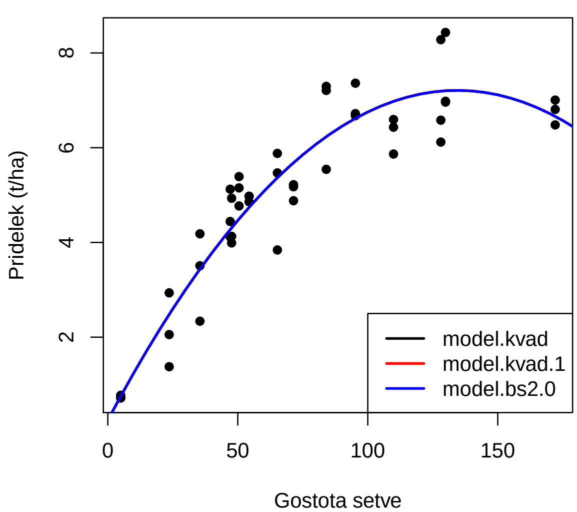
```
> head(model.matrix(model.kvad))

  (Intercept) gostsetve I(gostsetve^2)
1           1    65.230     4254.95290
2           1    23.610      557.43210
3           1   129.900    16874.01000
4           1    50.480     2548.23040
5           1    47.620     2267.66440
6           1     4.967       24.67109

> head(model.matrix(model.kvad.1))
```

```
  (Intercept) poly(gostsetve, 2)1 poly(gostsetve, 2)2
1           1         -0.03201627         -0.10913359
2           1         -0.17374630          0.13258280
3           1          0.18820671         -0.01498477
4           1         -0.08224496         -0.05055419
5           1         -0.09198421         -0.03575398
6           1         -0.23723195          0.31763110
```

```
> head(model.matrix(model.bs2.0))
```

```
  (Intercept) bs(gostsetve, degree = 2)1 bs(gostsetve, degree = 2)2
1           1                  0.4611181                 0.13001010
2           1                  0.1982068                 0.01244249
3           1                  0.3774811                 0.55876593
4           1                  0.3963200                 0.07415604
5           1                  0.3801498                 0.06512905
6           1                  0.0000000                 0.00000000
```

```
> novi.x<-data.frame(gostsetve=seq(0, 180, 5))
> plot(koruza$gostsetve,koruza$prid1.ha, pch=16,
+      ylab="Pridelek (t/ha)", xlab="Gostota setve",)
> lines(novi.x$gostsetve, predict(model.kvad, novi.x), lwd=2, col="black")
> lines(novi.x$gostsetve, predict(model.kvad.1, novi.x),lwd=2, col="red")
> lines(novi.x$gostsetve, predict(model.bs2.0, novi.x),lwd=2, col="blue")
> legend(100, 2.5, legend=c("model.kvad","model.kvad.1","model.bs2.0"),
+        col=c("black","red","blue"), lwd=2, lty=1)
```



Slika 10: Odvisnost pridelka (t/ha) od gostote setve: parabola izračunana z `model.kvad` in napovedi za `model.kvad.1` in `model.bs2.0`

Modelov `model.bs2.0` in `model.kvad` se ne da primerjati z F-testom, ker modela nista gnezdena, imata enako število parametrov. Vsoti kvadriranih ostankov sta enaki:

```
> anova(model.kvad, model.bs2.0)

Analysis of Variance Table

Model 1: prid1.ha ~ gostsetve + I(gostsetve^2)
Model 2: prid1.ha ~ bs(gostsetve, degree = 2)
  Res.Df    RSS Df   Sum of Sq F Pr(>F)
1     42 19.136
2     42 19.136  0 -7.1054e-15
```

V drugem primeru bomo pridelek koruze modelirali s kubičnimi in z naravnimi zlepki s tremi vozlišči določenimi s kvantilnimi rangi 0.25, 0.5 in 0.75.

```
> model.bs.3<-lm(prid1.ha ~ bs(gostsetve, knots=vozl), data=koruza)
> model.ns.3<-lm(prid1.ha ~ ns(gostsetve, knots=vozl), data=koruza)
> coef(summary(model.bs.3))

                              Estimate Std. Error    t value     Pr(>|t|)
(Intercept)                  0.7384488  0.3964574  1.8626184 7.026053e-02
bs(gostsetve, knots = vozl)1 0.3605557  0.9693291  0.3719642 7.119858e-01
bs(gostsetve, knots = vozl)2 3.7417868  0.6514026  5.7442000 1.277304e-06
bs(gostsetve, knots = vozl)3 4.8728190  0.6888817  7.0735212 1.956330e-08
bs(gostsetve, knots = vozl)4 6.8665761  1.1516307  5.9624808 6.409079e-07
bs(gostsetve, knots = vozl)5 6.3003290  1.3579431  4.6396119 4.064820e-05
bs(gostsetve, knots = vozl)6 6.0334575  0.5611930 10.7511268 4.391166e-13
```

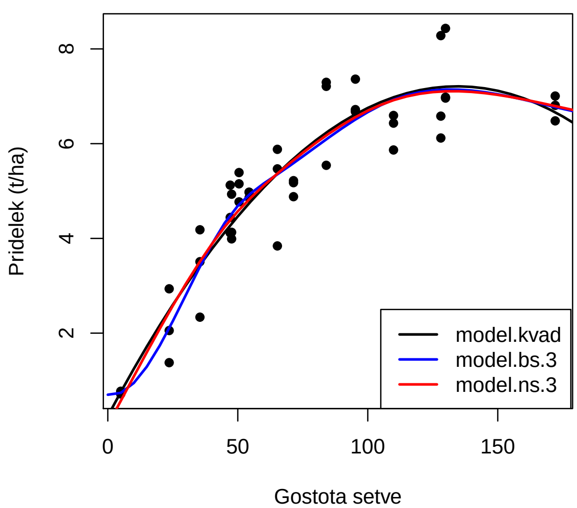
```
> coef(summary(model.ns.3))

                              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)                  0.5674518  0.3660588  1.550166 1.289785e-01
ns(gostsetve, knots = vozl)1 5.2579716  0.4883474 10.766867 2.197480e-13
ns(gostsetve, knots = vozl)2 5.7707941  0.4793138 12.039701 7.050328e-15
ns(gostsetve, knots = vozl)3 9.5461236  0.8772115 10.882351 1.594601e-13
ns(gostsetve, knots = vozl)4 4.2841747  0.4499559  9.521322 7.793808e-12
```

Primerjava modela polinomske regresije in regresijskih zlepkov:

