Emhimad A E Abdalla

Obtain Estimated Breeding Values (EBV) at all time points given RR coefficients from a random regression analysis:

Fitting a random regression random effect yields random coefficients. These coefficients can be used with the corresponding polynomials to obtain the estimates of that random regression effect over all time intervals. One of the most famous random regressions is the solutions (EBV) for animal effect.

The general approach as follows (matrix multiplication): Legendre polynomials **x** regression coefficient

Example: Assume the animal effect was fitted as a second order, and we obtained the following RR coefficients for a given animal:

Coefficients_1	Coefficients_2	Coefficients_3
6.33	4.08	-1.83

Assume also we have 5 times. The following Legendre polynomials were used to obtain the coefficients above:

```
0.7071068 -1.2247449  1.5811388
0.7071068 -0.6123724 -0.1976424
0.7071068  0.0000000 -0.7905694
0.7071068  0.6123724 -0.1976424
0.7071068  1.2247449  1.5811388
```

Now, to obtain EBV at first time, we multiply each coefficient by its corresponding Legendre polynomials at that time point:

```
(6.33 * 0.7071068) + (4.08 * -1.2247449) + (-1.83 *  1.5811388) = -3.414457
(6.33 * 0.7071068) + (4.08 * -0.6123724) + (-1.83 * -0.1976424) =  2.339194
(6.33 * 0.7071068) + (4.08 *  0.0000000) + (-1.83 * -0.7905694) =  5.922729
(6.33 * 0.7071068) + (4.08 *  0.6123724) + (-1.83 * -0.1976424) =  7.336150
(6.33 * 0.7071068) + (4.08 *  1.2247449) + (-1.83 *  1.5811388) =  6.579463
```

Using matrix multiplication, this step could be easier:

```
> EBVs = LegendrePolynomials %*% coefficients
> EBVs
          [,1]
[1,] -3.414457
[2,]  2.339194
[3,]  5.922729
[4,]  7.336150
[5,]  6.579463
```

We can see if we are interested in the sum of EBVs over all time points, we just to multiply the sum of Legendre polynomials by the EBV coefficients:

```
      0.7071068 -1.2247449  1.5811388
      0.7071068 -0.6123724 -0.1976424
      0.7071068  0.0000000 -0.7905694
      0.7071068  0.6123724 -0.1976424
      0.7071068  1.2247449  1.5811388
Sum = 3.5355350  0.0000000  1.9764250
```

(6.33 * 3.535535) + (4.08 * 0.000000) + (-1.83 * 1.976425) = **18.76308**

Which is the sum of the EBVs at each time point:

```
           [,1]
[1,] -3.414457
[2,]  2.339194
[3,]  5.922729
[4,]  7.336150
[5,]  6.579463
Sum = 18.763080
```

A more efficient way to calculate EBV for a large number on animals:

Using the matrix multiplication as shown above, it could be slow if the number of animals is large and that is because we need to loop over animals. In each round, we calculate the breeding values for an animal at all time points. The other option, which is much faster, is to loop over the time points instead of animals. Most of the time, the number of time points are smaller than the number of animals.

Assume, we have RR EBV coefficients for 3 animals:

Coefficients_1	Coefficients_2	Coefficients_3
6.33	4.08	-1.83
3.22	2.55	1.50
8.56	4.99	-2.01

1. The first approach to obtain EBVs over time for each animal is by looping over the animals, as follows:

```
> # Assume we have
> animals <- matrix(c(6.33, 4.08,-1.83,
+                     3.22, 2.55, 1.50,
+                     8.56, 4.99, -2.01), byrow = T, ncol = 3)
>
> for (i in 1:3){
+   cat('EBV animal',i,':', LegendrePolynomials  %*% animals[i,1:3],'\n')
+ }

EBV animal 1: -3.414457 2.3391940 5.922729 7.336150 6.579463
EBV animal 2:  1.525493 0.4188729 1.091031 3.541970 7.771693
EBV animal 3: -3.236731 3.3943600 7.641880 9.505833 8.986224
```

2. A much faster approach, especially if the number of animals is so large, is to loop over time points and only multiply columns. This can be easily done by R or Python for example.

Coeff 1	Coeff_2	Coeff_3	EBV at time 1
6.33	4.08	-1.83	= 6.33*0.7071068 + 4.08*-1.2247449 + -1.83 * 1.5811388 = -3.414457
3.22	2.55	1.50	= 3.22*0.7071068 + 2.55*-1.2247449 + 1.5 * 1.5811388 = 1.525493
8.56	4.99	-2.01	= 8.56*0.7071068 + 4.99*-1.2247449 + -2.01 * 1.5811388 = -3.236732

Emhimad A E Abdalla

```
> animals <- matrix(c(6.33, 4.08,-1.83,
+                     3.22, 2.55, 1.50,
+                     8.56, 4.99, -2.01), byrow = T, ncol = 3)
>
>
> animals <- as.data.frame(animals)
> colnames(animals) <- c('coeff1','coeff2','coeff3')
>
> for (i in 1:5){
+  animals[[paste0('EBV_',i)]] = animals[1:3,1] * LegendrePolynomials[i,1] +
animals[1:3,2] * LegendrePolynomials[i,2] + animals[1:3,3] *
LegendrePolynomials[i,3]
+
+ }
>
>
> animals
  coeff1 coeff2 coeff3     EBV_1     EBV_2    EBV_3    EBV_4    EBV_5
1   6.33   4.08  -1.83 -3.414457 2.3391944 5.922729 7.336150 6.579463
2   3.22   2.55   1.50  1.525493 0.4188729 1.091031 3.541970 7.771693
3   8.56   4.99  -2.01 -3.236731 3.3943601 7.641880 9.505833 8.986224
```