'rmap' Walkthrough (v.01)

Gail Gong and David Johnston

May 4, 2011

1. The example

```
[1] "################################################################################################ 1"
[1] "d"
  e   t     w     r c k
1 1 1.6 0.180 0.604 A 1
2 0 4.3 0.133 0.515 B 1
3 1 0.8 0.210 0.647 A 1
4 0 0.6 0.318 0.749 B 2
5 1 0.3 0.218 0.657 A 2
6 1 4.0 0.099 0.430 A 1
7 2 1.6 0.152 0.554 B 1
[1] "################################################################################################ 2"
[1] "N"
 A B
4 4
[1] "################################################################################################ 3"
[1] "n"
 A B
4 3
[1] "################################################################################################ 4"
[1] "d_k_equals_1"
  e   t c  aaa k
1 1 1.6 A 1.00 1
2 0 4.3 B 1.33 1
3 1 0.8 A 1.00 1
6 1 4.0 A 1.00 1
7 2 1.6 B 1.33 1
[1] "################################################################################################ 5"
[1] "d_k_equals_2"
  e   t c  aaa k
4 0 0.6 B 1.33 2
5 1 0.3 A 1.00 2
[1] "################################################################################################ 6"
[1] "names(lambdaHat)"
[1] "k1" "k2"
[1] "names(lambdaHat[[\"k1\"]])"
[1] "lambdaHat" "NAR"       "DDD"       "tau"       "denom"
[1] "################################################################################################ 7"
[1] "tau for k = 1"
 $`0.8`
[1] 1

$`1.6`
```

```
[1] 1 2

$`4`
[1] 1
[1] "######################################################################################################### 8"
[1] "tau for k = 2"
 $`0.3`
[1] 1
[1] "######################################################################################################### 9"
[1] "lambdaHat for k = 1"
        tau1  tau2  tau3
event1 0.176 0.214 0.429
event2 0.000 0.286 0.000
[1] "######################################################################################################### 10"
[1] "lambdaHat for k = 2"
        tau1
event1 0.429
event2 0.000
[1] "######################################################################################################### 11"
[1] "NAR for k = 1"
     tau1  tau2  tau3
[1,] TRUE  TRUE FALSE
[2,] TRUE  TRUE  TRUE
[3,] TRUE FALSE FALSE
[4,] TRUE  TRUE  TRUE
[5,] TRUE  TRUE FALSE
[1] "######################################################################################################### 12"
[1] "DDD for k = 1"
[[1]]
      tau1  tau2  tau3
[1,] FALSE  TRUE FALSE
[2,] FALSE FALSE FALSE
[3,]  TRUE FALSE FALSE
[4,] FALSE FALSE  TRUE
[5,] FALSE FALSE FALSE

[[2]]
      tau1  tau2  tau3
[1,] FALSE FALSE FALSE
[2,] FALSE FALSE FALSE
[3,] FALSE FALSE FALSE
[4,] FALSE FALSE FALSE
[5,] FALSE  TRUE FALSE
[1] "######################################################################################################### 13"
[1] "denom for k = 1"
[1] 5.67 4.67 2.33
[1] "#########################################################################################################   14"
[1] "uuu"
      [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7]  [,8]  [,9]
[1,]  1.41 -1.21  4.67  0.00 -1.21  0.0  0.00  0.00  0.00
[2,]  1.41 -1.21 -2.00 -1.75 -1.21 -2.0 -1.75  0.00  0.00
[3,]  1.41  5.67  0.00  0.00  0.00  0.0  0.00  0.00  0.00
[4,]  1.41 -1.21 -2.00  2.33 -1.21 -2.0  0.00  0.00  0.00
[5,]  1.41 -1.21  0.00  0.00 -1.21  3.5  0.00  0.00  0.00
[6,] -3.43  0.00  0.00  0.00  0.00  0.0  0.00 -1.75 -1.75
[7,] -3.43  0.00  0.00  0.00  0.00  0.0  0.00  2.33  0.00
[1] "######################################################################################################### 15"
```

```
[1] "VVV"
      [,1]  [,2]   [,3] [,4] [,5]   [,6] [,7] [,8] [,9]
 [1,] 0.207 0.000  0.000 0.00    0  0.000    0 0.00    0
 [2,] 0.000 0.205  0.000 0.00    0  0.000    0 0.00    0
 [3,] 0.000 0.000  0.289 0.00    0 -0.105    0 0.00    0
 [4,] 0.000 0.000  0.000 0.84    0  0.000    0 0.00    0
 [5,] 0.000 0.000  0.000 0.00    0  0.000    0 0.00    0
 [6,] 0.000 0.000 -0.105 0.00    0  0.350    0 0.00    0
 [7,] 0.000 0.000  0.000 0.00    0  0.000    0 0.00    0
 [8,] 0.000 0.000  0.000 0.00    0  0.000    0 0.84    0
 [9,] 0.000 0.000  0.000 0.00    0  0.000    0 0.00    0
[1] "################################################################################################### 16"
[1] "B2"
       [,1]    [,2]    [,3]   [,4]    [,5]    [,6]   [,7]    [,8]    [,9]
 [1,]  1.302 -0.3265 -0.2689 -0.235 -0.3265  0.2017 -0.235  0.4706  0.4706
 [2,] -0.327  0.0819  0.0675  0.059  0.0819 -0.0506  0.059 -0.1181 -0.1181
 [3,] -0.269  0.0675  0.2222  0.194  0.0675  0.4167  0.194 -0.0972 -0.0972
 [4,] -0.235  0.0590  0.1944  0.170  0.0590  0.3646  0.170 -0.0851 -0.0851
 [5,] -0.327  0.0819  0.0675  0.059  0.0819 -0.0506  0.059 -0.1181 -0.1181
 [6,]  0.202 -0.0506  0.4167  0.365 -0.0506  1.2917  0.365  0.0729  0.0729
 [7,] -0.235  0.0590  0.1944  0.170  0.0590  0.3646  0.170 -0.0851 -0.0851
 [8,]  0.471 -0.1181 -0.0972 -0.085 -0.1181  0.0729 -0.085  0.1701  0.1701
 [9,]  0.471 -0.1181 -0.0972 -0.085 -0.1181  0.0729 -0.085  0.1701  0.1701
[1] "################################################################################################### 17"
[1] "V2Stage"
        [,1]     [,2]     [,3]    [,4] [,5]     [,6] [,7]    [,8] [,9]
 [1,]  0.2622 -0.01384 -0.02041 -0.0408    0  0.02041    0  0.0816    0
 [2,] -0.0138  0.20862  0.00508  0.0102    0 -0.00508    0 -0.0203    0
 [3,] -0.0204  0.00508  0.29613  0.0150    0 -0.11245    0 -0.0300    0
 [4,] -0.0408  0.01017  0.01499  0.9596    0  0.08996    0 -0.0600    0
 [5,]  0.0000  0.00000  0.00000  0.0000    0  0.00000    0  0.0000    0
 [6,]  0.0204 -0.00508 -0.11245  0.0900    0  0.47980    0  0.0300    0
 [7,]  0.0000  0.00000  0.00000  0.0000    0  0.00000    0  0.0000    0
 [8,]  0.0816 -0.02034 -0.02999 -0.0600    0  0.02999    0  0.9596    0
 [9,]  0.0000  0.00000  0.00000  0.0000    0  0.00000    0  0.0000    0
```