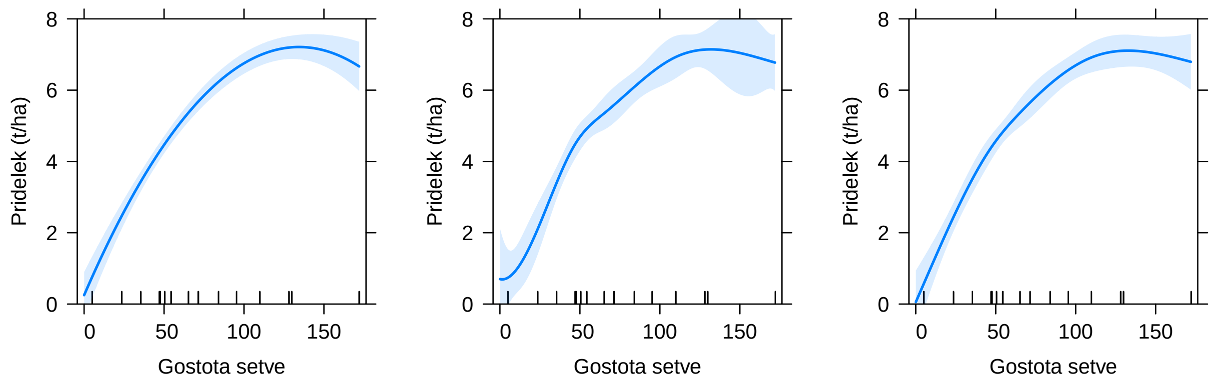
```
> anova(model.kvad, model.bs.3)

Analysis of Variance Table

Model 1: prid1.ha ~ gostsetve + I(gostsetve^2)
Model 2: prid1.ha ~ bs(gostsetve, knots = vozl)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     42 19.136
2     38 17.988  4    1.1478 0.6062 0.6606
```

```
> anova(model.kvad, model.ns.3)

Analysis of Variance Table

Model 1: prid1.ha ~ gostsetve + I(gostsetve^2)
Model 2: prid1.ha ~ ns(gostsetve, knots = vozl)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     42 19.136
2     40 18.645  2   0.49135 0.5271 0.5944
```

Za `model.kvad` velja, da se pridelek od optimalne gostote naprej enako hitro zmanjšuje, kot se je povečeval pred dosegom optimuma. Pri modelih z regresijskimi zlepki `model.bs.3` in `model.ns.3` pa je padec pridelka po optimumu počasnejši (Sliki 11, 12). Med modeli ni statistično značilnih razlik v pojasnjeni variabilnosti odzivne spremenljivke.



Slika 11: Odvisnost pridelka (t/ha) od gostote setve: parabola izračunana z `model.kvad` in napovedi za `model.bs.3` in `model.ns.3`



Slika 12: Odvisnost pridelka (t/ha) od gostote setve, napovedi za `model.kvad` (levo), `model.bs.3` (sredina) in za `model.ns.3` (desno) ter 95 % intervali zaupanja za povprečne napovedi

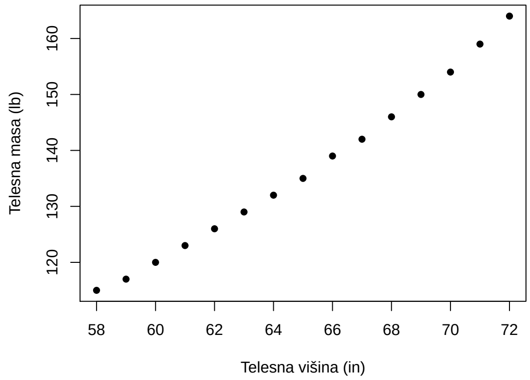
2 VAJE

2.1 Telesna masa in višina žensk

Analizirajte odvisnost telesne mase od telesne višine za ženske stare med 30 in 39 let. Podatki so v podatkovnem okviru `women` v paketu `stats`. Podatke grafično prikažite, naredite ustrezen model, preverite predpostavke izbranega modela, napovedi grafično prikažite in napišite obrazložitev statistične analize.

```
> str(women)

'data.frame':       15 obs. of  2 variables:
 $ height: num  58 59 60 61 62 63 64 65 66 67 ...
 $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

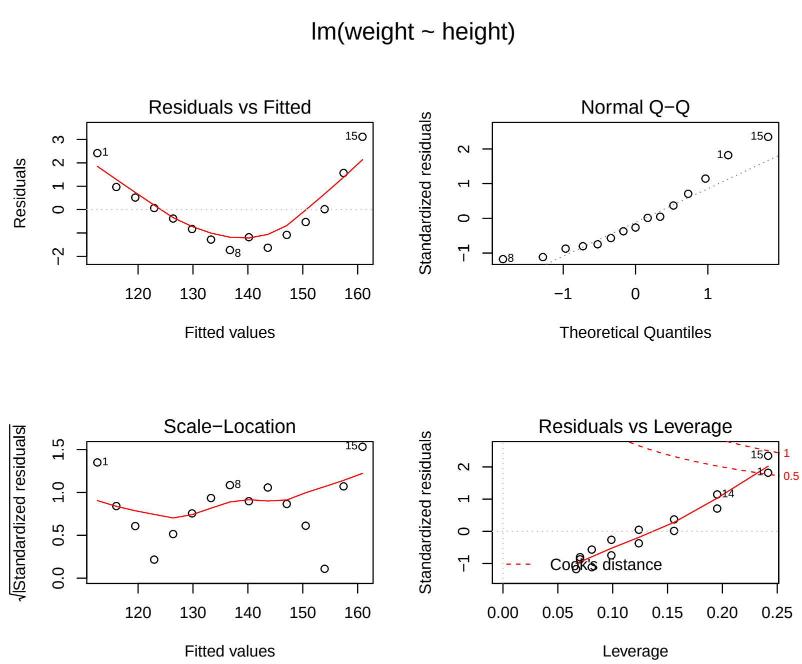
```
> plot(women, xlab = "Telesna višina (in)", ylab = "Telesna masa (lb)", pch=16)
```



Slika 13: Odvisnost telesne mase od telesne višine za ženske stare med 30 in 39 let

Na podlagi Slike 13 ocenimo, da je odvisnost približno linearna. Naredimo enostavi linearni model, za katerega se pokaže, da ostanki niso ustrezno porazdeljeni (Slika 14), kar kaže na nelinearni odnos med telesno maso in telesno višino.

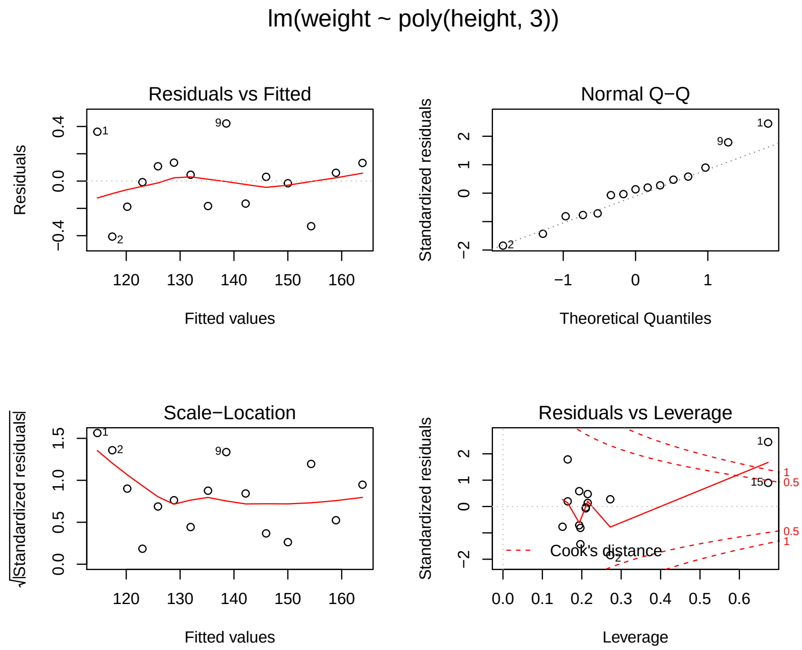
```
> mod.lin<-lm(weight~height, data=women)
```



Slika 14: Ostanki za `mod.lin`

Poskusimo najprej s polinomsko regresijo. Slika ostankov kaže, da bi bilo smiselno poskusiti s kvadratnim polinomom, morda tudi polinomom višje stopnje.

