Obtain parameters (e.g., EBV, variance, covariance, correlations …) at all time points given RR coefficients from a random regression analysis:

In general, RR coefficients can be transformed to over-time estimates (time function) using the polynomials used to get those coefficients.

Normally, these RR coefficients are obtained based on arrange of time points. For example, ages in days: 1,3,4,6,7,8. But, we could also predict the values where we do not have time points. For example, we could predict the estimates at ages 2,5. However, it is important that those missing time points fall within the range of the observed range (i.e., 1 to 8). In other words, **we should not use these RR coefficients for predicting time periods outside the observed range**.

A. RR coefficients without covariance:

Such as RR coefficients for estimated breeding values. In this case, we would have a number of coefficients responding to the order of fit without covariance.

For example, we got these RR coefficients estimates for animal effect (EBV):

RR coefficients	1	2	3
Estimates	6.33	4.08	-1.83

The EBV at each time point can be obtained via matrix multiplication:

Legendre polynomials **x** the regression coefficient

Using the code EBV_overTime.R or EBV_overTime.py,

```
> #=========================================================================#
> # Read Legendre Polynomials
> LegendrePolynomials = matrix( c(
+  0.707107,     -1.224745,      1.581139,
+  0.707107,     -0.612372,     -0.197642,
+  0.707107,      0.000000,     -0.790569,
+  0.707107,      0.612372,     -0.197642,
+  0.707107,      1.224745,      1.581139
+  ), nrow=5, byrow = TRUE)
> #=========================================================================#
> # EBVs at each time point:
> coeff=matrix( c(6.33, 4.08,-1.83))
> coeff
      [,1]
[1,]  6.33
[2,]  4.08
[3,] -1.83
> EBVs = LegendrePolynomials  %*% coeff
> EBVs
          [,1]
[1,] -3.414457
[2,]  2.339194
[3,]  5.922729
[4,]  7.336150
[5,]  6.579463
> #=========================================================================#
```

B. RR coefficients with covariance:

The general approach as follows (matrix multiplication):

(Legendre polynomials **x** the regression coefficient **x** the transpose of the Legendre polynomials)

1. **Single trait**:

For a single trait, it is quite simple because we do not have covariances between multiple traits. Note: the time points that have been used to obtain those variance components should be known as well.

Assume we would like to get variances and covariances (for the single trait) over time (File: 3_RR_coefficients_for_1_traits.txt):

RR coefficients	1	2	3
1	3.5	0.8	0.5
2	0.8	1.3	0.4
3	0.5	0.4	1

The estimates were obtained based on 5 time points, then the Legendre polynomials are:

```
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
```

Using the R code `(3_RR_coefficients_for_1_traits.R)` or the function `(RRcoeff_to_varComp_function.R)`, which is flexible for any number of traits, we get:

```
varCovr
          [,1]      [,2]      [,3]     [,4]       [,5]
[1,] 4.3832000 1.5719356 0.4739866 1.089353  3.4180339
[2,] 1.5719356 1.5408126 1.4041035 1.161808  0.8139272
[3,] 0.4739866 1.4041035 1.8159831 1.709625  1.0850307
[4,] 1.0893526 1.1618084 1.7096255 2.732804  4.2313441
[5,] 3.4180339 0.8139272 1.0850307 4.231344 10.2528681

#The additive genetic variance:
[1]  4.383200  1.540813  1.815983  2.732804 10.252868

correlation
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.6048730 0.1680020 0.3147521 0.5098683
[2,] 0.6048730 1.0000000 0.8393987 0.5661811 0.2047802
[3,] 0.1680020 0.8393987 1.0000000 0.7674337 0.2514568
[4,] 0.3147521 0.5661811 0.7674337 1.0000000 0.7993766
[5,] 0.5098683 0.2047802 0.2514568 0.7993766 1.0000000
```

2. Multiple traits

The same approach used for the single trait could be used here, but we need to do the same procedure for the covariances between the traits.

Note: The function "RRcoefToVarComponent" in file "RRcoeff_to_varComp_function.R" is much easier to use and it is fully flexible for any number of traits and RR coefficients.

Assume we would like to get variances and covariances (within traits and between traits) for two traits over time (File: 3_RR_coefficients_for_2_traits.txt):

		Trait 1			Trait 2		
	RR coefficients	1	2	3	1	2	3
Trait 1	1	3.5	0.8	0.5	0.8	0.9	1.5
	2	0.8	1.3	0.4	0.2	1.6	0.25
	3	0.5	0.4	1	0.5	5	-1.5
Trait 2	1	0.8	0.2	0.5	2	0.5	-2
	2	0.9	1.6	5	0.5	3	2
	3	1.5	0.25	-1.5	-2	2	5

To make simple to understand, converting the RR coefficients to variances/covariances should be done for each of the blocks in the table above and then we could combine them up to get the covariances tabulated and also we could obtain correlations.