Explanations follow

In the discussion to follow, #1# refers to the section headed with a line of hash marks follwed by the number 1.

2. Generate a tiny data set

Begin by generating a tiny data set `d` #1#, consisting of seven observations divided into `KKK = 2` risk groups. This is an artificial data set, doctored so that it has all the features we wish to illustrate but small enough so that the matrices which rmap produces are small enough to fit easily into the output of R.

Notice that this data set was generated using two-stage sampling with numbers of observations in the first stage described by `N` #2#, and the number in the second stage described by `n` #3#.

It will be useful in our thinking to split the data set by `k` #4# and #5# and write down the ordered distinct times `tau` to events `e = 1` and `e = 2` for each of `k = 1` #7# and `k = 2` #8#. Notice that for `k = 1`, there are $M_1 = 3$ distinct event times `tau1 = 0.8`, during which event `e = 1` occurred, `tau2 = 1.6`, during which both `e = 1` and `e = 2` occurred, and `tau3 = 4.0`, during which `e = 1` occurred. For `k = 2`, there was just $M_2 = 1$ distinct event time `tau1 = 0.3` during which `e = 1` occurred. In general M_k for a real data set can become quite large, resulting in very large matrices `uuu`, `VVV`, `B2`, and `V2Stage`.

3. lambdaHatFn

`lambdaHatFn` produces not only `lambdaHat` #9# and #10# but several intermediate quantities that will be useful later in `uuuFn` and `VVVFn`.

