Poglavje 1

Reševanje sistema enačb

Ko smo izpeljali metodo najmanjših kvadratov, smo dobili enačbo 1.1. Matrika koeficientov ($\mathbf{X}^T\mathbf{X}$) je redko polnega ranga, zato nima inverzne matrike. Da dobimo rešitve ($\widehat{\beta}$), se poslužimo posplošene inverzne matrike, ki jo označujemo z znakom $-$ v eksponentu.

$$\mathbf{X}^T\mathbf{X}\widehat{\beta} = \mathbf{X}^T\mathbf{y} \qquad \text{... (1.1)}$$

$$\widehat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{y} \qquad \text{... (1.2)}$$

V proceduri IML moramo opraviti naslednje korake.

1. V SAS-ovi podatkovni zbirki preverimo, če imamo uvožene podatke. Če podatkov nimamo, jih moramo uvoziti (algoritem ??).
2. Z ukazom Use izberemo niz podatkov (npr. "ovce"), morda potrebujemo tudi opis stolpcev (algoritem 1.1).
3. Iz stolpcev izbranega niza podatkov v SAS-ovi podatkovni zbirki preberemo vplive in lastnosti
4. Iz vektorjev z vplivi naredimo delne matrike dogodkov po vplivih in jih spnemo v celotno matriko dogodkov $\mathbf{X}$.
5. Potrebujemo še transponirano matriko dogodkov (algoritem 1.3). Z množenjem transponirane matrike dogodkov z matriko dogodkov dobimo matriko koeficientov ($\mathbf{X}^T\mathbf{X}$) v sistemu enačb za metodo najmanjših kvadratov, medtem ko z množenjem transponirane matrike dogodkov z vektorjem opazovanj dobimo desno stran ($\mathbf{X}^T\mathbf{y}$) enačb.
6. V zadnjem koraku poiščemo posplošeno inverzno matriko ($\left(\mathbf{X}^T\mathbf{X}\right)^{-}$) in jo pomnožimo z desno stranjo. Tako pridemo do rešitev oz. ocen parametrov ($\widehat{\beta}$) v sistematskem delu modela.

1.1 Preberemo podatke iz tabele v SAS-ovi podatkovni zbirki v vektorje in matrike

Za prikaz matrik dogodkov, matrik varianc in kovarianc, matrik koeficientov in sistema enačb za metodo najmanjših kvadratov ter mešanega modela bomo najprej uporabili podatke o rasti jagnjet (tabela 1.1). Od tu dalje delamo s podatki, ki so pred tem uvoženi v podatkovno zbirko (knjižnico work).

V stavku 1 algoritma 1.1 zaženemo proceduro IML in sprožimo tiskanje ukazom "RESET PRINT".

V drugem stavku smo z ukazom USE, da bomo uporabili podatke iz tabele OVCE.

V naslednjih stavkih smo prebrali vse stolpce in vrstice iz niza podatkov in tako dobimo vektorje (ali matrike) z vplivi in lastnostmi. Podatke lahko preberemo naenkrat in vektorji bodo poimenovani po stolpcih. Če želimo stolpce preimenovati, jih beremo lahko ločeno. V vrstici 4 in 5 smo vektorja poimenovali po stolpcu v tabeli iz podatkovne zbirke, v vrsticah 6 do 9 pa smo jih preimenovali. Vpliv živali imamo naveden v treh stolpcih, in sicer v 7. stolpcu imamo oznako jagnjeta, v naslednjem očeta in nato še mater. Stolpce tabele in vektorje, prebrane v vrsticah 10, 11 in 12 (algoritem 1.1), smo poimenovali enako in se hkrati izognili šumnikom.

Tabela 1.1: Poskus vzreje jagnjet do prodaje

Pasma	Spol	Rejec	RojMasa	StarProd	ProdMasa	Žival	Oče	Mati
T	1	KM	4,0	90	38,7	j1	13	17
T	2	KM	3,5	85	35,3	j2	13	17
T	1	ZA	3,0	95	37,9	j3	14	17
T	2	ZA	3,9	85	28,3	j4	14	17
JS	1	FT	3,4	88	26,2	j5	15	18
JS	2	FT	3,2	91	37,3	j6	15	18
JS	1	JU	3,0	97	33,4	j7	16	18
JS	2	JU	2,6	100	29,7	j8	16	18
B	1	KA	33	88	31,7	j9	-	-
B	2	KA	2,9	93	28,3	j10	-	-
B	1	RT	3,6	75	25,0	j11	-	-
B	2	RT	3,1	93	30,2	j12	-	-

Algoritem 1.1 Kreiranje vektorjev in matrik iz niza podatkov v SAS-ovih podatkovnih zbirk

```
/* ************************************************************************************ */
/*                          preberemo podatke iz NizPodatkov                            */
/* ************************************************************************************ */
proc iml; reset print;  /* ------------------------------------------  zaženemo proceduro IML   1. */
use ovce; /* ------------------------------------------------------------  uporabimo tabelo ovce  2. */
read all var  {pasma}    into pasma; /*  lahko stolpce preberemo ločeno v vektorje z istim imenom 3. */
read all var  {spol}     into spol;  /*  ---------------------------------------------------------- 4. */
read all var  {rejec}    into owner; /* ---------------------------------- ali pa jih preimenujemo 5. */
read all var  {RojMasa}  into y1;    /*  ---------------------------------------------------------- 6. */
read all var  {ProdMasa} into y2; /* ---- lahko preberemo kasneje, ko bomo rabili maso ob prodaji 7. */
read all var  {StarProd} into xij; /*  ------------------------------------------------------------ 8. */
read all var  {zival}    into zival; /*  --------------------------------------------------------- 9. */
read all var  {oce}      into oce; /*  ------------------------------------------------------------ 10. */
read all var  {mati}     into mati; /*  ---------------------------------------------------------- 11. */
/* ...................................................................................... */
print y1, y2; /* ------------------- z ukazom print lahko preverite vsebino posameznih vektorjev 12. */
close ovce; /* ------------------------------ sprostimo kanal do tabele ovce, ko ga ne rabimo več 13. */
```

S prvim ukazom "READ ALL VAR" (vrstica 3 iz algoritma 1.1) smo prebrali iz stolpca navedenem v zavitih oklepajih ("pasma") vse podatke v vektor "pasma". Tu smo uporabili isti naziv za stolpec v nizu podatkov "ovce" in za vektor. Naprej bomo uporabljali samo vektor "pasma". Ko smo prvič delali vajo z IML, smo uporabili različna imena, da bi se naučili, da uporabljamo vektor. V vektorju "pasma" so znakovne vrednosti - oznake pasem.

```
pasma      12 rows       1 col      (character, size 2)

                          T
                          T
                          T
                          T
                          JS
                          JS
                          JS
                          JS
                          B
                          B
                          B
                          B
```

read all var {pasma} into pasma;

Z drugim "READ" stavkom bomo iz niza podatkov "ovce" iz stolpca "spol" prebrali podatkov o spolu in jih spravili v vektor "spol". Vrednosti v stolpcu, in zato tudi v vektorju, so numerične. Tudi v tem primeru smo stolpec in vektor poimenovali enako, v nadaljevanju bomo uporabljali samo vektor.