```
> mod.kvad<-lm(weight~poly(height,2), data=women)
> anova(mod.lin, mod.kvad)

Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ poly(height, 2)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     13 30.2333
2     12  1.7701  1    28.463 192.96 9.322e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> mod.kub<-lm(weight~poly(height,3), data=women)
> anova(mod.kvad, mod.kub)

Analysis of Variance Table

Model 1: weight ~ poly(height, 2)
Model 2: weight ~ poly(height, 3)
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     12 1.77007
2     11 0.73415  1    1.0359 15.522 0.002313 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> mod.4<-lm(weight~poly(height,4), data=women)
> anova(mod.kub, mod.4)

Analysis of Variance Table

Model 1: weight ~ poly(height, 3)
Model 2: weight ~ poly(height, 4)
  Res.Df     RSS Df Sum of Sq     F  Pr(>F)
1     11 0.73415
2     10 0.50360  1   0.23055 4.578 0.05807 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

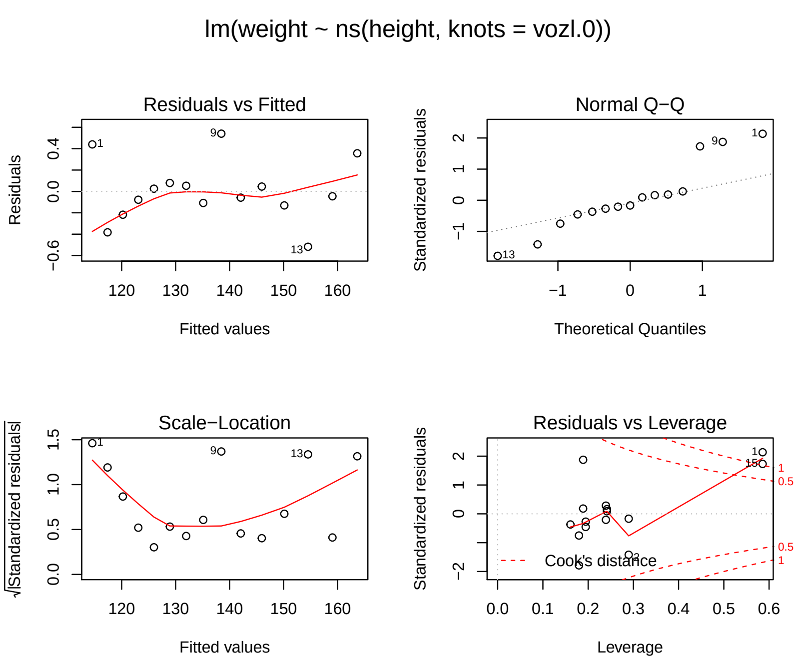


Slika 15: Ostanki za `mod.kub`

Poskusimo še z regresijo zlepkov. Nelinearnost bomo modelirali naravnimi kubičnimi zlepeki z dvema vozliščema.

```
> library(splines)
> vozl.0<-quantile(women$height, c(.33, .67));vozl.0

  33%   67%
62.62 67.38

> summary(mod.ns.2 <- lm(weight ~ ns(height, knots=vozl.0), data = women))

Call:
lm(formula = weight ~ ns(height, knots = vozl.0), data = women)

Residuals:
     Min       1Q   Median       3Q      Max
-0.51806 -0.11957 -0.04559  0.06589  0.54033

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 114.5600     0.2450  467.56  < 2e-16 ***
ns(height, knots = vozl.0)1  23.9074     0.3437   69.56 6.73e-16 ***
ns(height, knots = vozl.0)2  53.0911     0.6164   86.13  < 2e-16 ***
ns(height, knots = vozl.0)3  41.6421     0.2613  159.37  < 2e-16 ***
---
```

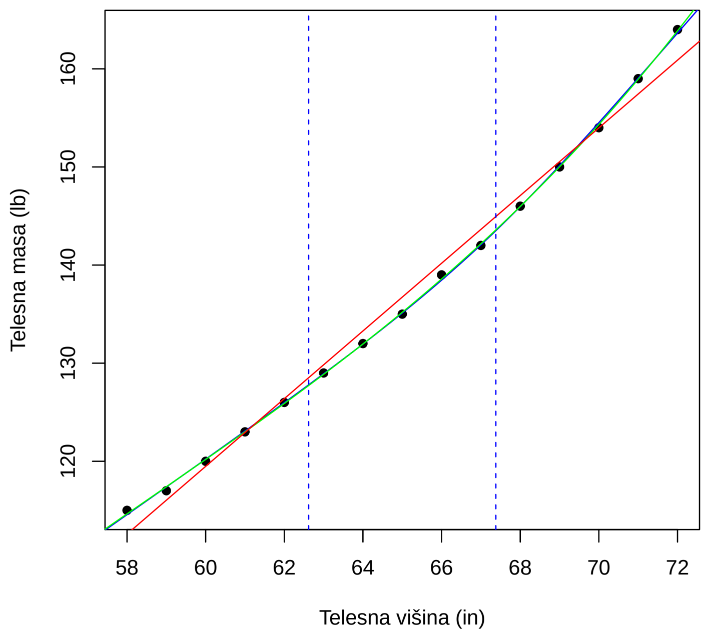
```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3201 on 11 degrees of freedom
Multiple R-squared:  0.9997,	Adjusted R-squared:  0.9996
F-statistic: 1.093e+04 on 3 and 11 DF,  p-value: < 2.2e-16

> anova(mod.lin, mod.ns.2)

Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ ns(height, knots = vozl.0)
  Res.Df     RSS Df Sum of Sq   F    Pr(>F)
1     13 30.2333
2     11  1.1274  2    29.106 142 1.392e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Slika 16: Ostanki za `mod.ns.2`

Model, ki vključuje naravne kubične zlepke z dvema vozliščema pojasni več variabilnosti oz. se bolje prilega podatkom kot linearni model. V modelu ocenjujemo dva parametra več kot v modelu enostavne linearne regresije.