NAR

`NAR[n,m]` $= N_{kmn}$ in Equation (29) in "rmap-formulas-v01.pdf" from the rmap website.

`NAR` is a matrix with one row for each of the observations in this risk group and one column for each `tau`. `NAR[n,m]` is TRUE if $t_n \geq \tau_m$, that is, if the person in the `n` th row (of `d_k_equals_1`) is at risk at time τ_m. #11# Shows `NAR` for `k = 1`. Since $\tau_1 = 0.8$ and every person in risk group `k = 1` #4# has survival time $t \geq \tau_1$, the column `NAR$tau1` are all `TRUE`. However $\tau_2 = 1.6$, and the third person has survival time $t = 0.8 < \tau_2$ and so the third entry in the column `NAR$tau2` is `FALSE`.

DDD

`DDD` for `k = 1` is shown in #12#. `DDD[[e]][n,m]` $= D_{kemn}$ in Equation (30) in "rmap-formulas-v01.pdf" from the rmap website. `DDD` is a list containing two elements, indexed by `e`. `DDD[[e]][n,m]` is TRUE if $e_n == e$ and $t_n == \tau_m$; in other words the `n` th observation has event `e` at time τ_m.

`DDD[[1]]` are all FALSE except the `[n = 3, tau1]`, `[n = 1, tau2]`, and `[n = 4, tau3]` entries because `e = 1` for the observation `n = 3` and it occurred at time `t = tau1`, `e = 1` for the observation `n = 1` and it occurred at time `t = tau2`, `e = 1` for the observation `n = 4` and it occurred at time `t = tau3`, `DDD[[2]]` are all FALSE except the `[n = 5, tau1]` entry because `e = 2` for observation `n = 5` occurred at time τ_2.

denom

`denom` for `k = 1` is shown in #13#. `denom` $= N_{km}$ in Equation (61) in "rmap-formulas-v01.pdf" from the rmap website.

For `k = 1`, at time `tau1 = 0.8`, all five observations are at risk. Because of two stage sampling, we need to add up the weights aaa for the five observations `1 + 1.33 + 1 + 1 + 1.33 = 5.67`. At time `tau2 = 1.6`, observations 1, 2, 4, 5 are at risk, and adding up their weights `1 + 1.33 + 1 + 1.33 = 4.67`, and we have corroborated the first two entries in `denom` in #13#.

lambdaHat

`lambdaHat` is displayed in #9# and #10#. `lambdaHat[e,m]` $= \tilde{\lambda}_{kem}$ in Equations (68) where the numerator is (59) in "rmap-formulas-v01.pdf" from the rmap website. `lambdaHat` is a list with two elements, one element for `k = 1` and a second element for `k = 2`. Each element is a matrix with two rows (for `event = e = 1` and `event e = 2`) and M_k columns.

4. uuuFn

`uuu` is displayed in #14# but is redisplayed here with informative headings. The shaded region will be verified below in a hand calculation.

			k	1						2	
			e	1			2			1	2
k	n	gamma	m	1	2	3	1	2	3	1	1
1	1	1.41		-1.21	4.67	0.00	-1.21	0.0	0.00	0.00	0.00
	2	1.41		-1.21	-2.00	-1.75	-1.21	-2.0	-1.75	0.00	0.00
	3	1.41		5.67	0.00	0.00	0.00	0.0	0.00	0.00	0.00
	6	1.41		-1.21	-2.00	2.33	-1.21	-2.0	0.00	0.00	0.00
	7	1.41		-1.21	0.00	0.00	-1.21	3.5	0.00	0.00	0.00
2	4	-3.43		0.00	0.00	0.00	0.00	0.0	0.00	-1.75	-1.75
	5	-3.43		0.00	0.00	0.00	0.00	0.0	0.00	2.33	0.00

The first `K - 1` columns of `uuu` are given by Equation (36) in "rmap-formulas-v01.pdf" from the rmap website. (In the example, `K = 2`, and so there is just one `gamma` column.) The remaining columns are given by Equation (52) in "rmap-formulas-v01.pdf" from the rmap website. These columns are ordered $\{(k, e, m) | k = 1, \cdots, K; e = 1, 2; m = 1, \cdots, M_k\}$ where `k` moves most slowly, followed by `e` and then `m` moves the fastest.