``spol 12 rows 1 col (numeric)``` 1 2 1 2 1 2 1 2 1 2 1 2	read all var {spol} into spol;

V tretjem "READ" stavku smo iz stolpca "rejec" prebrali znakovne vrednosti in jih shranili za nadaljnjo obdelavo v vektor "owner". V tem primeru se stolpec iz niza podatkov in naziv vektorja razlikujeta. Običajmo se v svoji kodi držimo istega sistem pri poimenovanju stolpcev, vektorjev in matrik, da si lažje zapomnimo in čez čas hitreje preberemo vsebino kode.

owner 12 rows 1 col (character, size 2) KM KM ZA ZA FT FT JU JU KA KA RT RT	read all var {rejec} into owner;

V zadnjem "READ" stavku smo prebrali še rojstno maso iz stolpca "RojMasa" in jih shranili v vektor y1, ker jo bomo uporabili kot lastnost.

y1 12 rows 1 col (numeric) 4 3.5 3 3.9 3.4 3.2 3 2.6 3.3 2.9 3.6 3.1	read all var {RojMasa} into y1;

1.2 Postavitev matrike dogodkov za sistematske vplive

Vzemimo, da model za rojstno maso (y_{ijkl}) vključuje pasmo (P_i), spol (S_j) in rejec (R_k) kot sistematske vplive (enačba 1.3). V modelu imamo samo vplive z razredi.

$$y_{ijkl} = \mu + P_i + S_j + R_k + e_{ijkl} \qquad \text{... (1.3)}$$

Algoritem 1.2 Kreiranje matrike dogodkov za sistematske vplive

```
/* ********************************************************************************** */
/* nastavimo matriko dogodkov X za model  y1 = mi + P + S + R * e                      */
/* ********************************************************************************** */
mi = J(12,1,1);        /* --------------------- oblikujemo stolpični enotski vektor   1. */
XP = design(pasma);    /* ------------------------- delna matrika dogodkov za pasmo   2. */
XS = design(spol);     /* ---------------------------- delna matrika dogodkov za spol 3. */
XR = design(owner);    /* -------------------------- delna matrika dogodkov za rejca  4. */
X  = mi||XP||XS;   /* -------------------- matrika dogodkov za sistematski del modela  5. */
X  = mi||XP||XS||XR;   /* ---------------- matrika dogodkov za sistematski del modela  6. */
Plist = unique(pasma); /* --------------------------------- seznam nivojev pri pasmi  7. */
Slist = unique(spol); /* ------------------------------------ seznam nivojev pri spol  8. */
Rlist = unique(owner); /* -------------------------------- seznam nivojev pri rejcih  9. */
```

Vzemimo, da model za maso ob prodaji (y_{ijkl}) vključuje pasmo (P_i), spol (S_j), rejca (R_k) in starost ob prodaji (x_{ijkl}) kot sistematske vplive (enačba 1.3). V trem vplivom z razredi smo dodali še kvantitativni vpliv starosti ob prodaji.

$$y_{ijkl} = \mu + P_i + S_j + R_k + b(x_{ijkl} - \overline{x}) + e_{ijkl} \qquad \text{... (1.4)}$$

1.2.1 Kvalitativni glavni vplivi

Nastavimo matriko dogodkov za sistematski del modela (enačba 1.5). Parametre praviloma označujemo s številčnimi indeksi. V tem delu pa smo navedli kar oznake pasem in rejcev ter jih razvrstila tako, kot se razvrstijo v funkciji DESIGN.

$$\begin{bmatrix} 4,0 \\ 3,5 \\ 3,0 \\ 3,9 \\ 3,4 \\ 3,2 \\ 3,0 \\ 2,6 \\ 3,3 \\ 2,9 \\ 3,6 \\ 3,1 \end{bmatrix} \quad \begin{array}{c} [\mu \quad P_B \quad P_{JS} \quad P_T \quad S_1 \quad S_2 \quad R_{FT} \quad R_{JU} \quad R_{KM} \quad R_{KA} \quad R_{RT} \quad R_{ZA}] \\ \begin{bmatrix} 1 & & & 1 & 1 & & & & & 1 & & \\ 1 & & & 1 & & 1 & & & & 1 & & \\ 1 & & & 1 & 1 & & & & & & & 1 \\ 1 & & & 1 & & 1 & & & & & & 1 \\ 1 & & 1 & & 1 & & 1 & & & & & \\ 1 & & 1 & & & 1 & 1 & & & & & \\ 1 & & 1 & & 1 & & & 1 & & & & \\ 1 & & 1 & & & 1 & & 1 & & & & \\ 1 & 1 & & & 1 & & & & 1 & & & \\ 1 & 1 & & & & 1 & & & 1 & & & \\ 1 & 1 & & & 1 & & & & & & 1 & \\ 1 & 1 & & & & 1 & & & & & 1 & \end{bmatrix} \end{array} \qquad \text{... (1.5)}$$

Sedaj pa nastavimo matriko dogodkov za sistematski del modela še v proceduri IML (algoritem 1.2), vendar tako, da se poslužimo orodij za kreiranje matrik.

Matriko dogodkov za sistematski del modela (**X**, enačba 1.5) bomo v proceduri IML naredili v petih stavkih, v prvih štirih bomo naredili delne matrike za srednjo vrednost in tri vplive, v zadnjem pa bomo delne matrike sestavili. S kodo od 6. do 10. stavka pa smo prikazali dva načina oblikovanja delne matrike za interakcijo med dvema vplivoma.

V prvem stavku bomo nastavili prvi stolpec, ki odgovarja delu matrike dogodkov **X**, ki povezuje dogodke (meritve) s parametrom μ. Vektor, ki smo ga poimenovali "mi", ima 12 vrstic in ima v stolpcu same 1. Uporabili smo se funkcije J().

```
mi     12 rows      1 col     (numeric)

                    1
                    1
                    1
                    1
                    1
                    1
                    1
                    1
                    1
                    1
                    1
                    1
```

```
mi=J(12,1,1);
```

V drugem stavku smo kreirali delno matriko XP, ki ima 12 vrstic in tri stolpce. Za kreiranje smo uporabili funckijo DESIGN(). V PROCEDURI IML bodo znakovne vrednosti prekodirane, in sicer v abecednem vrstnem redu. Tako bo prvi stolpec, ki je namenjen pasmi 1, namenjen pasmi B, drugi JS pasmi in tretji pasmi T. Če bi matriko dogodkov delali na papir, bi najbrž kodirali pasme po vrsti, kot se pojavljajo v podatkih. Dokler se držimo istega sistema kodiranja, se bo tudi izšlo.

```
XP     12 rows      3 cols    (numeric)


               0         0         1
               0         0         1
               0         0         1
               0         0         1
               0         1         0
               0         1         0
               0         1         0
               0         1         0
               1         0         0
               1         0         0
               1         0         0
               1         0         0
```

```
XP=design(pasma);
```

V tretjem stavku smo kreirali delno matriko XS, ki ima 12 vrstic in dva stolpca, ker imamo v podatkih samo dva spola. Ker je oznaka spola numerična, se pri kreiranju delne matrike ni zgodilo nič nepričakovanega.

```
XS     12 rows      2 cols    (numeric)


                    1         0
                    0         1
                    1         0
                    0         1
                    1         0
                    0         1
                    1         0
                    0         1
                    1         0
                    0         1
                    1         0
                    0         1
```

```
XS=design(spol);
```

V četrtem stavku smo kreirali delno matriko XR, ki ima 12 vrstic in šest stolpcev, ker imamo tudi šest rejcev. Rejce smo vnesli z znakovno kodo, zato so prekodirani po abecednem vrstnem redu.