```
> plot(women, xlab = "Telesna višina (in)", ylab = "Telesna masa (lb)", pch=16)
> ht <- seq(57, 73, length.out = 100)
> lines(ht, predict(mod.ns.2, data.frame(height = ht)), col="blue")
> lines(ht, predict(mod.kub, data.frame(height = ht)), col="green")
> abline(mod.lin, col="red")
> abline(v=c(62.62, 67.38),lty=2, col="blue")
```



Slika 17: Odvisnost telesne mase od telesne višine za ženske stare med 30 in 39 let z napovedmi `mod.lin` (rdeča), `mod.kub` in `mod.ns2` (modra), črtkane vertikalne črte predstavljajo vozlišča

2.2 Plača

V podatkovnem okviru `Wage` v paketu `ISLR` so podatki o plačah 3000 moških delavcev v srednje atlantski regiji. Analizirajmo odvisnost plače (`wage`) od starosti (`age`), leta pridobitve podatkov (`year`) in o izobrazbi delavcev (`education`).

Najprej analizirajmo odvisnost `wage` od `age`.

```
> library(ISLR)
> 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 ...
```

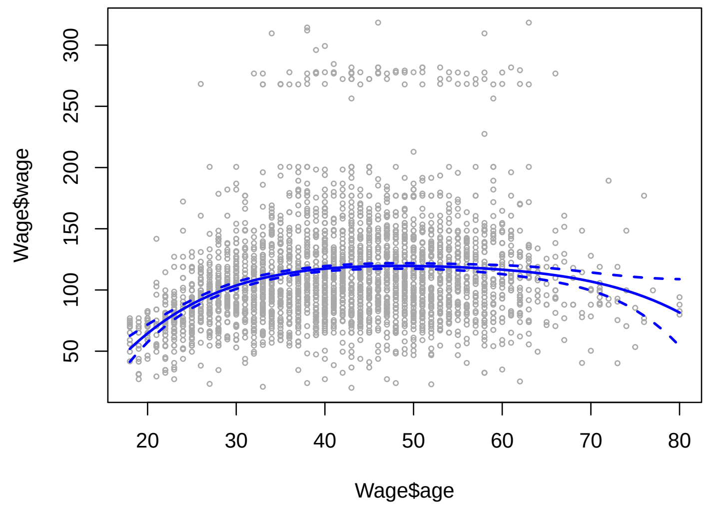
```
> mod.poly1=lm(wage~poly(age ,4) , data=Wage)
> coef(summary(mod.poly1))
```

```
                Estimate Std. Error    t value     Pr(>|t|)
(Intercept)    111.70361  0.7287409 153.283015 0.000000e+00
poly(age, 4)1  447.06785 39.9147851  11.200558 1.484604e-28
poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32
poly(age, 4)3  125.52169 39.9147851   3.144742 1.678622e-03
poly(age, 4)4  -77.91118 39.9147851  -1.951938 5.103865e-02
```

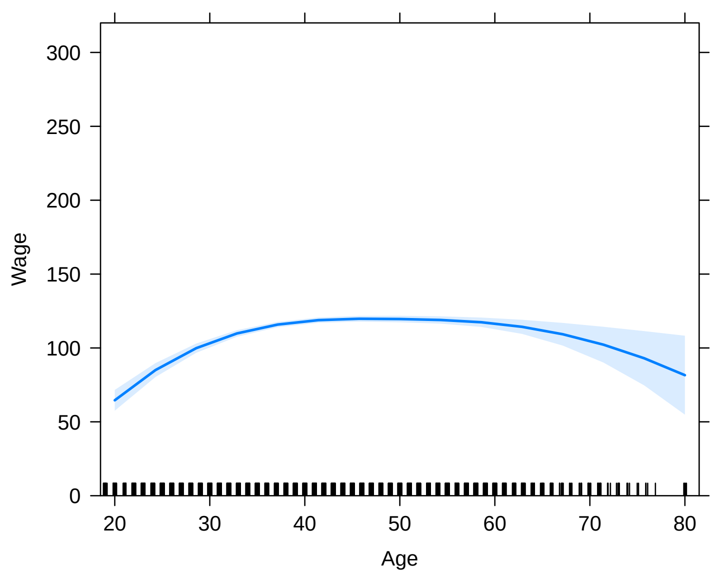
```
> mod.poly2=lm(wage~poly(age ,4, raw =T), data=Wage)
> coef(summary (mod.poly2))
```

```
                            Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)            -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
poly(age, 4, raw = T)1  2.124552e+01 5.886748e+00  3.609042 0.0003123618
poly(age, 4, raw = T)2 -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
poly(age, 4, raw = T)3  6.810688e-03 3.065931e-03  2.221409 0.0263977518
poly(age, 4, raw = T)4 -3.203830e-05 1.641359e-05 -1.951938 0.0510386498
```

