Odhad plemenné hodnoty pomocí sire modelu (otcovského modelu)

Př.:

Z populace jsou náhodně vybráni otcové a každý byl náhodně pářen se samicemi. Z každého páření byli sledováni potomci. Sledujeme několik potomků, kteří byli chováni v různých chovech (**Viz zadání**). Model předpokládá, že není interakce mezi otcem a chovem. **Chceme odhadnout PH otců podle užitkovosti jejich dcer (Sire model).**

$$y_{ijk} = b_i + u_j + e_{ijk}$$

Náhodný efekt – efekt j-tého otce (3 úrovně); pevný efekt – efekt i-tého stáda (2 úrovně)

Potomek	otec	chov	užitkovost
1	1	1	9
2	1	2	12
3	2	1	11
4	2	1	6
5	3	1	7
6	3	2	14

Smíšený model: **Y** = **Xb** + **Zu** + **e**

$$\mathbf{y} = \begin{bmatrix} y_{111} \\ y_{121} \\ y_{211} \\ y_{212} \\ y_{311} \\ y_{321} \end{bmatrix} = \begin{bmatrix} 9 \\ 12 \\ 11 \\ 6 \\ 7 \\ 14 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} e_{111} \\ e_{121} \\ e_{211} \\ e_{212} \\ e_{311} \\ e_{321} \end{bmatrix}$$

Předpoklady: - E(**u**) = E(**e**) = 0, V(**u**) = **G**, V(**e**) = **R**

- kovarianční matice pro vektor pozorování **y** je **V = ZGZ` + R**
- reziduální chyby mají konstantní variance a jsou nekorelovány $\Rightarrow$ **R** = **I** σ_e^2

Ve smíšeném modelu: pozorujeme **y**, **X**, **Z** zatímco **b**, **R** a **G** jsou obecně neznámé.

> Pro řešení odhadů pevných efektů se používá procedura BLUE a pro odhad náhodných efektů BLUP. Odhady jsou *nejlepší* v tom smyslu, že minimalizují výběrovou varianci, *lineární*, že jsou lineární funkcí pozorovaných fenotypů **y**, a nevychýlené, že E[BLUE(**b**)] = **b** a E[BLUP(**u**)] = **u**.

Chceme zjistit plemenné hodnoty otců (u_1, u_2, u_3)?

1. Výpočet vyžaduje variančně-kovarianční matici pro otce (**G**) a reziduální (**R**) .
   a. Zde **R** = **I** σ_E^2 **I**$_{(6)}$
   b. otcové jsou nepříbuzní – **G** = **I** σ_O^2 **I**$_{(3)}$
2. Za předpokladu jen aditivní genetické variance – efekt otců (PH) je polovina otcovské aditivní genetické hodnoty - $\sigma_O^2 = \sigma_A^2/4$ (σ_A^2 - aditivní genetická variance).
   - zadáme si, že $\sigma_A^2 = 8$ a $\sigma_E^2 = 6$

3. Variančně kovarianční matice **V** pro vektor **y** je dána **V** = **ZGZ`** + **R**.

Smíšený model: **Y** = **Xb** + **Zu** + **e**

1. řešení:

BLUE $\mathbf{b} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{y})$

BLUP $\mathbf{u} = (\mathbf{G}\mathbf{Z}'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\mathbf{b})$

2. řešení: normální rovnice smíšeného modelu:

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}$$

Vyřešte oba způsoby pomocí programu R!

prof. Ing. Tomáš Urban, Ph.D.

