Kazalo

1 IZBIRA MODELA

1.1 Kakovost napovedi

Pri izbiri regresijskega modela se pogosto soočamo z dilemo med kompleksnostjo modela in med njegovim prileganjem podatkom. Enkrat iščemo model, ki kar se da dobro obrazloži vpliv napovednih spremenljivk na odzivno spremenljivko. Drugič pa so nam pomembne predvsem modelske napovedi. V takih primerih se o sprejemljivosti modela odločamo na podlagi analize njegovih napovedi. Pravimo, da analiziramo **kakovost napovedi modela** (*model predictive performance*).

Za predstavitev analize kakovosti napovedi poglejmo model enostavne linearne regresije:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon \sim iid\ N(0, \sigma^2). \tag{1}$$

Na podlagi podatkov $x_1, ..., x_n$ in $y_1, ..., y_n$ po metodi najmanjših kvadratov izračunamo b_0 in b_1, oceni za β_0 in β_1. Pri vrednosti napovedne spremenljivke x_* je tako napoved $y_* = \beta_0 + \beta_1 x_*$, ocena napovedi je $\hat{y}_* = b_0 + b_1 x_*$. Zanima nas **pričakovana vrednost kvadrata napake napovedi** (*expected squared prediction errors*):

$$E[(y_* - \hat{y}_*)^2]. \tag{2}$$

Pričakovano vrednost za (2) zapišimo še drugače, pri tem uporabimo zvezo, ki velja za

varianco slučajne spremenljivke Z, $Var(Z) = E(Z^2) - E(Z)^2$, kar pomeni, da je $E(Z^2) = Var(Z) + E(Z)^2$:

$$\begin{aligned} E[(y_* - \hat{y}_*)^2] &= Var[y_* - \hat{y}_*] + E[y_* - \hat{y}_*]^2 \\ &= Var(y_*) + Var[\hat{y}_*] + E[y_* - \hat{y}_*]^2 \\ &= \sigma^2 + Var[\hat{y}_*] + E[y_* - \hat{y}_*]^2. \end{aligned} \tag{3}$$

V (3) je prvi člen varianca napak σ^2 in je s strani primerjave napovedi različnih modelov med seboj konstanta. Drugi člen $Var[\hat{y}_*]$ predstavlja varianco napovedi modela, tretji člen pa kvadrat pristranosti napovedi $E[y_* - \hat{y}_*]^2$ (*square of prediction bias*).

Za boljše razumevanje teh dveh členov primerjajmo model enostavne regresije (1) z ničelnim modelom, ki vsebuje samo presečišče:

$$y_i = \beta_0 + \varepsilon_i. \tag{4}$$

Za oceno β_0 v (4) minimiramo izraz $\sum_{i=1}^{n}(y_i - \beta_0)^2$ in po metodi najmanjših kvadratov dobimo $b_0 = \bar{y}$. Posledično je napoved $\hat{y}_* = \bar{y}$, njena varianca pa

$$Var(\hat{y}_*) = \frac{\sigma^2}{n}. \tag{5}$$

Za model enostavne regresije (1) je varianca napovedi praviloma večja:

$$Var(\hat{y}_*) = \sigma^2 \left(\frac{1}{n} + \frac{(x_* - \bar{x})^2}{S_{xx}} \right). \tag{6}$$

V splošnem velja, da se z večanjem števila ocenjenih parametrov v modelu varianca napovedi veča.

Tretji člen v (3) predstavlja kvadrat pristranosti napovedi, glede tega imamo dve situaciji:

- če je pravi model enostavne regresije, za napoved na osnovi enostavne regresije velja $E[(y_* - \hat{y}_*)^2] = 0$. Če privzamemo napačen ničelni model, je člen pristranosti napovedi lahko različen od nič;
- če je pravi ničelni model, je člen pristranosti enak 0 za oba modela.

Iz tega sledi, da je člen pristranosti napovedi za kompleksnejši model vedno manjši ali kvečjemu enak kot za model z manj ocenjenimi parametri. Kot smo videli, je z varianco napovedi ravno obratno, za kompleksnejše modele je večja.

Pri dodajanju nove napovedne spremenljivke v model mora biti povečanje variance napovedi na nek način uravnoteženo z zmanjšanjem pristranosti napovedi (*trade off between contributions of bias and variance to prediction error*). Pri izbiri napovednih spremenljivk v model iščemo kompromis med velikostjo variance napovedi in njeno nepristranostjo.

1.2 Ocenjevanje kakovosti napovedi osnovano na podatkih

1.2.1 PRESS statistika

Prvi korak v analizi kakovosti modela je analiza ostankov $e_i = y_i - \hat{y}_i$ in vsote njihovih kvadratov $SS_{residuals}$. Vemo, da se $SS_{residuals}$ zmanjša ob vsaki dodani napovedni spremenljivki, tudi če ta nima vpliva na odzivno spremenljivko. Pri ocenjevanju kakovosti napovedovanja z izbranim modelom za točko, ki ni v vzorcu, se lahko zgodi, da je z ostanki ocenjena napaka napovedi podcenjena. Boljšo mero za oceno napake napovedi za posamezno točko dobimo s **PRESS ostanki**:

$$e_{i,-i} = y_i - \hat{y}_{i,-i}. \tag{7}$$

V (7) je $\hat{y}_{i,-i}$ napoved za y_i na podlagi modela, ki je narejen na vseh podatkih brez i-te točke.