```
> age.grid=seq (from=range(Wage$age)[1], to=range(Wage$age)[2])
> napovedi<-predict(mod.poly1 ,newdata =list(age=age.grid), se=TRUE)
> se.bands<-cbind(napovedi$fit + 2*napovedi$se.fit, napovedi$fit - 2*napovedi$se.fit)
```



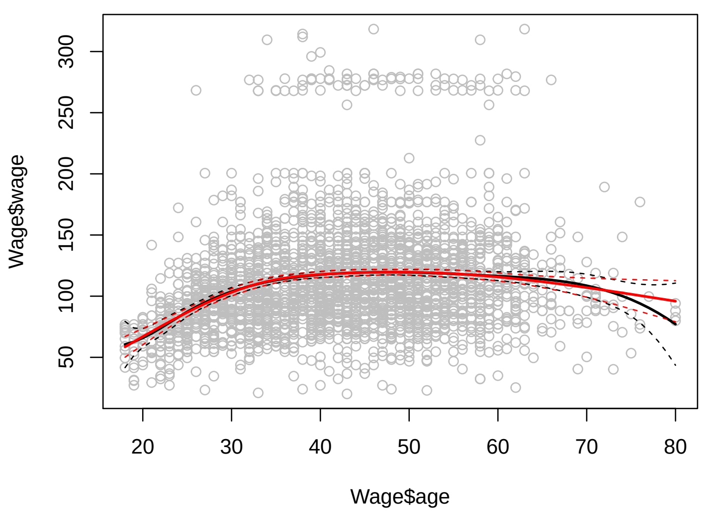
Slika 18: Napovedi za `wage` na podlagi `mod.poly1` s 95 % intervali zaupanja za povprečno napoved



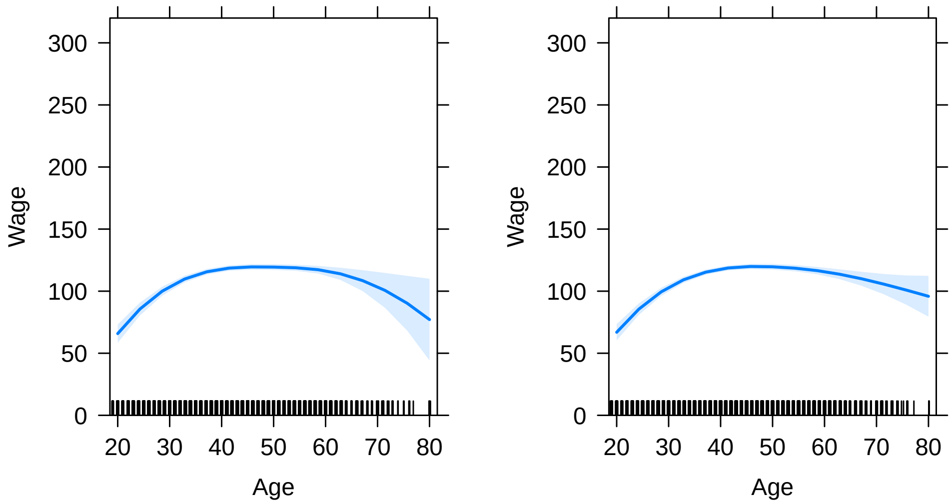
Slika 19: Napovedi za `wage` na podlagi `mod.poly1` s 95 % intervali zaupanja za povprečno napoved

```
> mod.1<- lm(wage~age ,data=Wage)
> mod.2<- lm(wage~poly(age ,2) ,data=Wage)
> mod.3<- lm(wage~poly(age ,3) ,data=Wage)
> mod.4<- lm(wage~poly(age ,4) ,data=Wage)
> mod.5<- lm(wage~poly(age ,5) ,data=Wage)
```

```
> anova(mod.1, mod.2, mod.3, mod.4, mod.5)

Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> library (splines )
> mod.bs<- lm(wage~bs(age, knots =c(25, 40, 60) ), data=Wage)
> napovedi.bs <- predict(mod.bs, newdata =list(age=age.grid), se=T)
> mod.ns<- lm(wage~ns(age, df = 4), data=Wage)
> napovedi.ns<-predict(mod.ns, newdata =list(age=age.grid), se=T)
```

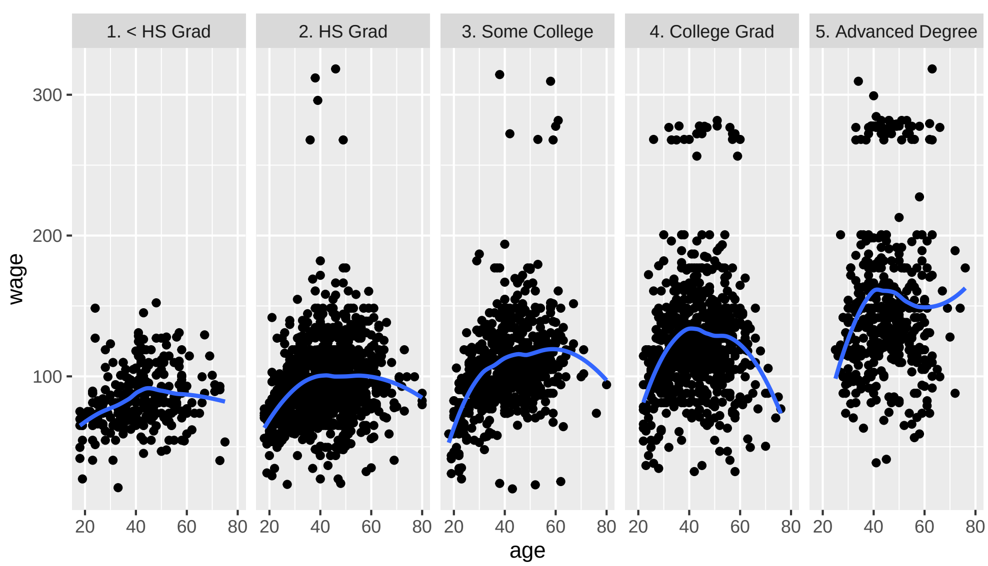


Slika 20: Napovedi za `wage` na podlagi `mod.bs` (črna) in `mod.ns` (rdeča) s 95 % intervali zaupanja za povprečno napoved



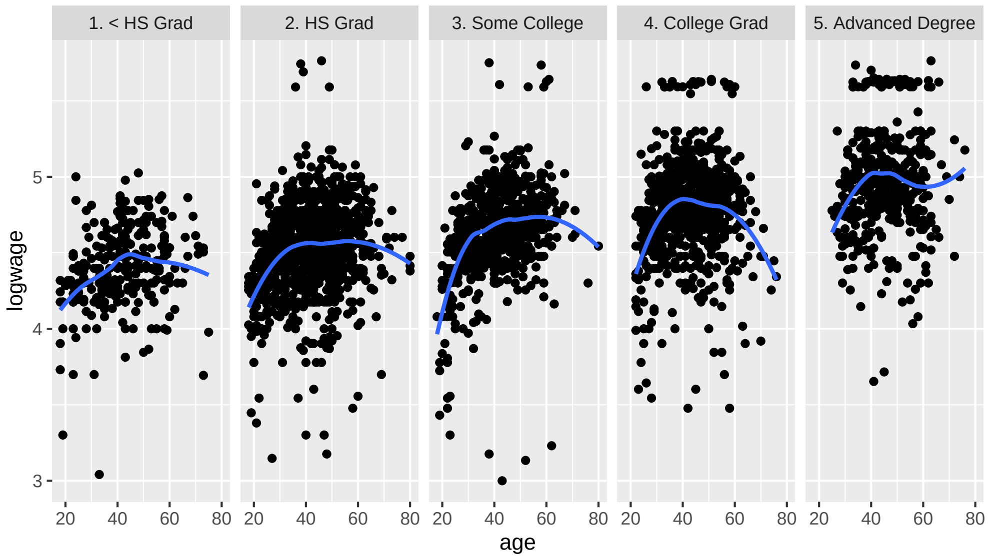
Slika 21: Napovedi za `wage` na podlagi `mod.bs` (levo) in `mod.ns` (desno) s 95 % intervali zaupanja za povprečno napoved

V podatkovnem okviru `Wage` so poleg podatkov o plači (`wage`) in starosti (`age`), tudi podatki o letu pridobljenih podatkov (`year`) in o izobrazbi delavcev (`education`).

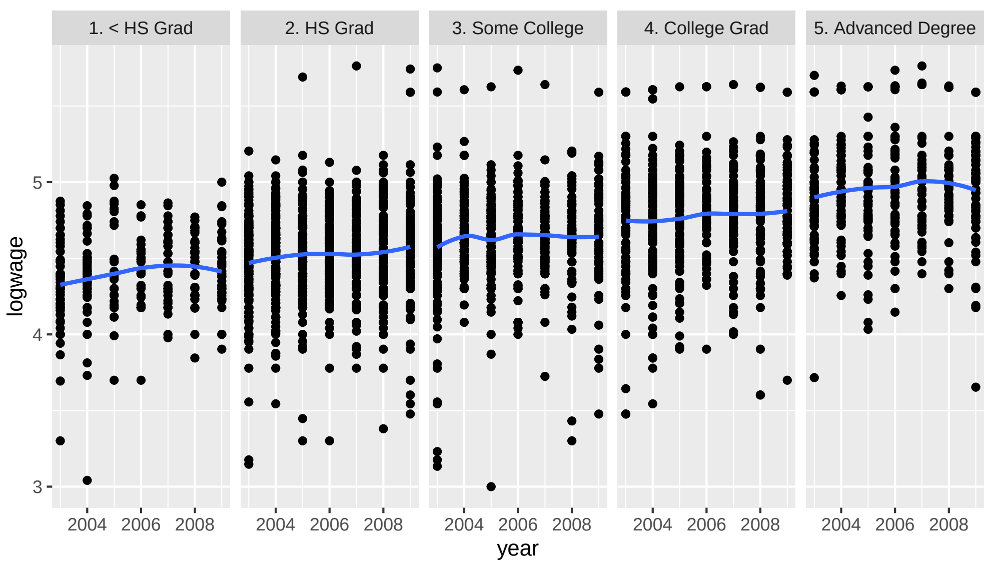


Slika 22: `wage` v odvisnosti od `age` za različne vrednosti spremenljivke `education`

Slika 22 kaže, da se variabilnost plač povečuje z višjo izobrazbo, prav tako se z izobrazbo poveča višina plače, zato je smiselno spremenljivko `wage` logaritmirati. V podatkovnem okviru `Wage` je že logaritmirana spremenljivka `logwage`. Slika 23 kaže, da je variabilnost `logwage` pri različnih izobrazbah veliko bolj podobna kot variabilnost wage.

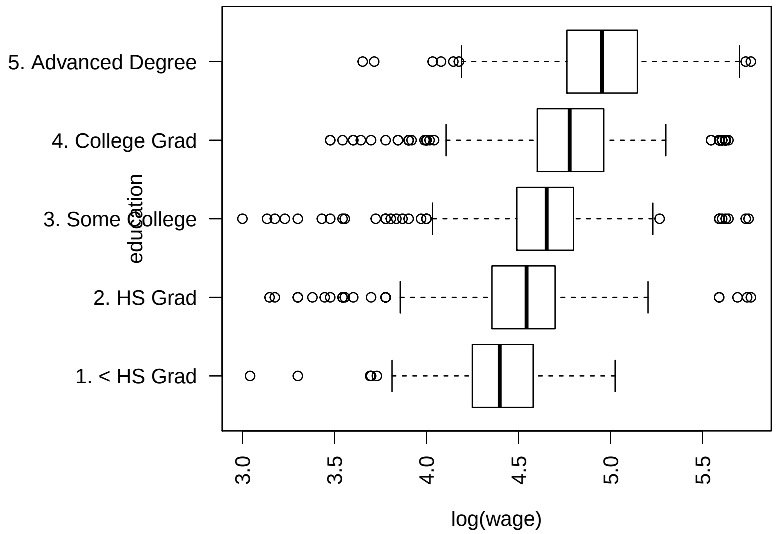


Slika 23: logwage v odvisnosti od age za različne vrednosti spremenljivke education



Slika 24: logwage v odvisnosti od age za različne vrednosti spremenljivke education