UMFGZ
MENDELU v Brně
urban@mendelu.cz

Otcovský (sire) model

```
> y <- matrix(c(9,12,11,6,7,14),6,1)
> y
     [,1]
[1,]    9
[2,]   12
[3,]   11
[4,]    6
[5,]    7
[6,]   14
> X <- matrix(c(1,0,1,1,1,0,
+               0,1,0,0,0,1),6,2)
> X
     [,1] [,2]
[1,]    1    0
[2,]    0    1
[3,]    1    0
[4,]    1    0
[5,]    1    0
[6,]    0    1
> Z <- matrix(c(1,1,0,0,0,0,
+               0,0,1,1,0,0,
+               0,0,0,0,1,1),6,3)
> Z
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    1    0    0
[3,]    0    1    0
[4,]    0    1    0
[5,]    0    0    1
[6,]    0    0    1
> sigma2e <- 6
```

```
> R <- diag(6)*sigma2e
> R
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    6    0    0    0    0    0
[2,]    0    6    0    0    0    0
[3,]    0    0    6    0    0    0
[4,]    0    0    0    6    0    0
[5,]    0    0    0    0    6    0
[6,]    0    0    0    0    0    6
> G <- diag(3)*(8/4)
> G
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    2    0
[3,]    0    0    2
> V <- Z %*% G %*% t(Z) + R
> V
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    8    2    0    0    0    0
[2,]    2    8    0    0    0    0
[3,]    0    0    8    2    0    0
[4,]    0    0    2    8    0    0
[5,]    0    0    0    0    8    2
[6,]    0    0    0    0    2    8
> invV <- solve(V)
> invV
            [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
[1,]  0.13333333 -0.03333333  0.00000000  0.00000000  0.00000000  0.00000000
[2,] -0.03333333  0.13333333  0.00000000  0.00000000  0.00000000  0.00000000
[3,]  0.00000000  0.00000000  0.13333333 -0.03333333  0.00000000  0.00000000
[4,]  0.00000000  0.00000000 -0.03333333  0.13333333  0.00000000  0.00000000
[5,]  0.00000000  0.00000000  0.00000000  0.00000000  0.13333333 -0.03333333
[6,]  0.00000000  0.00000000  0.00000000  0.00000000 -0.03333333  0.13333333
```

```
> b <- solve(t(X) %*% invV %*% X) %*% (t(X) %*% invV %*% y)
> b
          [,1]
[1,]  8.222222
[2,] 13.055556
> u <- (G %*%Z %*% invV) %*% (y - (X%*%b))
Error in G %*% Z : non-conformable arguments
> u <- (G %*% t(Z) %*% invV) %*% (y - (X%*%b))
> u
            [,1]
[1,] -0.05555556
[2,]  0.11111111
[3,] -0.05555556
```

BLUP - Pomocí soustavy normálních rovnic:

```
> XRX <- t(X) %*% solve(R) %*% X
> XRX
          [,1]      [,2]
[1,] 0.6666667 0.0000000
[2,] 0.0000000 0.3333333
> XRZ <- t(X) %*% solve(R) %*% Z
> XRZ
          [,1]      [,2]      [,3]
[1,] 0.1666667 0.3333333 0.1666667
[2,] 0.1666667 0.0000000 0.1666667
> ZRX <- t(Z) %*% solve(R) %*% X
> ZRX
          [,1]      [,2]
[1,] 0.1666667 0.1666667
[2,] 0.3333333 0.0000000
[3,] 0.1666667 0.1666667
> ZRZ <- t(Z) %*% solve(R) %*% Z
```

```
> ZRZ
          [,1]      [,2]      [,3]
[1,] 0.3333333 0.0000000 0.0000000
[2,] 0.0000000 0.3333333 0.0000000
[3,] 0.0000000 0.0000000 0.3333333
> ZRZG <- ZRZ + solve(G)
> ZRZG
          [,1]      [,2]      [,3]
[1,] 0.8333333 0.0000000 0.0000000
[2,] 0.0000000 0.8333333 0.0000000
[3,] 0.0000000 0.0000000 0.8333333
> LS1 <- cbind(XRX,XRZ)
> LS1
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.6666667 0.0000000 0.1666667 0.3333333 0.1666667
[2,] 0.0000000 0.3333333 0.1666667 0.0000000 0.1666667
> LS2 <- cbind(ZRX,ZRZG)
> LS2
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.1666667 0.1666667 0.8333333 0.0000000 0.0000000
[2,] 0.3333333 0.0000000 0.0000000 0.8333333 0.0000000
[3,] 0.1666667 0.1666667 0.0000000 0.0000000 0.8333333
> LS <- rbind(LS1,LS2)
> LS
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.6666667 0.0000000 0.1666667 0.3333333 0.1666667
[2,] 0.0000000 0.3333333 0.1666667 0.0000000 0.1666667
[3,] 0.1666667 0.1666667 0.8333333 0.0000000 0.0000000
[4,] 0.3333333 0.0000000 0.0000000 0.8333333 0.0000000
[5,] 0.1666667 0.1666667 0.0000000 0.0000000 0.8333333
> XRy <- t(X) %*% solve(R) %*% y
> XRy
         [,1]
```

```
[1,] 5.500000
[2,] 4.333333
> ZRy <- t(Z) %*% solve(R) %*% y
> ZRy
         [,1]
[1,] 3.500000
[2,] 2.833333
[3,] 3.500000
> PS <- rbind(XRy, ZRy)
> PS
         [,1]
[1,] 5.500000
[2,] 4.333333
[3,] 3.500000
[4,] 2.833333
[5,] 3.500000
> bu <- solve(LS) %*% PS
> bu
            [,1]
[1,]  8.22222222
[2,] 13.05555556
[3,] -0.05555556
[4,]  0.11111111
[5,] -0.05555556
```