Glede na definicijo PRESS ostankov jih izračunamo tako, da prilagodimo za vsako točko en model, torej n modelov. Pokaže se, da to ni potrebno. Izračunamo jih lahko na podlagi vzvodov h_{ii}, $i = 1, ..., n$.

$$e_{i,-i} = \frac{e_i}{1 - h_{ii}}. \tag{8}$$

Ker za ostanke e_i velja, da so normalno porazdeljeni in ker so vzvodi odvisni samo od modelske matrike, so tudi PRESS ostanki porazdeljeni normalno. Njihova varianca je

$$Var(e_{i,-i}) = \frac{Var(e_i)}{(1 - h_{ii})^2} = \frac{\sigma^2(1 - h_{ii})}{(1 - h_{ii})^2} = \frac{\sigma^2}{1 - h_{ii}}.$$

To pomeni, da njihova varianca ni konstantna.

PRESS ostanki predstavljajo povečane navadne ostanke modela, to povečanje je odvisno od tega, kako vplivna je posamezna točka v procesu ocenjevanja parametrov modela.

Mera za prileganje modela je vsota kvadratov PRESS ostankov, t. i. $PRESS$**-statistika** (*Predictive Residual Error Sum of Squares*):

$$PRESS = \sum_{i=1}^{n} e_{i,-i}^2. \tag{9}$$

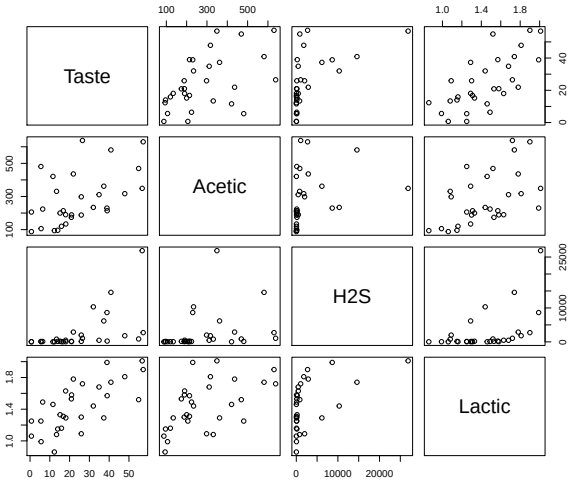
$PRESS$-statistika v nasprotju z $SS_{residuals}$ ne pada nujno z dodajanjem parametrov v model. Na podlagi $PRESS$-statistike lahko primerjamo različne modele narejene za isto odzivno spremenljivko. Model z najmanjšo vrednostjo $PRESS$-statistike je najboljši v smislu kakovosti napovedi.

Primer: `cheese`

V podatkovnem okviru `cheese` v paketu `GLMsData` so podatki iz študije o siru čedar v dolini

La Trobe v Victorii v Avstraliji. Subjektivno so ocenjevali okus sira (`Taste`), izmerili pa so koncentracijo ocetne kisline, koncentracijo žveplovodika in koncentracijo mlečne kisline (`Lactic`). Zanimalo jih je, kako je okus odvisen od teh spremenljivk v smislu najboljše možne napovedi za `Taste`.

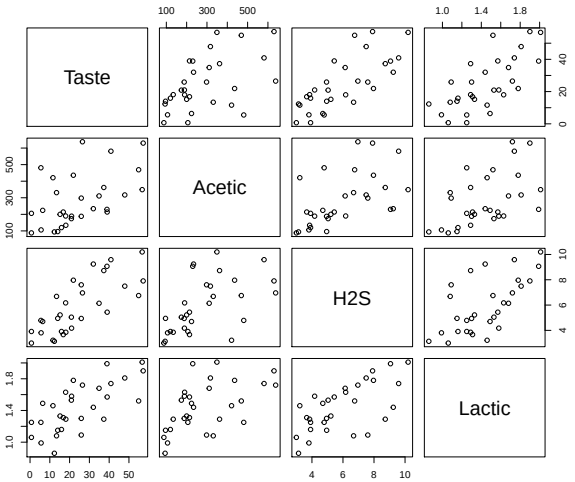
```
> library(GLMsData)
> data(cheese)
> str(cheese)

'data.frame':        30 obs. of  4 variables:
 $ Taste : num  12.3 20.9 39 47.9 5.6 25.9 37.3 21.9 18.1 21 ...
 $ Acetic: int  94 174 214 317 106 298 362 436 134 189 ...
 $ H2S   : int  23 155 230 1801 45 2000 6161 2881 47 65 ...
 $ Lactic: num  0.86 1.53 1.57 1.81 0.99 1.09 1.29 1.78 1.29 1.58 ...


> pairs(cheese)
```