```
> par(mar=c(4,9,4,4))
> boxplot(logwage~education, data=Wage, las=2, xlab="log(wage)", horizontal = T)
```



Slika 25: Okvir z ročaji za `wage` in `logwage` v odvisnosti od `education`

Analizirali bomo `logwage` v odvisnosti od `age`, `year` in `education`. Odvisnost `logwage` od `age` ni linearna (Slika 23) zato bomo to napovedno spremenljivko transformirali z naravnimi zlepki s tremi vozlišči (`ns(age, df=4)`). Model z naravnim zlepkom za `age` in ostalimi napovednimi spremenljivkami zapišemo takole:

$$y = \beta_0 + \beta_1 age + \beta_2(age - a_1)^3_+ + \beta_3(age - a_2)^3_+ + \beta_4(age - a_3)^3_+ + \beta_5 year + \\ + \beta_6 education2 + \beta_7 education3 + \beta_8 education4 + \beta_9 education5 + \varepsilon, \quad (17)$$

Funkcija `ns(age, df=4)` na podlagi spremenljivke `age` generira štiri regresorje tako, da se zlepki gladko stikajo v treh vozliščih, na prvem in četrtem odseku je zlepek linearen, na drugem in tretjem odseku pa kubični:

```
> str(ns(Wage$age, df=4))

 'ns' num [1:3000, 1:4] 0 0.0173 0.7511 0.7802 0.5293 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:4] "1" "2" "3" "4"
 - attr(*, "degree")= int 3
 - attr(*, "knots")= Named num [1:3] 33.8 42 51
  ..- attr(*, "names")= chr [1:3] "25%" "50%" "75%"
 - attr(*, "Boundary.knots")= int [1:2] 18 80
 - attr(*, "intercept")= logi FALSE
```

```
> head(ns(Wage$age, df=4))

              1           2          3           4
[1,] 0.00000000  0.00000000 0.00000000  0.00000000
[2,] 0.01731602 -0.13795411 0.31872157 -0.18076746
[3,] 0.75108560  0.16605047 0.09131609 -0.05061290
[4,] 0.78017192  0.07222489 0.11002552 -0.06235889
[5,] 0.52933205  0.38879374 0.13708340 -0.05540437
[6,] 0.34484721  0.49710028 0.19449432 -0.03644180

> mod.place.1<-lm(logwage ~ ns(age, df=4) + year + education, data=Wage)
> vif(mod.place.1)