		Trait 1			Trait 2		
	RR coefficients	1	2	3	1	2	3
Trait 1	1	Block 1			Block 2		
	2						
	3						
Trait 2	1	Block 2			Block 3		
	2						
	3						

From the R code (3_RR_coefficients_for_2_traits.R):

```
> #=============================================================================#
> Phi = as.matrix(read.table("Legendre_polynomials.txt"))
> H = as.matrix(read.table("3_RR_coefficients_for_2_traits.txt", header =F))
> #=============================================================================#
>
> H1 = H[1:3,1:3] # Block 1
> H1
     V1  V2  V3
[1,] 3.5 0.8 0.5
[2,] 0.8 1.3 0.4
[3,] 0.5 0.4 1.0
> H2 = H[1:3,4:6] # Block 2, same as transpose of H[4:6,1:3]
> H2
     V4  V5    V6
[1,] 0.8 0.9  1.50
[2,] 0.2 1.6  0.25
[3,] 0.5 5.0 -1.50
> H3 = H[4:6,4:6] # Block 3
> H3
      V4  V5 V6
[1,]  2.0 0.5 -2
[2,]  0.5 3.0  2
[3,] -2.0 2.0  5
>
> HH1 = Phi %*%H1 %*% t(Phi)
> HH1
          [,1]      [,2]      [,3]     [,4]       [,5]
[1,] 4.3832000 1.5719356 0.4739866 1.089353  3.4180339
[2,] 1.5719356 1.5408126 1.4041035 1.161808  0.8139272
[3,] 0.4739866 1.4041035 1.8159831 1.709625  1.0850307
[4,] 1.0893526 1.1618084 1.7096255 2.732804  4.2313441
[5,] 3.4180339 0.8139272 1.0850307 4.231344 10.2528681
> HH2 = Phi %*%H2 %*% t(Phi)
> HH2
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  -9.833141 -2.9254943  2.0643478  5.1363863  6.2906213
[2,]   3.778145  0.8209951 -0.7083495 -0.8098894  0.5163753
[3,]   7.734349  1.7073880 -1.6555339 -2.3544179 -0.3892639
[4,]   2.035473 -0.2663151 -0.7772059  0.5028001  3.5737032
[5,] -13.318485 -5.1001148  1.9266351  7.7617656 12.4052776
> HH3 = Phi %*%H3 %*% t(Phi)
> HH3
          [,1]       [,2]      [,3]       [,4]      [,5]
[1,]  4.915872 -2.3709473 -4.864555 -2.5649514  4.527863
[2,] -2.370947  2.9304400  3.930532  0.6293299 -6.973168
[3,] -4.864555  3.9305322  6.361068  2.4270534 -7.871513
[4,] -2.564951  0.6293299  2.427053  2.8282193  1.832828
[5,]  4.527863 -6.9731678 -7.871513  1.8328278 22.139856
>
```

```
> temp1 = cbind(HH1,HH2)
> temp1
          [,1]      [,2]      [,3]     [,4]       [,5]       [,6]       [,7]       [,8]       [,9]       [,10]
[1,] 4.3832000 1.5719356 0.4739866 1.089353  3.4180339  -9.833141 -2.9254943  2.0643478  5.1363863  6.2906213
[2,] 1.5719356 1.5408126 1.4041035 1.161808  0.8139272   3.778145  0.8209951 -0.7083495 -0.8098894  0.5163753
[3,] 0.4739866 1.4041035 1.8159831 1.709625  1.0850307   7.734349  1.7073880 -1.6555339 -2.3544179 -0.3892639
[4,] 1.0893526 1.1618084 1.7096255 2.732804  4.2313441   2.035473 -0.2663151 -0.7772059  0.5028001  3.5737032
[5,] 3.4180339 0.8139272 1.0850307 4.231344 10.2528681 -13.318485 -5.1001148  1.9266351  7.7617656 12.4052776
> temp2 = cbind(t(HH2),HH3) # Make sure to transpose Block 2
> temp2
          [,1]      [,2]      [,3]      [,4]       [,5]      [,6]       [,7]       [,8]       [,9]      [,10]
[1,] -9.833141  3.7781447  7.7343487  2.0354727 -13.318485  4.915872 -2.3709473 -4.864555 -2.5649514  4.527863
[2,] -2.925494  0.8209951  1.7073880 -0.2663151  -5.100115 -2.370947  2.9304400  3.930532  0.6293299 -6.973168