XR	12 rows	6 cols	(numeric)			
0	0	0	1	0	0	
0	0	0	1	0	0	
0	0	0	0	0	1	
0	0	0	0	0	1	
1	0	0	0	0	0	
1	0	0	0	0	0	
0	1	0	0	0	0	
0	1	0	0	0	0	
0	0	1	0	0	0	
0	0	1	0	0	0	
0	0	0	0	1	0	
0	0	0	0	1	0	XR=design(owner);

V petem stavku smo kreirali matriko dogodkov za sistematski del modela (**X**) brez vpliva rejca. To je samo zato, da jo lahko prikažemo pri tem opisu. Matriko z rejcem bomo prikazali nekoliko kasneje. Preverite z matriko v enačbi 1.5, vendar brez zadnjih šest stolpcev, ki pripadajo vplivu rejca.

1	0	0	1	1	0	
1	0	0	1	0	1	
1	0	0	1	1	0	
1	0	0	1	0	1	
1	0	1	0	1	0	
1	0	1	0	0	1	
1	0	1	0	1	0	
1	0	1	0	0	1	
1	1	0	0	1	0	
1	1	0	0	0	1	
1	1	0	0	1	0	
1	1	0	0	0	1	X=mi||XP||XS;

Matriko bi lahko nastavili tudi v enem korak, in sicer s stavkom:

```
X = J(12,1,1)||design(pasma)||design(spol)||design(owner);
```

Edina pomanjkljivost takšnega zapisa je, da je morda matrika pri preverjanju manj pregledna kot delne matrike. Zato na začetku, ko pišemo kodo priporočamo delo v več korakih. Algoritem bi lahko še nekoliko izpopolnili. Da smo naredili prvi stolpec za μ, smo morali prešteti zapise. To je bolje, če uporabimo funkcijo NROW(), ki nam vrne število vrstic. Število vrstic iz vektorja opazovanj smo shranili v skalar in ga v naslednji vrstici tudi uporabili.

```
nv = nrow(y1);
X = J(nv,1,1)||design(pasma)||design(spol)||design(owner);
```

V šestem stavku smo kreirali matriko dogodkov, ki poleg pasme in spola vključuje vpliv rejca. Izpis te matrike dogodkov je prikazan kasneje, saj smo sistem enačb nadaljevali na tem modelu.

V sedmem, osmem in devetem stavku smo naredili liste za nivoje posameznih vplivov, ki jih lahko združimo tako kot matrike dogodkov. Tako ohranimo vrstni red parametrov v modelu in lahko pojasnimo rešitve.

1.2.2 Kreiranje delne matrike dogodkov za interakcijo

Najprej sestavimo matriko dogodkov za interakcijo na roke. Interakcijo med pasmo in spolom smo vstavili med stolpcema za spol in pred stolpci za rejca. nekaj stolpcev za rejca smo izbrisali, ker nam manjka prostora. delne matrike za glavne vplive so ostale enake. Označujemo jih s črkami glavnih vplivov, indekse napišemo za oznako interakcije. V enačbi 1.6 najhitreje menjamo indeks za spol in nato za pasmo. Delno matriko za interakcijo smo obarvali rdeče.

$$
\begin{array}{c}
\begin{array}{ccccccccccccccc}
\mu & P_B & P_{JS} & P_T & S_1 & S_2 & PS_{B1} & PS_{B2} & PS_{JS1} & PS_{JS2} & PS_{T1} & PS_{T2} & R_{FT} & R_{JU} & \cdots
\end{array} \\
\left[\begin{array}{ccccccccccccccc}
1 & & & 1 & 1 & & & & & & {\color{red}1} & & & & \cdots \\
1 & & & 1 & & 1 & & & & & & {\color{red}1} & & & \cdots \\
1 & & & 1 & 1 & & & & & & {\color{red}1} & & & & \cdots \\
1 & & & 1 & & 1 & & & & & & {\color{red}1} & & & \cdots \\
1 & & 1 & & 1 & & & & {\color{red}1} & & & & 1 & & \cdots \\
1 & & 1 & & & 1 & & & & {\color{red}1} & & & 1 & & \cdots \\
1 & & 1 & & 1 & & & & {\color{red}1} & & & & & 1 & \cdots \\
1 & & 1 & & & 1 & & & & {\color{red}1} & & & & 1 & \cdots \\
1 & 1 & & & 1 & & {\color{red}1} & & & & & & & & 1 \\
1 & 1 & & & & 1 & & {\color{red}1} & & & & & & & 1 \\
1 & 1 & & & 1 & & {\color{red}1} & & & & & & & & \cdots \\
1 & 1 & & & & 1 & & {\color{red}1} & & & & & & & \cdots
\end{array}\right]
\end{array}
\quad \text{... (1.6)}
$$

V enačbi 1.7 najhitreje menjamo indeks za pasmo in nato za spol. Oba prikaza sta pravilna, a paziti moramo, da vrstnega reda nivojev pri posameznih vplivih ne spreminjamo. Tudi v tej enačbi smo delno matriko za interakcijo obarvali rdeče.

$$
\begin{array}{c}
\begin{array}{ccccccccccccccc}
\mu & P_B & P_{JS} & P_T & S_1 & S_2 & PS_{B1} & PS_{JS1} & PS_{T1} & PS_{B2} & PS_{JS2} & PS_{T2} & R_{FT} & R_{JU} & \cdots
\end{array} \\
\left[\begin{array}{ccccccccccccccc}
1 & & & 1 & 1 & & & & {\color{red}1} & & & & & & \cdots \\
1 & & & 1 & & 1 & & & & & & {\color{red}1} & & & \cdots \\
1 & & & 1 & 1 & & & & {\color{red}1} & & & & & & \cdots \\
1 & & & 1 & & 1 & & & & & & {\color{red}1} & & & \cdots \\
1 & & 1 & & 1 & & & {\color{red}1} & & & & & 1 & & \cdots \\
1 & & 1 & & & 1 & & & & & {\color{red}1} & & 1 & & \cdots \\
1 & & 1 & & 1 & & & {\color{red}1} & & & & & & 1 & \cdots \\
1 & & 1 & & & 1 & & & & & {\color{red}1} & & & 1 & \cdots \\
1 & 1 & & & 1 & & {\color{red}1} & & & & & & & & 1 \\
1 & 1 & & & & 1 & & & & {\color{red}1} & & & & & 1 \\
1 & 1 & & & 1 & & {\color{red}1} & & & & & & & & \cdots \\
1 & 1 & & & & 1 & & & & {\color{red}1} & & & & & \cdots
\end{array}\right]
\end{array}
\quad \text{... (1.7)}
$$

Sedaj pa si poglejmo, kako pridemo do delnih matrik v proceduri v IML. V algoritmu 1.3 imamo ponujenih več možnosti. Različni postopki možnost lahko dajo različno delno matriko. Paziti moramo le, da vemo, katerim parametrom pripadajo posamezni stolpci. Zgornji dve enačbi ne kažeta vseh možnih razvrstitev, sta pa bolj pogosti.

Prva možnost je oblikovanje novega stolpca kar v tabeli, v kateri imamo shranjene podatke (vrstice od 1 do 5 algoritma 1.3). V prvi vrstici povemo, v katerem nizu želimo narediti spremembo. Nov stolpec dodamo istemu nizu podatkov, ali pa naredimo nov niz. V drugi vrstici določimo vrednost, oznako za vsak nivo pri interakciji. Najprej moramo numerično oznako za spol spremeniti v znakovno oznako s funkcijo CHAR(), potem pa obe znakovni spremenljivki s funkcijo STRIP() oklestimo presledkov. Ker ima spol en znak, pasma pa dva ali enega, je oznako lažje prepoznati, če je spol na prvem mestu. Stolpec nato preberemo v vektor (vrstica 3) .