Slika 1: Matrika razsevnih grafikonov za podatkovni okvir `cheese`

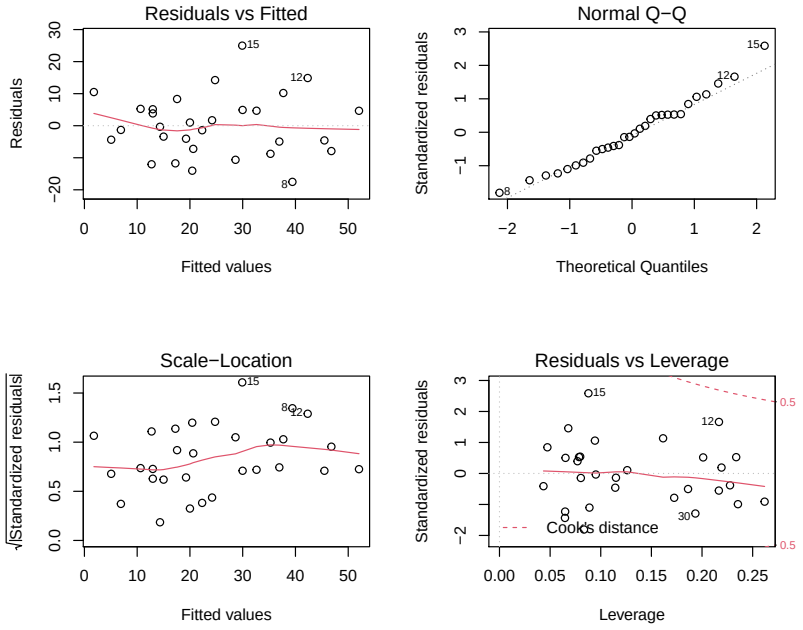
Slika 1 nakazuje, da bi bilo potrebno podatke spremenljivke H2S logaritmirati, ker je večina vrednosti zgoščena blizu 0, poleg teh pa imamo še nekaj bistveno višjih vrednosti. Slika 2 kaže, da z logaritmiranjem dokaj enakomerno pokrijemo regresorski prostor.

```
> cheese$H2S <- log(cheese$H2S)
> # vrednosti spremenljivke kar prekrijemo z logaritmiranimi vrednostmi
> pairs(cheese)
```



Slika 2: Matrika razsevnih grafikonov za podatkovni okvir `cheese` z logaritmiranimi vrednostmi H2S

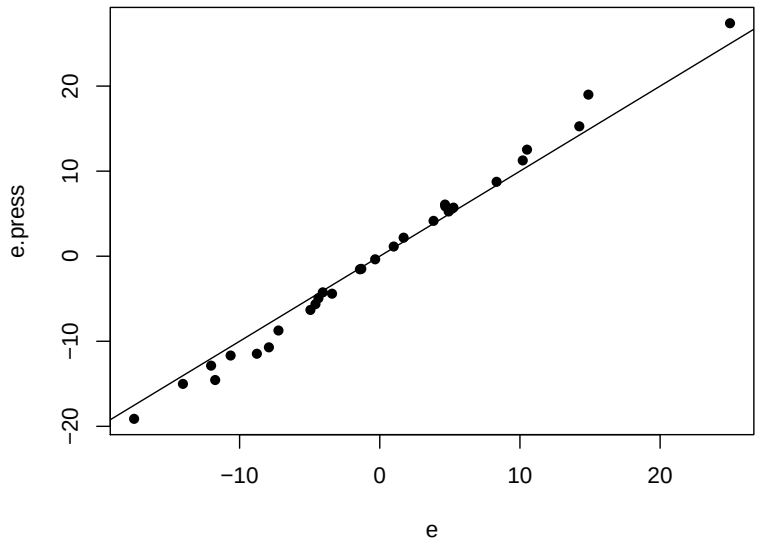
```
> mod.cheese<-lm(Taste ~ Acetic + H2S + Lactic, data=cheese)
> summary(mod.cheese)

Call:
lm(formula = Taste ~ Acetic + H2S + Lactic, data = cheese)

Residuals:
     Min       1Q   Median       3Q      Max
-17.5250  -6.6580  -0.8226   5.0833  24.9859

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.142493   9.277924  -2.925  0.00705 **
Acetic        0.004184   0.014916   0.281  0.78129
H2S           3.836799   1.219895   3.145  0.00413 **
Lactic       19.201965   8.457616   2.270  0.03171 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.12 on 26 degrees of freedom
Multiple R-squared:  0.6528,        Adjusted R-squared:  0.6127
F-statistic: 16.29 on 3 and 26 DF,  p-value: 3.675e-06
```



Slika 3: Ostanki za `mod.cheese`

Primerjajmo ostanke in PRESS ostanke za `mod.cheese`. PRESS ostanke bomo najprej izračunali na podlagi njihove definicije, kot razliko $y_i - \hat{y}_{i,-i}$:

```
> e<-residuals(mod.cheese)
> e.press<-numeric()
> for (i in 1:length(e)){
+   mod<-lm(Taste ~ Acetic + H2S + Lactic, data=cheese[-i,])
+   novi<-cheese[i,]
+   e.press[i]<-cheese[i,"Taste"] - predict(mod, newdata=novi)
+ }
> sum(e.press^2)

[1] 3456.207
```



Slika 4: Ostanki in PRESS ostanki za `mod.cheese`

Na Sliki 4 vidimo, da so PRESS ostanki v absolutnem smislu večji kot navadni ostanki, kar smo tudi pričakovali v skladu z njihovo odvisnostjo od vzvodov.

Izračunajmo PRESS ostanke in $PRESS$-statistiko za `mod.cheese` še na podlagi vzvodov:

```
> h<-hatvalues(mod.cheese)
> press.ost<-residuals(mod.cheese)/(1-h)
> PRESS<-sum(press.ost^2)
> PRESS

[1] 3456.207
```