`uuu` is ordered by risk groups `k`; the first five rows correspond to the five rows in `d_k_equals_1` in #4#, and the last two rows correspond to the two rows in `d_k_equals_2` in #5#. I will do a hand calculation of Equation (52) in "rmap-formulas-v01.pdf" from the rmap website for `k = 1` here.

$$u(\lambda_{kem}) = N_{kmn} \left(\frac{D_{kemn}}{\lambda_{kem}} - \frac{1 - D_{k \bullet mn}}{1 - \lambda_{k \bullet m}} \right) \tag{1}$$

```
e = 1
NAR = lambdaHat[["k1"]]$NAR
DThis = lambdaHat[["k1"]]$DDD[[e]]
DDot =  lambdaHat[["k1"]]$DDD[[1]] +  lambdaHat[["k1"]]$DDD[[2]]
lambdaThis = lambdaHat[["k1"]]$lambdaHat[e,]
lambdaDot = lambdaHat[["k1"]]$lambdaHat[1,] + lambdaHat[["k1"]]$lambdaHat[2,]

mapply(function(N1, D1, lambda1, DDot1, lambdaDot1){
  term1 = ifelse(is.nan(D1/lambda1), 0, D1/lambda1)
  N1 * (term1 - (1 - DDot1)/(1 - lambdaDot1))
}, as.data.frame(NAR), as.data.frame(DThis), lambdaThis, as.data.frame(DDot), lambdaDot, SIMPLIFY = TRUE)
```

```
      tau1  tau2  tau3
 [1,] -1.21  4.67  0.00
 [2,] -1.21 -2.00 -1.75
 [3,]  5.67  0.00  0.00
 [4,] -1.21 -2.00  2.33
 [5,] -1.21  0.00  0.00
```

I have matched the shaded region of the pretty picture of `UUU`.

5. VVVFn

`VVV` is displayed in #15# but is redisplayed here with informative headings and shading.

				k	1						2	
				e	1			2			1	2
k	e	m	gamma	m	1	2	3	1	2	3	1	1
		gamma	0.207		0.000	0.000	0.00	0	0.000	0	0.00	0
k	e	m										
1	1	1	0.000		0.205	0.000	0.00	0	0.000	0	0.00	0
		2	0.000		0.000	0.289	0.00	0	-0.105	0	0.00	0
		3	0.000		0.000	0.000	0.84	0	0.000	0	0.00	0
	2	1	0.000		0.000	0.000	0.00	0	0.000	0	0.00	0
		2	0.000		0.000	-0.105	0.00	0	0.350	0	0.00	0
		3	0.000		0.000	0.000	0.00	0	0.000	0	0.00	0
2	1	1	0.000		0.000	0.000	0.00	0	0.000	0	0.84	0
	2	1	0.000		0.000	0.000	0.00	0	0.000	0	0.00	0

`VVV` is a block diagonal matrix with `K + 1` boxes down the diagonal. The first box is a square symmetric matrix with `K - 1` rows and columns and given by Equation (51) in "rmap-formulas-v01.pdf" from the rmap website. The second box is for `k = 1`, and so on until the `K + 1` th box for `k = K` . The `k + 1` st box has row headings $\{(e_1, m)|e_1 = 1, 2; m = 1, \cdots, M_k\}$, where e_1 grows more slowly than m, and column headings $\{(e_2, m)|e_2 = 1, 2; m = 1, \cdots, M_k\}$. The entry in each cell in the box is

$$V_{k,e_1,e_2,m} = \begin{cases} \lambda_{ke_1m}(1 - \lambda_{ke_1m}) & \text{if } e_1 = \text{e}_2 \\ -\lambda_{k1m}\lambda_{k2m} & \text{if } e_1 \neq \text{e}_2 \end{cases} \tag{2}$$

For each of the " `k` -boxes", the only nonzero off-diagonal elements occur at $e_1 \neq e_2$ for any (k, m) in which the distinct event time τ_{km} contains both events `e = 1` and `e = 2`. For `k = 1` (see #4#) there is an observation for which `e = 1` and `t = tau2` (the first row) and there is another observation for which `e = 2` and `t = tau2` (the last row). This leads to a nonzero value in `VVV` for `k = 1`, `m = 2`, and `e1` not equal `e2`.

In the example, `K = 2` , so there are `K + 1 = 3` boxes, colored green. There is one distinct event time τ_{km} , with `k = 1` and `m = 2` , containing both events `e = 1` and `e = 2` . The nonzero values in `VVV` for these `k` , `m` and `e` are colored in red.