Druga možnost je podana v vrstici 4 v algoritmu 1.3. Na enak način kot stolpec v tabeli, lahko oblikujemo vektor iz vektorjev za glavne vplive, v našem primeru vektorjev za spol in pasmo.

Prva in druga možnost (pravilno) izpustita nivoje, pri katerih ni opazovanj. Lahko si oblikujemo tudi listo nivojev za interakcijo tako, da iz vektorja interSP s funkcijo UNIQUE() vzamemo nivo le enkrat in ga shranimo v vektor PSlist (vrstica .5). Ko oblikujemo vektor za interakcijo, nadaljujemo z vrstico 6 in s funkcijo DESIGN() oblikujemo delno matriko.

Tretjo možnost prikazujemo v vrsticah 5 in 6 algoritma 1.3. Uporabili smo že oblikovane matrike dogodkov za pasmo in spol. V šesti vrstici smo delno matriko za pasmo množili po parih prvič s prvim stolpcem za spol in nato še z drugim stolpcem za spol in oba zmnožka sestavili s horizontalnim spajanjem. V tem primeru lahko dobimo tudi prazen stolpec za nivo interakcije brez opazovanj. Stolpec bi naj zbrisali. V šesti vrstici smo delno matriko za spol množili po parih s posameznimi stolpci iz delne matrike za pasmo.

Algoritem 1.3 Kreiranje matrike dogodkov za sistematske vplive

```
/* ********************************************************************************** */
/*      nastavimo matriko dogodkov X za model  y1 = mi + P + S + PS + R + e             */
/* ********************************************************************************** */
data NizPodatkov; set  NizPodatkov; /* -------------- tabeli bomo dodali nov stolpec  1. */
/* ---------------------------- pasmo sem dala na drugo mesto, ker je različno dolga.   */
   interPS = strip(char(spol))||strip(pasma);  /* ----------- definiramo nov stolpec   2. */
run;
read all var  {interPS}  into interPS; /* ---------------- preberemo stolpec v vektor  3. */
/* .................................................................................... */
interPS = strip(char(spol))||strip(pasma);  /* ---- oblikujemo vektor kar iz vektorje  4. */
PSlist = unique (interPS);   /* ------------------------- sestavimo listo parametrov  5. */
XPS = design(interPS);   /* -------------- oblikujemo delno matriko s funkcijo design 6. */
/* ********************************************************************************** */
/*      nastavimo matriko dogodkov X za model  y1 = mi + P + S + PS + R + e             */
/* ********************************************************************************** */
XPS  = XP#XS[ ,1]||XP#XS[ ,1]; /* - matrika dogodkov z interakcijo med pasmo in spolom  7. */
XSP  = XS#XP[ ,1]||XS#XP[ ,2]||XS#XP[ ,3]; /* -- ali z interakcijo med spolom in pasmo  8. */
/* .................................................................................... */
/*         do delne matrike za interakcijo lahko pridemo tudi bolj splošno               */
XSP  = XS#XP[ ,1];  /* -------------- prvi blok pri interakciji za pasmo=1 in oba spola  9.*/
Do i = 2 to ncol(XP);   /* ---------------------------------------------------------- 10.*/
   XSP = XSP || XS#XP[ ,i]; /* ------ dodamo naslednji blok za pasmo = i in oba spola  11.*/
End;
/* .................................................................................... */
XI  = mi||XP||XS||XPS||XR;   /* --------------------- matrika dogodkov z interakcijo  12.*/
/* ********************************************************************************** */
```

Četrta možnost je samo bolj splošen postopek, kot sta postopka v vrstici 5 in 6. Ni nam potrebno vedeti, koliko stolpcev ima druga matrika, iz katere pobiramo stolpce za množenje po parih s prvo matriko. To smo prešteli s funkcijo NCOL() v vrstici 9. Najprej smo morali oblikovati za izhodišče delno matriko dogodkov XSP in ji tako določili število vrstic. Če tega ne bi storili, bi oznaka XSP v vrstici 10 na desni strani enačaja bila skalar, ki pa ga ne moremo speti z matriko, ki jo dobimo za znakom ||. Do zanko potrebujemo, da prvo matriko množi po parih z vsemi stolpci od drugega do zadnjega, s prvim stolpcem pa smo opravili že v vrstici 8. S stavkom END med vrsticama 11 in 12 zaključujemo DO zanko. Ta postopek je zahtevnejši, je pa bolj splošen in lahko vanj vključimo še posebnosti.

Na koncu (vrstica 12 v algoritmu 1.3) smo sestavili še celotno matriko dogodkov za sistematske vplive. Uporabimo lahko možnost, ki nam v danem primeru bolj odgovarja.

1.2.3 Kreiranje delne matrike dogodkov za kvantitativne vplive

Delne matrike za kvalitativne vplive so enake za obe lastnosti, ker so v modelu isto kvalitativni vplivi, prav tako pa so enaki tudi vplivi v podatkih.

Kvantitativni vpliv smo vključili v model za maso ob prodaji (enačba 1.4).

V prvem stavku smo izračunali povprečje s funkcijo AVG(), v drugi vrstici pa prešteli še vrstice v vektorju z neodvisno spremenljivko. Uporabili smo funkcijo NROW().

V tretjem in četrtem stavku smo prikazali dva načina, kako kvantitativnemu vplivu odštejemo povprečje. V prvem smo preprosto odšteli skalar, česar matrična algebra ne pozna, v proceduri IML pa se povprečje odšteje vsaki vrstici v vektorju. V vrstici 4 prav tako odštejemo povprečje in je bližje matematičnemu zapisu.

V petem stavku smo zapisali parameter iz linearne regresije. Parametri pri kvantitativnih vplivih so regresijski koeficienti in jih praktično vedno označujemo zs črko "b".

V šestem stavku smo sestavili matriko dogodkov za sistematske vplive v modelu 1.4 za maso ob prodaji.

V sedmem stavku smo stavku smo naredili delno matriko za ugnezdeno linearno regresijo, in sicer znotraj spola. Delno matriko smo dobili z množenjem parov delne matrike za vpliv spola, znotraj katerega smo ugnezdili linearno regresijo, z vektorjem z neodvisno spremenljivko.

Algoritem 1.4 Kreiranje delne matrike dogodkov za kvantitativne vplive

```
/* ************************************************************************************ */
/*    nastavimo matriko dogodkov X za model  y1 = mi + P + S  + R + b(xij-povp)+ e       */
/* ************************************************************************************ */
/* .................................................................................... */
povp  = avg(xij);  /* ---------------- izračunamo povprečje za neodvisno spremenljivk  1. */
nv    = nrow(xij);  /* -------------------------------------------- preštejemo vrstice  2. */
xij   = xij - povp;  /* --------------------------------------- odštejemo povprečje  3. */
xij   = xij - J(nv,1,povp) ;  /* ------------------- odštejemo povprečje malo drugače  4. */
xlist = “b”;   /* ------------------------------------------ sestavimo listo parametrov  5. */
X2    = mi||XP||XS||XR||xij;   /* ------------------------ oblikujemo matriko dogodkov 6. */
/* ************************************************************************************ */
/*    nastavimo matriko dogodkov X za model  y1 = mi + P + S + R + bi(xij-povp) + e      */
/* ************************************************************************************ */
XSxij = XS # xij; /* ------------------------- delna matrika dogodkov za bi(xij-povp)  7. */
XPxij = XP # xij; /* -------------------------- ali z interakcijo med spolom in pasmo  8. */
xSlist = strip(char(Slist)) # “b”; /* -------------- kreiramo poimenovanje parametrov  9. */
xPlist = strip(Plist) # “b”; /* ------------------- kreiramo poimenovanje parametrov  10. */
X2    = mi||XP||XS||XR||XSxij; /* ----------------------- oblikujemo matriko dogodkov 11. */
/* ************************************************************************************ */
/*  nastavimo matriko X za model y1 = mi + P + S + R + bi(xij-povp)+ biII(xij-povp)2 + e  */
/* ************************************************************************************ */
xij2   = xij # xij; /* ----------------------- delna matrika dogodkov za bi(xij-povp) 12. */
XSxij2 = XS  # xij2; /* ------------------- delna matrika dogodkov za biII(xij-povp)2 13. */
XPxij2 = XP  # xij2; /* ------------------- delna matrika dogodkov za bjII(xij-povp)2 14. */
/* .................................................................................... */
X2  = mi||XP||XS||XR||XSxij||XSxij2;  /* ----- matrika dogodkov z ugnezdeno regresijo 15. */
X2  = mi||XP||XS||XR||XPxij||XPxij2;  /* ----- matrika dogodkov z ugnezdeno regresijo 16. */
/* ************************************************ ************************************ */
```