                    GVIF Df GVIF^(1/(2*Df))
ns(age, df = 4) 1.037196  4        1.004576
year            1.006510  1        1.003250
education       1.032480  4        1.004003
```

V `mod.place.1` nimamo težav z heteroskedatičnostjo, prav tako ni težav s kolinearnostjo, saj imamo samo dve številski spremenljivki. $GVIF$ smo izračunali zato, da pokažemo, kako se izračuna v primeru prisotnosti naravnih zlepkov in opisne napovedne spremenljivke v modelu. V modelu je ocenjenih 10 parametrov.

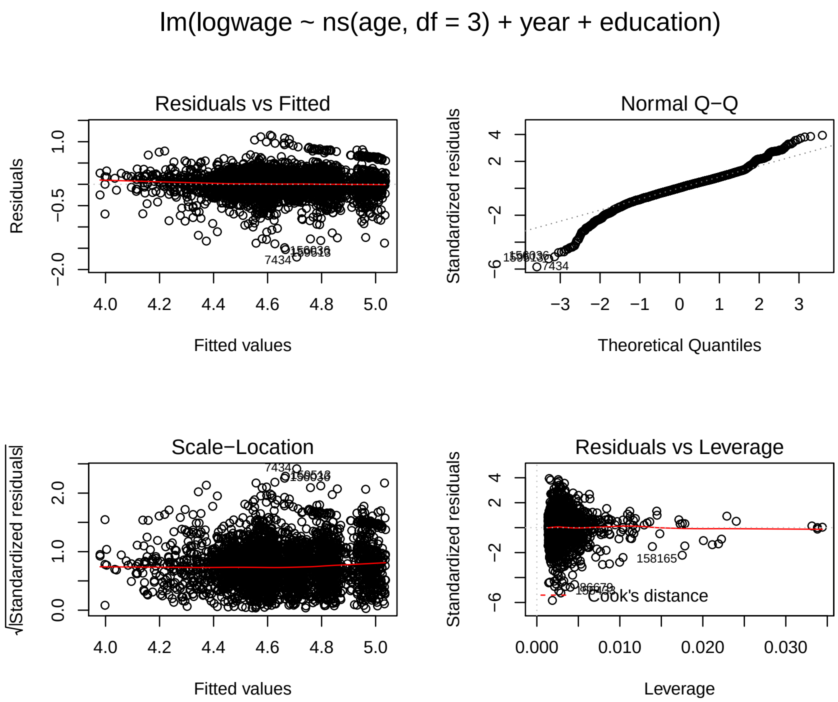
```
> summary(mod.place.1)

Call:
lm(formula = logwage ~ ns(age, df = 4) + year + education, data = Wage)

Residuals:
     Min       1Q   Median       3Q      Max
-1.69854 -0.15676  0.01237  0.16729  1.15718

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 -19.972158   5.311733  -3.760 0.000173 ***
ns(age, df = 4)1              0.500308   0.031033  16.122  < 2e-16 ***
ns(age, df = 4)2              0.326375   0.032321  10.098  < 2e-16 ***
ns(age, df = 4)3              0.800142   0.076798  10.419  < 2e-16 ***
ns(age, df = 4)4              0.093531   0.063642   1.470 0.141762
year                          0.011940   0.002648   4.509 6.78e-06 ***
education2. HS Grad           0.115149   0.020228   5.693 1.37e-08 ***
education3. Some College      0.234262   0.021311  10.993  < 2e-16 ***
education4. College Grad      0.348731   0.021183  16.463  < 2e-16 ***
education5. Advanced Degree   0.514490   0.022994  22.375  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2929 on 2990 degrees of freedom
Multiple R-squared:  0.3087,	Adjusted R-squared:  0.3066
F-statistic: 148.3 on 9 and 2990 DF,  p-value: < 2.2e-16
```

Glede ustreznega števila vozlišč pri modeliranju z zlepki naredimo še model z dvema vozliščema in z štirimi vozlišči ter ju primerjajmo z `mod.place.1` (tri vozlišča).

```
> mod.place.2<-lm(logwage~ns(age, df=3)+year+education, data=Wage)
> anova(mod.place.2,mod.place.1)

Analysis of Variance Table

Model 1: logwage ~ ns(age, df = 3) + year + education
Model 2: logwage ~ ns(age, df = 4) + year + education
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   2991 256.64
2   2990 256.52  1   0.11979 1.3962 0.2375

> mod.place.4<-lm(logwage~ns(age, df=5)+year+education, data=Wage)
> anova(mod.place.1,mod.place.4)

Analysis of Variance Table

Model 1: logwage ~ ns(age, df = 4) + year + education
Model 2: logwage ~ ns(age, df = 5) + year + education
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   2990 256.52
2   2989 256.42  1   0.10204 1.1895 0.2755
```

Primerjava modelov na podlagi F-testa pokaže, da nadaljujemo z analizo `mod.place.2`. Testirajmo še vpliv spremenljivke `education`:

```
> mod.place.2a<-lm(logwage~ns(age, df=4)+year, data=Wage)
> anova(mod.place.2a,mod.place.2)

Analysis of Variance Table

Model 1: logwage ~ ns(age, df = 4) + year
Model 2: logwage ~ ns(age, df = 3) + year + education
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   2994 325.74
2   2991 256.64  3    69.097 268.43 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Slika 26: Ostanki za `mod.place.2`

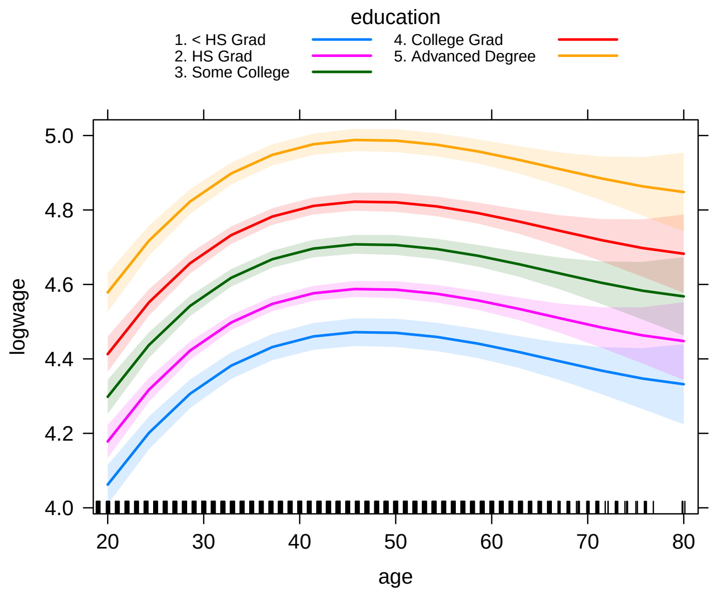
```
> outlierTest(mod.place.2)

        rstudent unadjusted p-value Bonferroni p
7434   -5.868386         4.8832e-09   0.00001465
159513 -5.261429         1.5303e-07   0.00045910
156036 -5.104481         3.5247e-07   0.00105740
155433 -4.794820         1.7078e-06   0.00512350
161447 -4.737063         2.2695e-06   0.00680840
452906 -4.734653         2.2964e-06   0.00688920
86679  -4.572309         5.0196e-06   0.01505900
160269 -4.525402         6.2632e-06   0.01879000
228764 -4.437969         9.4095e-06   0.02822900
160130 -4.395531         1.1435e-05   0.03430500
```

Slika ostankov (Slika 26) kaže, da imamo v modelu deset regresijskih osamelcev (delavcev, za katere model napove previsoko vrednost `logwage`), ki pa ne predstavljajo vplivnih točk.