Izračunajmo $PRESS$-statistiko za vse možne modele na podlagi treh napovednih spremenljivk (brez interakcij):

```
> PRESS<-numeric()
> nap.sprem <- names(cheese)
> nap.sprem <- nap.sprem[! nap.sprem %in% "Taste"]
> k <- length(nap.sprem)
> # id predstavlja vse možne kombinacije treh spremenljivk
> id <- unlist(lapply(1:k,function(i) combn(1:k,i,simplify=FALSE)),
+              recursive=FALSE)
```

```
> formule <- sapply(id, function(i) paste("Taste~", paste(nap.sprem[i],
+                                                         collapse="+")))
> formule

[1] "Taste~ Acetic"            "Taste~ H2S"
[3] "Taste~ Lactic"            "Taste~ Acetic+H2S"
[5] "Taste~ Acetic+Lactic"     "Taste~ H2S+Lactic"
[7] "Taste~ Acetic+H2S+Lactic"

> for (i in (1:length(formule))){
+   mod<-lm(formule[i], data=cheese)
+   h<-lm.influence(mod)$hat
+   press.ost<-residuals(mod)/(1-h)
+   PRESS[i]<-sum(press.ost^2)
+  }
> data.frame(formule, PRESS)

                   formule    PRESS
1            Taste~ Acetic 6547.165
2               Taste~ H2S 3687.930
3            Taste~ Lactic 4375.643
4        Taste~ Acetic+H2S 3952.583
5     Taste~ Acetic+Lactic 4564.764
6        Taste~ H2S+Lactic 3135.388
7 Taste~ Acetic+H2S+Lactic 3456.207
```

Na podlagi $PRESS$-statistike izberemo model, ki kot napovedni spremenljivke vsebuje `Lactic` in `H2S`.

1.2.2 Navzkrižno preverjanje

Izbira ustreznega modela na podlagi $PRESS$-statistike predstavlja najbolj osnovni način **navzkrižnega preverjanja** (*cross validation*), ki ga pogosto imenujemo tudi **navzkrižno preverjanje brez ene enote** (*leave one out cross validation*).

V splošnem pri **osnovnem navzkrižnem preverjanju** razdelimo enote iz vzorca na dva dela: **učni vzorec** (I_{ucni}), ki ga uporabimo za oceno parametrov modela (*training sample*) in **testni vzorec** (I_{test}), ki ga uporabimo za ugotavljanje kakovosti napovedi modela (*validation sample*). Izbira enot v posamezni vzorec mora biti slučajna.

Postopek osnovnega načina navzkrižnega preverjanja bomo predstavili na primeru `cheese`. Najprej enote razdelimo v učni in v testni podvzorec. To lahko naredimo na različne načine. Za primer izberimo enako velika podvzorca.

```
> n<-dim(cheese)[1]
> n.u<-n.t<-n/2
> # naredimo vektor z vrednostmi TRUE in FALSE, vsaka vrednost po 15 krat
> (ind<-rep(c(TRUE, FALSE), each=n.u))

 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE

> # slučajno razporedimo vrednosti
> set.seed(12345) # zaradi ponovljivosti
> (ind<-sample(ind))

 [1]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
[13]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[25]  TRUE FALSE FALSE  TRUE FALSE FALSE

> cheese.ucni<-cheese[ind,]
> cheese.test<-cheese[!ind,]
```

Parametre modela $\mathbf{b}_{ucni}$ ocenimo na podlagi učnega vzorca I_{ucni}, napovedi izračunamo za enote v testnem vzorcu I_{testni}.

```
> # izračun napovedi samo za en izbrani model
> mod.ucni<-lm(Taste~H2S+Lactic, data=cheese.ucni)
> y.nap<-predict(mod.ucni, cheese.test)
```

Vsota kvadratov napak napovedi na testnem vzorcu predstavlja t. i. **kriterij navzkrižnega preverjanja** CVC:

$$CVC = \sum_{i \in I_{test}} (y_i - \hat{y}_i)^2 = \sum_{i \in I_{test}} e_i^2 \tag{10}$$

Na podlagi CVC izračunamo t. i. **povprečno napako napovedi** oziroma **koren povprečja kvadratov napak napovedi** ($RMSE$, *root mean square prediction error*):

$$RMSE = \sqrt{\frac{1}{n_{test}} \sum_{i \in I_{test}} e_i^2} = \sqrt{\frac{CVC}{n_{test}}}. \tag{11}$$

```
> (CVC<-sum((cheese.test$Taste-y.nap)^2))

[1] 1686.761

> (RMSE<-sqrt(CVC/n.t))

[1] 10.60428
```

Postopek navzkrižnega preverjanja ponovimo za vse kandidate za model. Izberemo model z najmanjšo vrednostjo CVC oziroma $RMSE$, nato za izbrani model izračunamo ocene parametrov na podlagi vseh podatkov.

Osnovni način navzkrižnega preverjanja se izkaže kot neprimerna metoda, če osnovni vzorec ni dovolj velik. Druga težava je v tem, da razdelitev na vzorce vpliva na izid. Za ilustracijo tega vpliva naredimo pet različnih razporeditev enot v učni in testni vzorec na podlagi podatkovnega okvira `cheese` in izračunajmo CVC in $RMSE$:

```
> tabela<-data.frame(formule)
> for (j in 1:5) {
+   izbor<-rep(c(TRUE, FALSE), each=n.u)
+   set.seed(j*10)
+   izbor<-sample(izbor)
+   cheese.ucni<-cheese[izbor,]
+   cheese.test<-cheese[!izbor,]
+   CVC<-numeric()
+   for (i in (1:length(formule))){
+     mod<-lm(formule[i], data=cheese.ucni)
+     y.nap<-predict(mod, cheese.test)
+     CVC[i]<-sum((cheese.test$Taste-y.nap)^2)
+   }
+   # za primerjavo v nadaljevanju izračunamo tudi RMSE
+  tabela<-data.frame(tabela, round(CVC, 1), round(sqrt(CVC/15),1))
+ }
> names(tabela)<-c("formula", "CVC1", "RMSE1","CVC2", "RMSE2", "CVC3",
+                  "RMSE3", "CVC4", "RMSE4", "CVC5", "RMSE5")
> tabela[, c(1:2,4,6,8,10)]

                   formula   CVC1   CVC2   CVC3   CVC4   CVC5
1            Taste~ Acetic 4164.4 3263.5 2679.4 2832.3 3313.9
2               Taste~ H2S 2589.2 1263.7 2404.7 1522.4 1483.1
3            Taste~ Lactic 2618.3 1803.7 2394.7 2228.2 1290.4
4        Taste~ Acetic+H2S 2531.4 1668.2 2460.4 1451.1 1569.8
5     Taste~ Acetic+Lactic 2617.4 1972.1 2499.3 2105.6 1952.9
6        Taste~ H2S+Lactic 2113.5  998.5 2027.0 2040.8  946.7
7 Taste~ Acetic+H2S+Lactic 2110.8 1319.5 2368.0 2116.8 1181.5
```