Algoritem 1.5 Postavitev matrike koeficientov in desne strani sistema enačb

```
X    = mi||XP||XS||XR;    /* --------------- matrika dogodkov za sistematski del modela  1. */
tX   = t(X);            /* --------------------- transponirana oblika matrike dogodkov  2. */
tXX  = tX*X;      /* ------------------------ matrika koeficientov: produkt dveh matrik  3. */
gXX  = ginv(tXX); /* ------------------------------------ inverza matrike koeficientov  4. */
Xy   = tX*y1;     /* ---------------------------------------------------- desna stran  5. */
beta = gXX*Xy;   /* -------------------------------- dobimo beta - rešitev sistema enačb 6. */
beta = ginv(t(X)*X)*t(X)*y;   /* ------ lahko tudi s to kodo pridemo do rešitev sistema  7. */
```

V osmem stavku smo ugnezdili linearno regresijo znotraj pasme. Prav tako smo uporabili množenje po parih.

V devetem in desetem stavku smo oblikovali nazive pa parametre. Uporabimo samo tiste stavke, ki se nanašajo na model, ki ga obdelujemo.

V enajstem stavku smo sestavili matriko dogodkov za model, ki vsebuje poleg vplivov z razredi tudi kvantitativni vpliv, ki smo ga opisali z linearno regresijo ugnezdeno znotraj spola-

1.3 Metoda najmanjših kvadratov

V algoritmu 1.5 smo prikazali en način, kako nastavimo matriko dogodkov oz. levo stran in desno stran sistema enačb. Najprej bomo prikazali primer za metodo najmanjših kvadratov (enačba 1.1). Ker smo pri postavitvi matrike dogodkov za sistematski del modela (**X**) izpustili vpliv rejca, bomo v prvi vrstici ponovno sestavili vse delne matrike, vključno z delno matriko za rejca.

V prvi vrstici kode smo ponovno sestavili delne matrike v matriko dogodkov za sistematski del modela, v katerega smo vključili pasmo, spol in rejca.

V drugi vrstici kode smo poiskali transponirano matriko dogodkov za sistematski del modela. V naslednjem prikazu prikazujemo samo matriko **X**, matriko koeficientov (tXX), ki smo jo dobili v **tretji vrstici** algoritma in njeno posplošeno inverzo (gXX) iz **četrte vrstice** algoritma.

V peti vrstici kode smo postavili desno stran enačbe, kjer dobimo posamezne vsote opazovanj po posameznih nivojih. V tem primeru smo jo označili z Xy, pogosto uporabimo tudi RHS iz angleščine, poimenujemo ga pa lahko tudi po slovensko.

<pre>Xy 12 rows 1 col (numeric) 39.5 12.9 12.2 14.4 20.3 19.2 6.6 5.6 6.2 7.5 6.7 6.9</pre>	Xy=t(X)*y;

V šesti vrstici kode smo izračunali rešitve (ocene parametrov v β) z množenjem posplošene inverze matrike koeficientov z desno stranjo. Tudi v **sedmi vrstici** dobimo rešitve. V tem primeru smo se povsem izognili vmesnim korakom. Če še nismo vešči in nam ni jasno, kaj s kodo dobimo, je priporočljiva pot z več manjšimi koraki.

X

1	0	0	1	1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	1
1	0	0	1	0	1	0	0	0	0	0	1
1	0	1	0	1	0	1	0	0	0	0	0
1	0	1	0	0	1	1	0	0	0	0	0
1	0	1	0	1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1	0	0	0	0
1	1	0	0	1	0	0	0	1	0	0	0
1	1	0	0	0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	0	0	0	1	0
1	1	0	0	0	1	0	0	0	0	1	0

tXX

COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12
12	4	4	4	6	6	2	2	2	2	2	2
4	4	0	0	2	2	0	0	2	0	2	0
4	0	4	0	2	2	2	2	0	0	0	0
4	0	0	4	2	2	0	0	0	2	0	2
6	2	2	2	6	0	1	1	1	1	1	1
6	2	2	2	0	6	1	1	1	1	1	1
2	0	2	0	1	1	2	0	0	0	0	0
2	0	2	0	1	1	0	2	0	0	0	0
2	2	0	0	1	1	0	0	2	0	0	0
2	0	0	2	1	1	0	0	0	2	0	0
2	2	0	0	1	1	0	0	0	0	2	0
2	0	0	2	1	1	0	0	0	0	0	2

gXX

COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12
0.0208333	0.0069444	0.0069444	0.0069444	0.0104167	0.0104167	0.0034722	0.0034722	0.0034722	0.0034722	0.0034722	0.0034722
0.0069444	0.0763889	-0.034722	-0.034722	0.0034722	0.0034722	-0.017361	-0.017361	0.0381944	-0.017361	0.0381944	-0.017361
0.0069444	-0.034722	0.0763889	-0.034722	0.0034722	0.0034722	0.0381944	0.0381944	-0.017361	-0.017361	-0.017361	-0.017361
0.0069444	-0.034722	-0.034722	0.0763889	0.0034722	0.0034722	-0.017361	-0.017361	-0.017361	0.0381944	-0.017361	0.0381944
0.0104167	0.0034722	0.0034722	0.0034722	0.0885417	-0.078125	0.0017361	0.0017361	0.0017361	0.0017361	0.0017361	0.0017361
0.0104167	0.0034722	0.0034722	0.0034722	-0.078125	0.0885417	0.0017361	0.0017361	0.0017361	0.0017361	0.0017361	0.0017361
0.0034722	-0.017361	0.0381944	-0.017361	0.0017361	0.0017361	0.2690972	-0.230903	-0.008681	-0.008681	-0.008681	-0.008681
0.0034722	-0.017361	0.0381944	-0.017361	0.0017361	0.0017361	-0.230903	0.2690972	-0.008681	-0.008681	-0.008681	-0.008681
0.0034722	0.0381944	-0.017361	-0.017361	0.0017361	0.0017361	-0.008681	-0.008681	0.2690972	-0.008681	-0.230903	-0.008681
0.0034722	-0.017361	-0.017361	0.0381944	0.0017361	0.0017361	-0.008681	-0.008681	-0.008681	0.2690972	-0.008681	-0.230903
0.0034722	0.0381944	-0.017361	-0.017361	0.0017361	0.0017361	-0.008681	-0.008681	-0.230903	-0.008681	0.2690972	-0.008681
0.0034722	-0.017361	-0.017361	0.0381944	0.0017361	0.0017361	-0.008681	-0.008681	-0.008681	-0.230903	-0.008681	0.2690972