Z `mod.place.1` je pojasnjene 30.9 % variabilnosti `logwage`. Pri vseh stopnjah izobrazbe se do srednjih let (med 30 in 40 let) `logwage` povečuje, potem začne počasi padati. Spremenljivka `year`, ki pove, katerega leta so bili podatki pridobljeni, ima pozitiven vpliv, s časom se je v povprečju spremenljivka `wage` vsako leto povečala za 1.2 %.

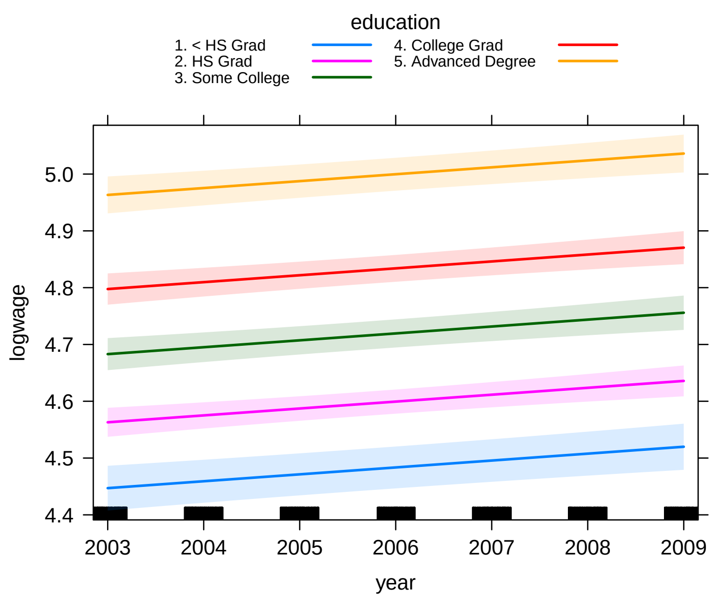
Na sliki 29 so napovedi `mod.place.2` s pripadajočimi intervali zaupanja, ki so široki pri velikih starostih, kjer imamo tudi relativno malo podatkov.

```
> plot(Effect(c("age","education"), mod.place.2), multiline=T, ci.style="bands", main="")
```



Slika 27: Napovedi za `mod.place.2`

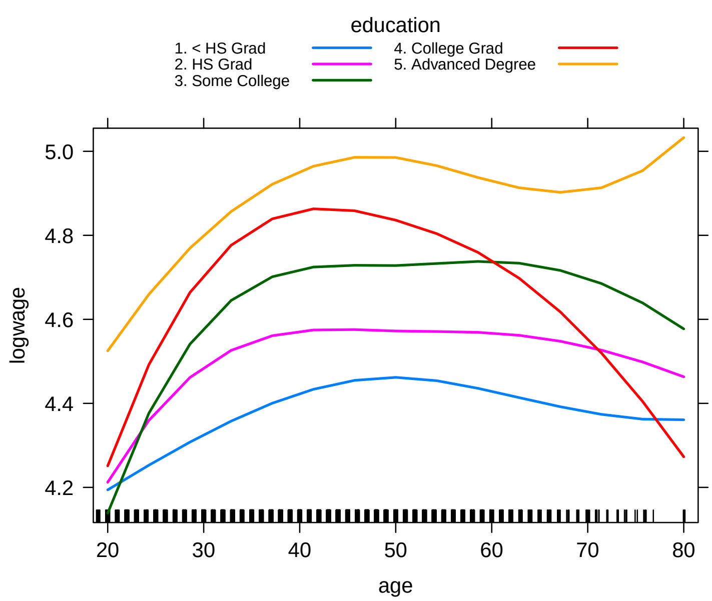
```
> plot(Effect(c("year","education"), mod.place.2), multiline=T, ci.style="bands", main="")
```



Slika 28: Napovedi za `mod.place.2`

Dodatek: po vsebinskem premisleku je smiselno, da se plača različno izobraženih s starostjo spreminja drugače, kar pomeni, da je smiselno v model vključiti tudi interakcijo med zlepkom `age` in `education`.

```
> mod.place.2.int<-lm(logwage~ns(age, df=4)*education+year, data=Wage)
> anova(mod.place.2, mod.place.2.int)

Analysis of Variance Table

Model 1: logwage ~ ns(age, df = 3) + year + education
Model 2: logwage ~ ns(age, df = 4) * education + year
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   2991 256.64
2   2974 252.46 17    4.1871 2.9015 5.937e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(mod.place.2.int)$r.squared

[1] 0.3196432

> plot(Effect(c("age","education"), mod.place.2.int), multiline=T,  main="")
```



Slika 29: Napovedi za `mod.place.2.int`, ki vključuje interakcijo med starostjo in izobrazbo

Naredite diagnostiko modela z interakcijo in ga obrazložite.

2.3 Pljučna kapaciteta, nadaljevanje

V podatkovnem okviru `lungcap` iz paketa `GLMsData` so podatki o pljučni kapaciteti (litri), starosti (dopolnjena leta), telesni višini (inče), spolu in kajenju za vzorec mladostnikov v Bostonu sredi sedemdesetih let (Kahn in Michael, 2005).

- Naredite statistični povzetek za vse spremenljivke v naboru podatkov in ga na kratko obrazložite. Podatke za telesno višino preračunajte v cm.
- Grafično prikažite odvisnost pljučne kapacitete od ostalih spremenljivk v podatkovnem okviru. Grafikone na kratko obrazložite.
- Analizirajte odvisnost pljučne kapacitete od starosti, telesne višine, spola in kajenja. Ali je v model smiselno vključiti kakšno interakcijo? Zakaj? Uporabite grafični prikaz, ki podpira vaš odgovor glede interakcije. Opišite postopek modeliranja.
- Za izbrani model predstavite diagnostiko (ostanki, posebne točke, VIF) in obrazložite ocene parametrov modela ter koeficient determinacije.
- Grafično prikažite napovedi modela s 95 % intervali zaupanja za povprečno napoved.
- Na izbranem modelu uporabite ponovno vzorčenje primerov (bootstrap) in primerjajte bootstrap intervale zaupanja za parametre modela s tistimi iz izbranega modela. Uporabite funkcijo `Boot` iz paketa `car`.