6. B2Fn

`B2` is displayed in #16#. `B2` is given in Equation (83) in "rmap-formulas-v01.pdf" from the rmap website. I've included the remaining steps (after calculating uuu) to `B2Fn` here.

muHat

`muHat` is given by Equation (81) in "rmap-formulas-v01.pdf" from the rmap website.

$$\hat{\mu}_c = \frac{1}{\bar{N}_c} \sum_{n \in \bar{Q}_c} u_n \tag{3}$$

```
cUni = names(baseArgs$N)
muHat = t(sapply(cUni, function(ccc) colMeans(uuu[baseArgs$c[order(baseArgs$k)] == ccc, ])))   ###DJDJ
                 # Had to reorder based on the order of $k to get results consistent with rime 0.00.008
print("muHat")
muHat
```

```
 [1] "muHat"
     [,1]  [,2]   [,3]   [,4]   [,5] [,6]   [,7]   [,8]   [,9]
 A  0.202  0.81  0.667  0.583 -0.607 -0.5  0.000  0.583  0.000
 B -0.202 -0.81 -0.667 -0.583 -0.810  0.5 -0.583 -0.583 -0.583
```

Hand calculation of `muHat` .

```
d_k_equals_2 = cbind(d[d$k == 2,c("e", "t", "c")], aaa = N[d[d$k == 2,]$c]/n[d[d$k == 2,]$c], k = 2)
x = rbind(d_k_equals_1, d_k_equals_2)
structure(rbind(apply(uuu[x$c == "A",], 2, sum)/n["A"],apply(uuu[x$c == "B",], 2, sum)/n["B"]),
          dimnames = list(c("A","B"), NULL))
```

```
     [,1]  [,2]   [,3]   [,4]   [,5] [,6]   [,7]   [,8]   [,9]
 A  0.202  0.81  0.667  0.583 -0.607 -0.5  0.000  0.583  0.000
 B -0.202 -0.81 -0.667 -0.583 -0.810  0.5 -0.583 -0.583 -0.583
```

PhiHat

`PhiHat` is given by Equation (82) in "rmap-formulas-v01.pdf" from the rmap website.

$$\hat{\Phi}_c = \frac{1}{\bar{N}_c} \sum_{n \in \bar{Q}_c} u_n u_n^T \tag{4}$$

```
PhiHat = structure(lapply(cUni, function(ccc) {
  uuu_c = uuu[baseArgs$c[order(baseArgs$k)] == ccc, ]
  uuT = vvTsumFn(uuu_c)
  uuT / nrow(uuu_c)
}), .Names = cUni)
print("PhiHat")
PhiHat
```

```
[1] "PhiHat"
$A
        [,1]   [,2]   [,3]   [,4]   [,5]   [,6] [,7]  [,8] [,9]
 [1,]  4.434  1.143  0.941  0.824 -0.857 -0.706    0 -2.00    0
 [2,]  1.143  8.765 -0.810 -0.708  0.737  0.607    0  0.00    0
 [3,]  0.941 -0.810  6.444 -1.167 -0.810  1.000    0  0.00    0
 [4,]  0.824 -0.708 -1.167  1.361 -0.708 -1.167    0  0.00    0
 [5,] -0.857  0.737 -0.810 -0.708  0.737  0.607    0  0.00    0
 [6,] -0.706  0.607  1.000 -1.167  0.607  1.000    0  0.00    0
 [7,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0
 [8,] -2.000  0.000  0.000  0.000  0.000  0.000    0  1.36    0
 [9,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0

$B
        [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7] [,8] [,9]
 [1,]  5.247 -1.143 -0.941 -0.824 -1.143  0.706 -0.824 2.00 2.00
 [2,] -1.143  0.983  0.810  0.708  0.983 -0.607  0.708 0.00 0.00
 [3,] -0.941  0.810  1.333  1.167  0.810  1.333  1.167 0.00 0.00
 [4,] -0.824  0.708  1.167  1.021  0.708  1.167  1.021 0.00 0.00
 [5,] -1.143  0.983  0.810  0.708  0.983 -0.607  0.708 0.00 0.00
 [6,]  0.706 -0.607  1.333  1.167 -0.607  5.417  1.167 0.00 0.00
 [7,] -0.824  0.708  1.167  1.021  0.708  1.167  1.021 0.00 0.00
 [8,]  2.000  0.000  0.000  0.000  0.000  0.000  0.000 1.02 1.02
 [9,]  2.000  0.000  0.000  0.000  0.000  0.000  0.000 1.02 1.02
```