Algoritem 1.6 Izbira posplošene inverze iz procedure GLM

```
/*Kako dobimo rešitve enake kot iz proc GLM?
GLM procedura postavi vrstico in stolpec za zadnji nivo
VSAKEGA sistematskega kvalitativnega vpliva na NIČ*/
MK=tXX;
nr=nrow(MK);
MK[4,  ] = J(1,nr,0); /* vrstico za pasmo P3 damo na nič */
MK[  ,4] = J(nr,1,0); /* stolpec za pasmo P3 damo na nič */
MK[6,  ] = J(1,nr,0); /* vrstico za spol  S2 damo na nič */
MK[  ,6] = J(nr,1,0); /* stolpec za spol  S2 damo na nič */
MK[12, ] = J(1,nr,0); /* vrstico za rejo  R6 damo na nič */
MK[ ,12] = J(nr,1,0); /* stolpec za rejo  R6 damo na nič */
beta=ginv(MK)*tX*y1;  /*rešitev kot iz proc GLM*/
```

V zadnji vrstici algoritma smo dobili rešitve sistema enačb z množenjem posplošene inverze (gXX) z desno stranjo (Xy). Pri nazivih spremenljivk v SAS-u izpuščamo posebne znake, kot je črtica za transponirano vrstico. Tu in tam smo jo nadomestili s črko t ali p. Ko pa pišete enačbe, pa posebnih znakov ne smete izpuščati!

beta 12 rows 1 col (numeric)	
1.6458333	
0.5041667	
0.3875	
0.7541667	
0.9145833	
0.73125	
0.44375	
-0.05625	
0.1270833	
0.5270833	
0.3770833	
0.2270833	beta=gXX*Xy

1.3.1 Rezultati kot v proceduri GLM

Proceduri IML in GLM izbereta različni posplošeni inverzi. Posplošenih inverz je pravzaprav neskončno mnogo, težave pa imamo, ko želimo rezultate primerjati. Vsaka posplošena inverza nam nudi različne rešitve. Da bi bila primerjava rešitev enostavna, lahko v proceduri IML dosežemo, da je uporabljena ista posplošena inverza kot v GLM-u. V GLM-u so vse enačbe, ki so linearne kombinacije preostalih, postavljene na nič. Pri glavnih vplivih z razredi postavi v vrstici in stolpcu v matriki koeficientov pri zadnjem razredu vse vrednosti na 0 (algoritem 1.6

1.4 Metoda mešanih modelov

1.4.1 Enolastnostni modeli z enostavnim naključnim vplivom

Vzemimo, da model za rojstno maso (y_{ijkl}) vključuje pasmo (P_i) in spol (S_j) kot sistematske vplive ter rejec (r_k) kot naključni vpliv (enačba 1.8). V modelu imamo samo vplive z razredi.

$$y_{ijkl} = \mu + P_i + S_j + r_k + e_{ijkl} \qquad \text{... (1.8)}$$

Ko smo izpeljali metodo najmanjših kvadratov, smo dobili enačbo 1.9. Izpeljavo mešanega sistema za eno lastnost izpeljujemo na predavanjih, za lažje razumevaje pa postopek na hitro ponovimo.

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} \quad +\mathbf{G}_r^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\beta} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{y} \end{bmatrix} \qquad \text{... (1.9)}$$

Pri enolastnostnih modelih z eno meritvijo na živali so ostanki identično in neodvisno porazdeljeni, kar pomeni, da lahko inverzno matriko prikažemo na način iz enačbe 1.10.

$$\mathbf{R}^{-1} = \left(\mathbf{I}_n \times \sigma_e^2\right)^{-1} = \mathbf{I}_n \times \sigma_e^{-2} \qquad \text{... (1.10)}$$

Če vstavimo rezultat v enačbo 1.9, dobimo enačbo 1.11.

$$\begin{bmatrix} \mathbf{X}^T\mathbf{I}_n\mathbf{X}\times\sigma_e^{-2} & \mathbf{X}^T\mathbf{I}_n\mathbf{Z}\times\sigma_e^{-2} \\ \mathbf{Z}^T\mathbf{I}_n\mathbf{X}\times\sigma_e^{-2} & \mathbf{Z}^T\mathbf{I}_n\mathbf{Z}\times\sigma_e^{-2} \quad +\mathbf{G}_r^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\beta} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{I}_n\mathbf{y}\times\sigma_e^{-2} \\ \mathbf{Z}^T\mathbf{I}_n\mathbf{y}\times\sigma_e^{-2} \end{bmatrix} \qquad \text{... (1.11)}$$

Ko množimo katerokoli matriko z identično matriko, je rezultat ista matrika, zato bomo identične matrike izpustili (enačba . 1.13). Sistem enačb pomnožimo z varianco za ostanek σ_e^2. Pri vseh členih dobimo produkt $\sigma_e^{-2}\times\sigma_e^2 = 1$, razen pri členu z matriko varianc in kovarianc za skupno okolje v gnezdu (enačba 1.12).

$$\mathbf{G}_r^{-1}\times\sigma_e^2=\left(\mathbf{I}_r\times\sigma_r^2\right)^{-1}\times\sigma_e^2=\mathbf{I}_r\times\left(\sigma_e^2\times\sigma_r^{-2}\right)=\mathbf{I}_r\times\alpha_r \qquad \text{... (1.12)}$$

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Z} \\ \mathbf{Z}^T\mathbf{X} & \mathbf{Z}^T\mathbf{Z} \quad +\mathbf{I}_r\alpha_r \end{bmatrix} \begin{bmatrix} \widehat{\beta} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}^T\mathbf{y} \end{bmatrix} \qquad \text{... (1.13)}$$

Matrika koeficientov je redko polnega ranga, zato pogosto nima inverzne matrike. Da dobimo rešitve ($\widehat{\beta}$), se poslužimo posplošene inverzne matrike, ki jo označujemo z znakom "–" v eksponentu (enačba 1.14).

$$\begin{bmatrix} \widehat{\beta} \\ \widehat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Z} \\ \mathbf{Z}^T\mathbf{X} & \mathbf{Z}^T\mathbf{Z} \quad +\mathbf{I}_r\alpha_r \end{bmatrix}^{-} \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}^T\mathbf{y} \end{bmatrix} \qquad \text{... (1.14)}$$

Da bomo nastavili sistem enačb in ga nato rešili, moramo nastaviti:

- matriko dogodkov za sistematski del modela $\mathbf{X}$
- matriko dogodkov za naključni vpliv $\mathbf{Z}$
- izračunati moramo α

Matrika dogodkov za sistematski del modela smo nastavljali pri metodi najmanjših kvadratov. Za naš primer bomo uporabili kar matriko, za katero smo delne matrike nastavili v algoritmu 1.2 in smo ga prikazali v vrstici 1 algoritma 1.7. V vrstici 2 smo prešteli stolpce s funkcijo NCOL() v matriki dogodkov za sistematski del modela.

Matrika dogodkov za naključni vpliv oblikujemo enostavno s funkcijo DESIGN(), v oklepaju pa navedemo vektor, v katerega smo prebrali izbrani naključni vpliv (vrstica 4 v algoritmu 1.7). S funkcijo UNIQUE() smo v vrstici 5 naredili seznam rejcev in ga shranili v vektor RejList. V vrstici 6 smo s funkcijo NCOL() prešteli število rejcev.