[3,]  2.064348 -0.7083495 -1.6555339 -0.7772059   1.926635 -4.864555  3.9305322  6.361068  2.4270534 -7.871513
[4,]  5.136386 -0.8098894 -2.3544179  0.5028001   7.761766 -2.564951  0.6293299  2.427053  2.8282193  1.832828
[5,]  6.290621  0.5163753 -0.3892639  3.5737032  12.405278  4.527863 -6.9731678 -7.871513  1.8328278 22.139856
> varCovr = rbind(temp1,temp2)
> varCovr
           [,1]      [,2]      [,3]      [,4]       [,5]       [,6]       [,7]       [,8]       [,9]      [,10]
 [1,]  4.3832000  1.5719356  0.4739866  1.0893526   3.4180339  -9.833141 -2.9254943  2.0643478  5.1363863  6.2906213
 [2,]  1.5719356  1.5408126  1.4041035  1.1618084   0.8139272   3.778145  0.8209951 -0.7083495 -0.8098894  0.5163753
 [3,]  0.4739866  1.4041035  1.8159831  1.7096255   1.0850307   7.734349  1.7073880 -1.6555339 -2.3544179 -0.3892639
 [4,]  1.0893526  1.1618084  1.7096255  2.7328040   4.2313441   2.035473 -0.2663151 -0.7772059  0.5028001  3.5737032
 [5,]  3.4180339  0.8139272  1.0850307  4.2313441  10.2528681 -13.318485 -5.1001148  1.9266351  7.7617656 12.4052776
 [6,] -9.8331410  3.7781447  7.7343487  2.0354727 -13.3184853   4.915872 -2.3709473 -4.8645547 -2.5649514  4.5278634
 [7,] -2.9254943  0.8209951  1.7073880 -0.2663151  -5.1001148  -2.370947  2.9304400  3.9305322  0.6293299 -6.9731678
 [8,]  2.0643478 -0.7083495 -1.6555339 -0.7772059   1.9266351  -4.864555  3.9305322  6.3610679  2.4270534 -7.8715127
 [9,]  5.1363863 -0.8098894 -2.3544179  0.5028001   7.7617656  -2.564951  0.6293299  2.4270534  2.8282193  1.8328278
[10,]  6.2906213  0.5163753 -0.3892639  3.5737032  12.4052776   4.527863 -6.9731678 -7.8715127  1.8328278 22.1398560
> correlation = cov2cor(varCovr)
> correlation
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]       [,8]       [,9]       [,10]
 [1,]  1.0000000  0.60487302  0.16800203  0.31475207  0.5098683 -2.1183434 -0.81627631  0.3909509  1.4588327  0.63857293
 [2,]  0.6048730  1.00000000  0.83939874  0.56618109  0.2047802  1.3727878  0.38636630 -0.2262599 -0.3879666  0.08841037
 [3,]  0.1680020  0.83939874  1.00000000  0.76743366  0.2514568  2.5886157  0.74013326 -0.4870989 -1.0388937 -0.06139045
 [4,]  0.3147521  0.56618109  0.76743366  1.00000000  0.7993766  0.5553423 -0.09410769 -0.1864090  0.1808567  0.45943793
 [5,]  0.5098683  0.20478021  0.25145682  0.79937657  1.0000000 -1.8759959 -0.93044531  0.2385678  1.4413893  0.82337259
 [6,] -2.1183434  1.37278780  2.58861573  0.55534233 -1.8759959  1.0000000 -0.62467658 -0.8699170 -0.6878953  0.43401591
 [7,] -0.8162763  0.38636630  0.74013326 -0.09410769 -0.9304453 -0.6246766  1.00000000  0.9103742  0.2186026 -0.86571772
 [8,]  0.3909509 -0.22625994 -0.48709893 -0.18640897  0.2385678 -0.8699170  0.91037417  1.0000000  0.5722131 -0.66329357
 [9,]  1.4588327 -0.38796655 -1.03889371  0.18085669  1.4413893 -0.6878953  0.21860265  0.5722131  1.0000000  0.23162088
[10,]  0.6385729  0.08841037 -0.06139045  0.45943793  0.8233726  0.4340159 -0.86571772 -0.6632936  0.2316209  1.00000000
```