V zgornji tabeli vidimo, da je izbira ustreznega modela na podlagi osnovnega navzkrižnega preverjanja odvisna od slučajne izbire enot v učni in testni vzorec, najmanjšo vrednost CVC imajo pri različnih delitvah v učni in testni podvzorec različni modeli.

Pomanjkljivosti osnovnega navzkrižnega preverjanja so v veliki meri odpravljene pri t. i. **K-kratnem navzkrižnem preverjanju** (*K-fold cross validation*). V tem primeru na začetku enote razdelimo v K enako velikih vzorcev. Parametre modela ocenjujemo K-krat. Vsakič izmed podatkov izločimo en vzorec (testni vzorec), na ostalih podatkih ($K-1$ vzorcev skupaj)

ocenimo parametre modela in nato izračunamo CVC iz napovedi na testnem vzorcu. Na koncu K vrednosti CVC povprečimo za vsak model. Prednost tega načina navzkrižnega preverjanja je v tem, da vsak podatek nastopa $K-1$-krat v učnem vzorcu in natanko enkrat v testnem vzorcu.

Za K-kratno navzkrižno preverjanje lahko uporabimo funkcijo `cvFit` v paketu `cvTools`. Za vsak model funkcija `cvFit` izračuna povprečno napako napovedi $RMSE$ kot povprečje K-tih $RMSE$ izračunanih na testnih vzorcih velikosti n/K.

```
> library(cvTools)
> cv<-numeric()
> for (i in (1:length(formule))){
+   mod<-lm(formule[i], data=cheese)
+   mod.cv<-cvFit(mod, data=cheese, y=cheese$Taste, K=5, seed=7)
+   cv[i]<-mod.cv$cv
+   }
> data.frame(formule, cv=round(cv,1))

                  formule   cv
1            Taste~ Acetic 14.6
2               Taste~ H2S 11.0
3            Taste~ Lactic 11.8
4        Taste~ Acetic+H2S 10.9
5     Taste~ Acetic+Lactic 11.7
6        Taste~ H2S+Lactic  9.7
7 Taste~ Acetic+H2S+Lactic  9.9
```

Za primerjavo so v spodnji tabeli $RMSE$ za pet primerov osnovnega načina navzkrižnega preverjanja za model odvisnosti `Taste` od ostalih spremenljivk v podatkovnem okviru `cheese`.

```
> tabela[, c(1,3,5,7,9,11)]

                  formula RMSE1 RMSE2 RMSE3 RMSE4 RMSE5
1            Taste~ Acetic  16.7  14.8  13.4  13.7  14.9
2               Taste~ H2S  13.1   9.2  12.7  10.1   9.9
3            Taste~ Lactic  13.2  11.0  12.6  12.2   9.3
4        Taste~ Acetic+H2S  13.0  10.5  12.8   9.8  10.2
5     Taste~ Acetic+Lactic  13.2  11.5  12.9  11.8  11.4
6        Taste~ H2S+Lactic  11.9   8.2  11.6  11.7   7.9
7 Taste~ Acetic+H2S+Lactic  11.9   9.4  12.6  11.9   8.9
```

1.3 Asimptotske metode ocenjevanja kakovosti modela

Asimptotske metode ocenjevanja kakovosti modela lahko uporabimo, kadar je n dovolj velik. Podatkov pri tem ne delimo na učni in testni vzorec. Ideja teh metod je, da ocenimo **povprečno kvadratno napako napovedi** ($MSE(\hat{y}(x_i))$) na podlagi vsote kvadratov ostankov modela, velikosti vzorca n in števila regresorjev v modelu p, kjer je $p \leq k$ in je v polnem regresijskem modelu k regresorjev.

1.3.1 Mallow-a C_p-statistika

Mallow-a C_p-statistika je osnovana na ideji, da želimo preprečiti preprileganje napovedi modela in hkrati ne izpustiti pomembnih regresorjev iz končnega modela. To statistiko uporabimo, če so parametri modela ocenjeni po metodi najmanjših kvadratov. Iščemo model za katerega je ocena izraza (12) minimalna.

$$\sum_{i=1}^{n} \frac{MSE(\hat{y}(x_i))}{\sigma^2} = \sum_{i=1}^{n} \frac{E[(y(x_i) - \hat{y}(x_i))^2]}{\sigma^2}. \tag{12}$$