Inverzno matriko varianc in kovarianc za skupno okolje pri rejcu smo izpeljali v komentarju in enačbi 1.12). Ker ima matrika varianc in kovarianc enostavno strukturo, je tudi njena inverzna matrika matrika z enostavno strukturo.

1.4.1.1 Uporaba sestavljene matrike dogodkov za sistematski del modela in naključne vplive

Sistem enačb mešanega modela nastavimo lahko tako, da sestavimo matrike dogodkov za sistematski del modela in naključni vpliv (vrstica 1 v algoritmu 1.8. Postopek prikazujemo tudi v enačbi 1.15. V vrstici 2 smo prešteli stolpce sestavljeni matriki dogodkov, nato pa smo v vrstici 3 pomnožili transponirano matriko $t(\mathbf{XZ})$ s sestavljeno matriko dogodkov $\mathbf{XZ}$. Spodnji desni del matrike koeficientov je del, kjer moramo prišteti še prispevek matrike varianc in kovarianc za skupno okolje v čredi (vrstica 4 v algoritmu 1.8). Del se prične v vrstici in stolpcu ns+1 matrike koeficientov in se zaključi v zadnji vrstici in stolpcu (no).

$$\begin{array}{cccccc} & \vdots & \mathbf{X} & & \mathbf{Z}_r & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{X}^T & \vdots & \mathbf{X}^T\mathbf{X} & & \mathbf{X}^T\mathbf{Z}_r & \\ \mathbf{Z}_r^T & \vdots & \mathbf{Z}_r^T\mathbf{X} & & \mathbf{Z}_r^T\mathbf{Z}_r & {\color{red}+I_r\alpha} \end{array} \qquad \text{... (1.15)}$$

V vrstici 5 smo dobili ocene parametrov za vplive v sistematskem delu modela in napovedi parametrov za naključne vplive tako, da smo posplošeno inverzo matrike koeficientov pomnožili z desno stranjo enačbe.

Algoritem 1.7 Kreiranje potrebnih matrik za mešani model

```
/* ................................................................................  */
X    = mi||XP||XS;    /* --------------- matrika dogodkov za sistematski del modela  1. */
ns   = ncol(X);  /* -------------------- število parametrov oz. stolpcev v matriki X 2. */
SisList = “mi”||unique(pasma)||unique(char(spol)); /*seznam paramet. za sist. vplive 3. */
/* =================================================================================== */
/*                     tXX   tXZr       *  beta =    tXy                              */
/*                    tZrX   tZrZr + iGr     u      tZry                              */
/* =================================================================================== */
Zr        = design(rejec); /* ---------------- matrika dogodkov za naključne vplivi   4. */
RejList   = unique(rejec); /* ------------- v vektorju RejList dobimo seznam rejcev   5. */
nrej      = ncol(RejList); /* ------------------------------------- število rejcev   6. */
/* =================================================================================== */
/*          inverzna matrika varianc in kovarianc za enostavni naključni vpliv        */
/* =================================================================================== */
/*  iGr = inv(I(nrej)*sigma2_r)*sigma2_e = I(nrej)*sigma2_e/sigma2_r = I(nrej)* alfa_r  */
/* ................................................................................  */
sigma2_V  = var(RojMasa); /* ----------------------- približek fenotipske variance  7. */
sigma2_r  = sigma2_V * 0.20; /* ------------ b2 = 0.20 delež od fenotipske variance  8. */
sigma2_e  = sigma2_V - sigma2_r; /* -------------------------- varianca za ostanek  9. */
alfa_r    = sigma2_e / sigma2_r; /* ----------------------------- razmerje varianc 10. */
/* ................................................................................  */
iGr = I(nrej)*alfa_r ; /*  inverzna matrika (ko)varianc za skupno okolje pri rejcu  11. */
```

Algoritem 1.8 Mešani model z enostavnim naključnim vplivom

```
/* =================================================================================== */
/*                     tXX   tXZr       *  beta =    tXy                              */
/*                    tZrX   tZrZr + iGr     u      tZry                              */
/* =================================================================================== */
/*                                  Možnost I                                         */
/* =================================================================================== */
XZ        = X||Zr;     /* združeni matriki dogodkov za sistematski in naključni del  1. */
no        = ncol(XZ); /* ------------------------------------------- število rejcev  2. */
MatKoef   = t(XZ)*XZ;  /* -------------- prispevek podatkov k matriki koeficientov  3. */
/* --------------------------------------------- matriki koeficientov dodamo še iGr    */
MatKoef[ns+1:no,ns+1:no]  = MatKoef[ns+1:no,ns+1:no] + iGr; /* -------------------- 4. */
betau     = ginv(MatKoef)*t(XZ)*RojMasa; /* -------------------- ocene in napovedi  5. */
/* =================================================================================== */
/*                                  Možnost II                                        */
/* =================================================================================== */
tXX        = t(X)*X;  /* ----------------------------------------- 1. delna matrika  6. */
tXZr       = t(X)*Zr; /* ----------------------------------------- 2. delna matrika  7. */
tZrX       = t(tXZr); /* ----------------------------------------- 3. delna matrika  8. */
tZrZr      = t(Zr)*Zr + iGr; /* ---------------------------------- 4. delna matrika  9. */
Blok1      = tXX  || tXZr; /* -------------------- spenjamo 1. in 2. delno matriko 10. */
Blok2      = tZrX || tZrZr; /* ------------------- spenjamo 3. in 4. delno matriko 11. */
MatKoef    = Blok1//Blok2 ;  /* ----------------------------- matrika koeficientov 12. */
RHS        = (t(X)*RojMasa) // (t(Zr)*RojMasa); /* ------ desna stran sistema enačb 13. */
betau      = ginv(MatKoef)*RojMasa; /* ------------------------ ocene in napovedi  14. */
```

Algoritem 1.9 Kreiranje matrike dogodkov za vpliv žival

```
/* ..................................................................................  */
X    = mi||XP||XS||XPS;    /* ------------------- matrika dogodkov za sistematski del modela  1. */
ns   = ncol(X);  /* ----------------------------- število parametrov oz. stolpcev v matriki X 2. */
interPS = strip(char(spol))||strip(pasma);  /* --------------- poimenujmo nivoje interakcije  3. */
SisList = "mi"
       // unique(pasma)
       // unique(char(spol))
       // unique(interPS);  /* ------------------------------------------------ seznam paramet. 4. */
/* ================================================================================================ */
/*            tXX    tXZr         tXZa         *  beta =    tXy                                   */
/*            tZrX   tZrZr + iGr  tZrZa            ur       tZry                                  */
/*            tZrX   tZrZr        tZaZa + iGa      ua       tZay                                  */
/* ================================================================================================ */
read all var  {oce}  where oce  ^= . into {sire}; /*------------------ preberemo znane očete  5. */
read all var  {mati} where mati ^= . into {dam} ; /*------------------ preberemo znane očete  5. */
nivo      = unique(zival); /*-------------- v vektorju nivo dobimo seznam živali z meritvami  5. */
nz        = ncol(nivo); /*---------------------------------------- število živali z meritvami  5. */
AList     = unique(zival//dam//sire); /*-------------- v vektorju AList dobimo seznam živali  5. */
na        = ncol(AList); /* ----------------------------------------------- število vseh živali  6. */
/* ..................................................................................  */
Za   = design(zival)||j(nv, na,0); /* --------------- samo, če so sorodniki brez meritev zadaj 4. */
/* ..................................................................................  */
/*         kode živali z meritvami in sorodniki pomešani in vrinemo stolpce z ničlami          */
/* ..................................................................................  */
C1   = design(zival); /* ------------------ stolpci v matriki dogodkov za živali z meritvami  4. */
Za   = j(nv, na,0);    /* ---------- oblikujemo matriko dogodkov in napolnimo s samimi ničlami 4. */
do i  = 1 to na by 1; /* ---------------------------- pregledamo vse stolpce živali z meritvami 6. */
    j   = nivo[i];  /* ------------------------- v katerem stolpcu mora žival z meritvami biti 6. */
    Za[1:nv,j:j] =  C1[1:nv,i:i];  /* ---------------- postavi stolpec za žival na pravo mesto 6. */
end;
/*    ko je delo končano, dobimo v matriki dogodkov še stolpce sorodnikov brez meritev          */
```