Hand calculation of `PhiHat`

```
uThis = uuu[x$c == "A",]
vvTsumFn(uThis)/n["A"]
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6] [,7]  [,8] [,9]
[1,]  4.434  1.143  0.941  0.824 -0.857 -0.706    0 -2.00    0
[2,]  1.143  8.765 -0.810 -0.708  0.737  0.607    0  0.00    0
[3,]  0.941 -0.810  6.444 -1.167 -0.810  1.000    0  0.00    0
[4,]  0.824 -0.708 -1.167  1.361 -0.708 -1.167    0  0.00    0
```

```
[5,] -0.857  0.737 -0.810 -0.708  0.737  0.607    0  0.00    0
[6,] -0.706  0.607  1.000 -1.167  0.607  1.000    0  0.00    0
[7,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0
[8,] -2.000  0.000  0.000  0.000  0.000  0.000    0  1.36    0
[9,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0
```

We recently introduced `vvTsumFnC` to make calculations faster.

```
vvTsumFnC(uThis)/n["A"]
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6] [,7]  [,8] [,9]
[1,]  4.434  1.143  0.941  0.824 -0.857 -0.706    0 -2.00    0
[2,]  1.143  8.765 -0.810 -0.708  0.737  0.607    0  0.00    0
[3,]  0.941 -0.810  6.444 -1.167 -0.810  1.000    0  0.00    0
[4,]  0.824 -0.708 -1.167  1.361 -0.708 -1.167    0  0.00    0
[5,] -0.857  0.737 -0.810 -0.708  0.737  0.607    0  0.00    0
[6,] -0.706  0.607  1.000 -1.167  0.607  1.000    0  0.00    0
[7,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0
[8,] -2.000  0.000  0.000  0.000  0.000  0.000    0  1.36    0
[9,]  0.000  0.000  0.000  0.000  0.000  0.000    0  0.00    0
```

`PhiHat` is a list containing an element for each c in `cUni`. In our example `cUni = c(A, B)`. `PhiHat[[c]]` is a square matrix of dimension $K - 1 + 2 * \sum_{k=1}^{K} M_k$ squared. In our example, each `PhiHat[[c]]` is of dimension 9.

PhiHatPart - muHatPart

```
omegaHat = baseArgs$N / sum(baseArgs$N)
print("omegaHat")
omegaHat
pHat = baseArgs$n / baseArgs$N
print("pHat")
pHat
PhiHatPart = apply(
  array(unlist(mapply(PhiHatFn, omegaHat, pHat, baseArgs$n, PhiHat, SIMPLIFY = FALSE)),
    c(ncol(uuu), ncol(uuu), length(cUni))),
  c(1,2), sum)
multiplier = omegaHat * ((1 - pHat)/pHat) * baseArgs$n / (baseArgs$n - 1)
print("multiplier")
multiplier
muHatPart = vvTsumFn(muHat, multiplier)
print("B2 by hand")
PhiHatPart - muHatPart
```

```
[1] "omegaHat"
  A   B
0.5 0.5
[1] "pHat"
   A    B
1.00 0.75
[1] "multiplier"
   A    B
0.00 0.25
[1] "B2 by hand"
        [,1]    [,2]    [,3]   [,4]    [,5]    [,6]   [,7]    [,8]    [,9]
 [1,]  1.302 -0.3265 -0.2689 -0.235 -0.3265  0.2017 -0.235  0.4706  0.4706
 [2,] -0.327  0.0819  0.0675  0.059  0.0819 -0.0506  0.059 -0.1181 -0.1181
 [3,] -0.269  0.0675  0.2222  0.194  0.0675  0.4167  0.194 -0.0972 -0.0972
 [4,] -0.235  0.0590  0.1944  0.170  0.0590  0.3646  0.170 -0.0851 -0.0851
 [5,] -0.327  0.0819  0.0675  0.059  0.0819 -0.0506  0.059 -0.1181 -0.1181
```

```
 [6,]  0.202 -0.0506  0.4167  0.365 -0.0506  1.2917  0.365  0.0729  0.0729
 [7,] -0.235  0.0590  0.1944  0.170  0.0590  0.3646  0.170 -0.0851 -0.0851
 [8,]  0.471 -0.1181 -0.0972 -0.085 -0.1181  0.0729 -0.085  0.1701  0.1701
 [9,]  0.471 -0.1181 -0.0972 -0.085 -0.1181  0.0729 -0.085  0.1701  0.1701
```

7. V2StageFn

`V2Stage` is displayed in #17#. `V2Stage` is given by Equation (80) in "rmap-formulas-v01.pdf" from the rmap website. `V2Stage` has the same dimensions as `VVV` .