Če pričakovano vrednost kvadrata v števcu (12) razvijemo, kot smo to naredili v (3), $MSE(\hat{y}(x_i))$ izrazimo kot vsoto variance napovedi in kvadrata njene pristranosti:

$$MSE(\hat{y}(x_i)) = Var[\hat{y}(x_i)] + E[y(x_i) - \hat{y}(x_i)]^2, \tag{13}$$

in (12) izrazimo kot vsoto dveh členov:

$$\sum_{i=1}^{n} \frac{MSE(\hat{y}(x_i))}{\sigma^2} = \frac{\sum_{i=1}^{n} Var[\hat{y}(x_i)]}{\sigma^2} + \frac{\sum_{i=1}^{n} E[y(x_i) - \hat{y}(x_i)]^2}{\sigma^2}. \tag{14}$$

Za model s p, $p < k$, regresorji lahko pokažemo, da je p nepristrana cenilka za prvi člen v (14). Nepristrana cenilka za drugi člen v (14) je $(n-p)(\hat{\sigma}_p^2 - \sigma^2)/\sigma^2$:

$$\sum_{i=1}^{n} \frac{MSE(\hat{y}(x_i))}{\sigma^2} = p + \frac{(n-p)(\hat{\sigma}_p^2 - \sigma^2)}{\sigma^2}. \tag{15}$$

V praksi je potrebno σ^2 oceniti. Nepristrana cenilka za σ^2 je $\hat{\sigma}_k^2$ ocenjena na podlagi polnega modela $p = k$ (model, ki vsebuje vse možne regresorje). Ko v (15) vstavimo oceno $\hat{\sigma}_k^2$, dobimo izraz za C_p-statistiko:

$$C_p = p + \frac{(n-p)(\hat{\sigma}_p^2 - \hat{\sigma}_k^2)}{\hat{\sigma}_k^2} = \frac{SS_{p,residual}}{MS_{k,residual}} - n + 2p, \tag{16}$$

$SS_{p,residual}$ je vsota kvadratov ostankov modela s p regresorji in $MS_{k,residual}$ je srednji kvadrat ostankov oziroma ocena variance napak polnega modela s k-regresorji .

Z minimiranjem C_p uravnotežimo prileganje modela (če izpustimo pomemben regresor, bo

imel izraz ($\hat{\sigma}^2 - \hat{\sigma}_k^2$) pozitivno vrednost) in njegovo kompleksnost (število parametrov v modelu p).

Na primeru `cheese` bomo za izračun C_p-statistike uporabili funkcijo `ols.mallows.cp` iz paketa `olsrr`:

```
> library(olsrr)
> polni_model <- lm(Taste~.,data=cheese)
> Cp<-numeric()
> for (i in (1:length(formule))){
+   mod<-lm(formule[i], data=cheese)
+   Cp[i] <-ols_mallows_cp(mod, polni_model)
+ }
> izpis<-data.frame(formule, Cp)
> izpis<-izpis[order(Cp),]; izpis[1:5,]
```

```
                   formule        Cp
6        Taste~ H2S+Lactic  2.078697
7 Taste~ Acetic+H2S+Lactic  4.000000
2               Taste~ H2S  6.108163
4        Taste~ Acetic+H2S  7.154606
3            Taste~ Lactic 11.741046
```

Po Cp kriteriju je najboljši model, ki ima regresorja `H2S` in `Lactic`. Tak rezultat so dale tudi ostale metode za oceno kakovosti napovedi modela. V splošnem ni nujno, da različni kriteriji vrnejo kot najboljši isti model.

1.3.2 Akaike informacijski kriterij AIC

Akaike informacijski kriterij je v praksi najpogosteje uporabljen kriterij za izbiro najboljšega modela. Ni primeren samo za modele, kjer se parametre modela ocenjuje z metodo najmanjših kvaratov, temveč tudi za posplošene linearne modele, kjer se parametre ocenjuje po metodi največjega verjetja. V primeru normalnega linearnega modela obe metodi vrneta iste rezultate.

Akaike informacijski kriterij temelji na statistiki AIC:

$$AIC = -2ln\hat{L} + 2p. \tag{17}$$

V (17) je p število parametrov v modelu in $log\hat{L}$ je logaritem verjetja normalnega modela ovrednoten z ocenami parametrov modela $\mathbf{b}$:

$$log\hat{L} = -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}(\mathbf{y} - \mathbf{Xb})^T(\mathbf{y} - \mathbf{Xb}). \tag{18}$$

Tudi AIC upošteva prileganje in kompleksnost modela. Absolutna vrednost AIC nima

vsebinskega pomena, zanimiva je relativno glede na AIC vrednosti drugih modelov. Najboljši je model z najmanjšo vrednostjo AIC.

Izraz za AIC v okviru normalnih linearnih modelov poenostavimo, če v (18) $\hat{\sigma}^2$ zamenjamo za $SS_{residual}/n$ in $(\mathbf{y}-\mathbf{Xb})^T(\mathbf{y}-\mathbf{Xb})$ z $SS_{residual}$ ter v izrazu izpustimo konstantne člene, ki niso odvisni od prilagajanja modela:

$$AIC = 2p + n\ ln(SS_{residual}). \tag{19}$$

```
> AIC<-numeric()
> for (i in (1:length(formule))){
+   mod<-lm(formule[i], data=cheese)
+   AIC[i]<-AIC(mod)
+   }
> data.frame(formule, AIC=round(AIC,1))
```

```
                    formule   AIC
1            Taste~ Acetic 248.3
2               Taste~ H2S 232.0
3            Taste~ Lactic 236.9
4        Taste~ Acetic+H2S 233.1
5     Taste~ Acetic+Lactic 237.4
6        Taste~ H2S+Lactic 227.8
7 Taste~ Acetic+H2S+Lactic 229.7
```

Tudi na podlagi AIC kriterija izberemo model `Taste~H2S+Lactic`.