1.4.1.2 Nastavitev posameznih delnih matrik pri matriki koeficientov za metodo mešanega modela

Druga možnost je, da nastavimo posamezne delne matrike za matriko koeficientov in delne vektorje za desno stran sistema enačb mešanega modela. V vrsticah od 6 do 9 smo nastavili posamezne delne matrike koeficientov. V vrsticah 10 do 12 smo delne matrike sestavili v matriko koeficientov. V vrstici 13 smo sestavili oba delna vektorja desne strani enačbe: prvi del je vektor za sistematski del modela, drugi del pa je vektor za naključni vpliv. V vrstici 14 smo sistem enačb rešili in v vektorju dobili ocene parametrov za sistematski del modela in napovedi za naključni del modela.

1.4.2 Enolastnostni modeli s plemensko vrednostjo

Modelu (enačba 1.16) smo dodali vpliv živali, da bi dobili plemenske vrednosti za živali. V sistematskem delu pa smo dodali interakcijo med pasmo in sezono.

$$y_{ijkl} = \mu + P_i + S_j + PS_{ij} + r_k + a_{ijkl} + e_{ijkl} \qquad \text{... (1.16)}$$

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Z}_r & & \mathbf{X}^T\mathbf{Z}_a & \\ \mathbf{Z}_r^T\mathbf{X} & \mathbf{Z}_r^T\mathbf{Z}_r & +\mathbf{I}_r\alpha_r & \mathbf{Z}_r^T\mathbf{Z}_a & \\ \mathbf{Z}_a^T\mathbf{X} & \mathbf{Z}_a^T\mathbf{Z}_r & & \mathbf{Z}_a^T\mathbf{Z}_a & +\mathbf{A}\alpha_a \end{bmatrix} \begin{bmatrix} \widehat{\beta} \\ \widehat{\mathbf{u}}_r \\ \widehat{\mathbf{u}}_a \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}_r^T\mathbf{y} \\ \mathbf{Z}_a^T\mathbf{y} \end{bmatrix} \qquad \text{... (1.17)}$$

1.4.3 Večlastnostni modeli

1.5 Naloge

Podatki za naloge: mapa 'M:\Oddelek za zootehniko\Pedag\MKovac\BiomI_podatki'

Koda za vaje ali naloge: mapa 'M:\Oddelek za zootehniko\Pedag\MKovac\BiomI\'

Naloga 1

Podatki "klavne_KP.xls" so iz poskusa pri krškopoljskem prašiču, kjer smo spremljali klavne lastnosti. Zanima nas, kako skupina in spol vplivata na maso toplih polovic.

Algoritem 1.10 Kreiranje matrike genetskih varianc in kovarianc

```
/* ========================================================================================== */
/*            inverzna matrika varianc in kovarianc za enostavni naključni vpliv             */
/* ========================================================================================== */
/*      iGr = inv(I(nrej)*sigma2_r)*sigma2_e = I(nrej)*sigma2_e/sigma2_r = I(nrej)* alfa_r    */
/* .......................................................................................... */
sigma2_V  = var(RojMasa); /* ------------------------------ približek fenotipske variance  7. */
sigma2_r  = sigma2_V * 0.20; /* -------------------- b2 = 0.2 delež od fenotipske variance  8. */
sigma2_a  = sigma2_V * 0.25; /* ------------------------------- h2 = 0.25 dednostni delež  8. */
sigma2_e  = sigma2_V - sigma2_r - sigma2_a; /*------------------------ varianca za ostanek  9. */
alfa_a    = sigma2_e / sigma2_a; /* ------------------------------------ razmerje varianc 10. */
/* .......................................................................................... */
/*                                     matrika sorodstva                                      */
/* .......................................................................................... */
/*              vnesemo zgornjo trikotno matriko in polovico na diagonali                 10. */
Au = {0.5  0.5  0.25  0.25   0   0   0     0    0   0    0   0   0.5  0    0    0    0.5   0,
      0    0.5  0.25  0.25   0   0   0     0    0   0    0   0   0.5  0    0    0    0.5   0,
      0    0    0.5   0.5    0   0   0     0    0   0    0   0   0    0.5  0    0    0.5   0,
      0    0    0     0.5    0   0   0     0    0   0    0   0   0    0.5  0    0    0.5   0,
      0    0    0     0     0.5 0.5 0.25 0.25   0   0    0   0   0    0    0.5  0    0     0.5,
      0    0    0     0      0  0.5 0.25 0.25   0   0    0   0   0    0    0.5  0    0     0.5,
      0    0    0     0      0   0  0.5  0.25   0   0    0   0   0    0    0    0.5  0     0.5,
      0    0    0     0      0   0   0   0.5    0   0    0   0   0    0    0    0.5  0     0.5,
      0    0    0     0      0   0   0     0   0.5  0    0   0   0    0    0    0    0     0,
      0    0    0     0      0   0   0     0    0  0.5   0   0   0    0    0    0    0     0,
      0    0    0     0      0   0   0     0    0   0   0.5  0   0    0    0    0    0     0,
      0    0    0     0      0   0   0     0    0   0    0  0.5  0    0    0    0    0     0,
      0    0    0     0      0   0   0     0    0   0    0   0   0.5  0    0    0    0     0,
      0    0    0     0      0   0   0     0    0   0    0   0   0    0.5  0    0    0     0,
      0    0    0     0      0   0   0     0    0   0    0   0   0    0    0.5  0    0     0,
      0    0    0     0      0   0   0     0    0   0    0   0   0    0    0    0.5  0     0,
      0    0    0     0      0   0   0     0    0   0    0   0   0    0    0    0    0.5   0,
      0    0    0     0      0   0   0     0    0   0    0   0   0    0    0    0    0     0.5};
A  = Au + t(Au);   /* ----------- seštejemo zgornjo trikotno matriko in njeno transponirano 11. */
/* .......................................................................................... */
iGa = A * alfa_a ; /* ---------------------------- inverzna matrika  genetskih (ko)varianc  11. */
```

- Zapišite model v skalarni in matrični obliki!
- Uvozite podatke v SAS!
- Preberite podatke v matrike (vektorje)!
- V SAS/IML nastavite vektor opazovanj y in matriko X za vaš model!
- Izračunajte ocene parametrov za sistematske vplive!

Naloga 2

V podatkih "ABwt.dat" sta vpliva A (spol) in B (krma) ter telesna masa ob koncu poskusa kot opazovana lastnost.

- Za telesno maso sestavite model in nastavite ter rešite sistem enačb!

Naloga 3

Stolpci v podatkih "krm_pos.xls" so genotip, krma spol, dnevni prirast (dm, v g/dan) in delež mesa (dm, v %).

- Za dnevni prirast sestavite model in nastavite ter rešite sistem enačb!
- Nastavite sistem enačb še za dvolastnostni model (dp in dm)!