1.4 Sekvenčne metode za izbiro najboljšega modela za napovedovanje

V vseh do sedaj predstavljenih metodah izbire najboljšega modela smo računali različne statistike na podlagi katerih smo postavili kriterij izbora za vse kandidate za model. Če je število napovednih spremenljivk veliko, postane množica kandidatov za model zelo velika (2^k) in računanje postane računsko zahtevno. V takih primerih je smiselno uporabiti **sekvenčno metodo izbire najboljšega modela**. V splošnem se uporabljajo trije pristopi: **izbira naprej** (*forward selection*), **izbira nazaj** (*backward selection*) in **izbira po korakih** (*stepwise selection*).

1.4.1 Izbira naprej

Pri metodi izbire naprej začnemo z ničelnim modelom, ki vsebuje samo presečišče. V prvem koraku je to t. i. **veljaven model**. Postopamo po naslednjih korakih:

1. Izračunamo kriterijsko statistiko za veljaven model (AIC, Cp, CVC, prilagojen R^2, ...).

2. V veljaven model dodamo en regresor in ponovno izračunamo kriterijsko statistiko, to naredimo za vse regresorje, ki še niso v modelu.

3. Med modeli iz prejšnje točke poiščemo model z najmanjšo vrednostjo kriterijske statistike. Če je ta vrednost manjša od kriterijske statistike veljavnega modela, postane ta model veljaven in se vrnemo k točki 2.

4. Če nima noben model z dodanim enim regresorjem manjše vrednosti kriterijske statistike, postane veljaven model izbrani model.

Na tak način v procesu izbire naredimo $1 + k(k+1)/2$ modelov, kar je pri velikem številu napovednih spremenljivk v modelu k precej manj kot 2^k (npr. $k = 20$, $1 + k(k+1)/2 = 211$, $2^k = 1048576$). Metoda ne zagotavlja, da je izbrani model res najboljši.

Za sekvenčno metodo izbire naprej bomo uporabili funkcijo `stepAIC` iz paketa `MASS`, ki modele izbira na podlagi najmanjše vrednosti AIC statistike.

```
> library(MASS)
> mod.nul<-lm(Taste~1, data=cheese)
> step<-stepAIC(mod.nul, scope=~H2S+Lactic+Acetic, direction="forward")

Start:  AIC=168.29
Taste ~ 1

         Df Sum of Sq    RSS    AIC
+ H2S     1    4376.9 3286.0 144.89
+ Lactic  1    3800.4 3862.5 149.74
+ Acetic  1    2018.2 5644.7 161.12
<none>                7662.9 168.29

Step:  AIC=144.89
Taste ~ H2S

         Df Sum of Sq    RSS    AIC
+ Lactic  1    617.07 2668.9 140.65
<none>                3286.0 144.89
+ Acetic  1     97.59 3188.4 145.98

Step:  AIC=140.65
Taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2668.9 140.65
+ Acetic  1     8.054 2660.9 142.56
```

Ta metoda vrne za izbrani model `Taste`$\sim$`H2S+Lactic`.

1.4.2 Izbira nazaj

Pri metodi izbire nazaj je v prvem koraku veljaven polni model, ki vsebuje vse regresorje. Postopamo po naslednjih korakih:

1. Izračunamo kriterijsko statistiko za veljaven model (AIC,...).
2. Iz veljavnega modela odstranimo en regresor in ponovno izračunamo kriterijsko statistiko, to naredimo za vse regresorje, ki so v modelu.
3. Med modeli iz prejšnje točke poiščemo model z najmanjšo vrednostjo kriterijske statistike. Če je ta vrednost manjša od kriterijske statistike veljavnega modela, postane ta model veljaven in se vrnemo k točki 2.
4. Če nima noben model z odstranjenim enim regresorjem manjše vrednosti kriterijske statistike, postane veljaven model izbrani model.

```
> mod.polni<-lm(Taste~H2S+Lactic+Acetic, data=cheese)
> step<-stepAIC(mod.polni, direction="backward")

Start:  AIC=142.56
Taste ~ H2S + Lactic + Acetic

         Df Sum of Sq    RSS    AIC
- Acetic  1      8.05 2668.9 140.65
<none>                2660.9 142.56
- Lactic  1    527.53 3188.4 145.98
- H2S     1   1012.39 3673.3 150.23

Step:  AIC=140.65
Taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2668.9 140.65
- Lactic  1    617.07 3286.0 144.89
- H2S     1   1193.55 3862.5 149.74
```

Tudi ta metoda vrne za izbrani model `Taste`$\sim$`H2S+Lactic`.

1.4.3 Izbira po korakih

Pri metodi izbire po korakih začnemo s poljubnim modelom. V prvem koraku je to veljaven model. Postopamo po naslednjih korakih:

1. Izračunamo kriterijsko statistiko za veljaven model (AIC,...).
2. Iz veljavnega modela odstranimo po en regresor in tudi dodamo po en regresor ter za vsak popravljeni model izračunamo kriterijsko statistiko.

3. Med modeli iz prejšnje točke poiščemo model z najmanjšo vrednostjo kriterijske statistike. Če je ta vrednost manjša od kriterijske statistike veljavnega modela, postane ta model veljaven in se vrnemo k točki 2.

4. Če nima noben model z dodanim ali odstranjenim enim regresorjem manjše vrednosti kriterijske statistike, postane veljaven model izbrani model.

Rezultati so odvisni od tega, kateri model izberemo v prvem koraku.

```
> mod.prvi<-lm(Taste~Acetic, data=cheese)
> step<-stepAIC(mod.prvi, scope=~H2S+Lactic+Acetic, direction="both")

Start:  AIC=161.12
Taste ~ Acetic

         Df Sum of Sq    RSS    AIC
+ H2S     1    2456.3 3188.4 145.98
+ Lactic  1    1971.4 3673.3 150.23
<none>                5644.7 161.12
- Acetic  1    2018.2 7662.9 168.29

Step:  AIC=145.98
Taste ~ Acetic + H2S

         Df Sum of Sq    RSS    AIC
+ Lactic  1    527.53 2660.9 142.56
- Acetic  1     97.59 3286.0 144.89
<none>                3188.4 145.98
- H2S     1   2456.27 5644.7 161.12

Step:  AIC=142.56
Taste ~ Acetic + H2S + Lactic

         Df Sum of Sq    RSS    AIC
- Acetic  1      8.05 2668.9 140.65
<none>                2660.9 142.56
- Lactic  1    527.53 3188.4 145.98
- H2S     1   1012.39 3673.3 150.23

Step:  AIC=140.65
Taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2668.9 140.65
+ Acetic  1      8.05 2660.9 142.56
```

```
- Lactic  1    617.07 3286.0 144.89
- H2S     1   1193.55 3862.5 149.74
```

1.4.4 Problemi pri sekvenčnih metodah

V zgornjem primeru smo dobili enak rezultat izbire najboljšega modela v vseh treh primerih sekvenčnih metod. V praksi ni vedno tako. Kriterij izbire modela smo osnovali na podlagi minimalne vrednosti AIC. Namesto tega se v praksi v kontekstu sekvenčnih metod še vedno pogosto uporablja F-statistika in F-test, kar je sporno zaradi večkratnega testiranja domnev. Prisoten je tudi problem pristranosti ocen parametrov modela, ker se isti podatki uporabljajo za oceno parametrov in za proces izbire najboljšega modela.

Za ocene parametrov velja, da so nepristrane, če je vnaprej izbrani model pravi. V primeru, ko model izberemo na podlagi sekvenčne metode, inferenca na parametrih modela ni več upravičena. Ko uporabljamo sekvenčne metode, želimo odgovoriti na vprašanje, katera množica regresorjev vrne najboljšo napoved. Kakršnakoli nadaljnja inferenca ni veljavna, dobljeni model sam v tem primeru predstavlja inferenco.

2 VAJE

2.1 Napovedovanje porabe goriva

Zanima nas, kako bi najbolje napovedali porabo goriva na avtocestah v odvisnosti od lastnosti avtomobila: `MPG.highway, Weight, EngineSize, Horsepower, Type` in `Origin`. Podatki so v podatkovnem okviru `Cars93` v paketu `MASS`. Uporabite različne pristope za izbiro najboljšega modela za napovedovanje porabe goriva: $PRESS$-statistika, navzkrižno preverjanje, C_p-statistika, AIC, sekvenčne metode.

```
> library(car)
> library(MASS)
> # spremenimo podatke v nam razumljive merske enote.
> Cars93$Poraba<-235.21/Cars93$MPG.highway  # v l/100 km
> Cars93$Masa<-Cars93$Weight*0.45359/100    # v 100 kg
> Cars93$Prostornina<-Cars93$EngineSize     # v litih
> Cars93$Moc<-Cars93$Horsepower             # v KM
> Cars93$Poreklo<-Cars93$Origin
> Cars93$Tip<-Cars93$Type
> avti <- subset(Cars93, select=c(Poraba, Masa, Prostornina, Moc, Poreklo, Tip))
> rownames(avti)<-Cars93$Make
> str(avti)

'data.frame':        93 obs. of  6 variables:
 $ Poraba     : num  7.59 9.41 9.05 9.05 7.84 ...
 $ Masa       : num  12.3 16.1 15.3 15.4 16.5 ...
 $ Prostornina: num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
 $ Moc        : int  140 200 172 172 208 110 170 180 170 200 ...
 $ Poreklo    : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ...
 $ Tip        : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3 2 ...
```

Izbira modela, ki da najbolj kakovostne napovedi

```
> # pripravimo formule za vse možne modele
> nap.sprem<-names(avti)
> (nap.sprem<-nap.sprem[!nap.sprem %in% c("Poraba")])

[1] "Masa"        "Prostornina" "Moc"         "Poreklo"     "Tip"

> n<-length(nap.sprem)
> id<-unlist(lapply(1:n, function(i) combn(1:n, i, simplify=FALSE)),
+            recursive=FALSE)
> formule<-sapply(id,
+                 function(i) paste("Poraba~", paste(nap.sprem[i], collapse="+")))
> class(formule)
```

```
[1] "character"

> formule[1:10]

 [1] "Poraba~ Masa"             "Poraba~ Prostornina"
 [3] "Poraba~ Moc"              "Poraba~ Poreklo"
 [5] "Poraba~ Tip"              "Poraba~ Masa+Prostornina"
 [7] "Poraba~ Masa+Moc"         "Poraba~ Masa+Poreklo"
 [9] "Poraba~ Masa+Tip"         "Poraba~ Prostornina+Moc"

> (length(formule))  # naredimo lahko 2^k-1= 31 različnih modelov,

[1] 31

> # model brez regresorja (ničelni model) pri tem ni vključen
```