Multinomial Logit Models with R*

```
> # install.packages("mlogit", dependencies = TRUE) # Only need to do this once
> library(mlogit)
> math =
read.table("http://www.utstat.toronto.edu/~brunner/312f12/code_n_data/mathcat.data")
> math[1:3,]
  hsgpa hsengl hscalc   course passed outcome
1  78.0     80    Yes Mainstrm     No  Failed
2  66.0     75    Yes Mainstrm    Yes  Passed
3  80.2     70    Yes Mainstrm    Yes  Passed
> # Try a simple logistic regression.
```

The explanatory vars can be characteristics of the individual case (individual specific), or of the alternative (alternative specific) -- that is the value of the response variable.

The `mlogit` function requires its own special type of data frame, and there are two data formats: ``wide" and ``long." When there are individual specific variables and lots of individuals, the long (that is, long rows) format may be preferable, and we'll have n rows, which is what we're accustomed to. But if there are response-specific covariates, each such variable requires a separate column for each value of the response variable.

The `mlogit.data` function converts ordinary data frames to a type required by `mlogit`. I can only make the wide format work.

```
> # Try a simple logistic regression.
> # Make an mlogit data frame in wide format
> wide0 = mlogit.data(math,shape="wide",choice="passed")
> wide0[1:8,]
  hsgpa hsengl hscalc   course passed outcome chid alt   idx
1  78.0     80    Yes Mainstrm   TRUE  Failed    1  No  1:No
2  78.0     80    Yes Mainstrm  FALSE  Failed    1 Yes 1:Yes
3  66.0     75    Yes Mainstrm  FALSE  Passed    2  No  2:No
4  66.0     75    Yes Mainstrm   TRUE  Passed    2 Yes 2:Yes
5  80.2     70    Yes Mainstrm  FALSE  Passed    3  No  3:No
6  80.2     70    Yes Mainstrm   TRUE  Passed    3 Yes 3:Yes
7  81.7     67    Yes Mainstrm  FALSE  Passed    4  No  4:No
8  81.7     67    Yes Mainstrm   TRUE  Passed    4 Yes 4:Yes
```

`chid` is Choice Identification. Each person in the sample represents a "choice."
alt stands for alternatives. The alternatives are No and Yes.

Model description (formula) is more complex than for `glm`, because the models are more complex. Have the `mformula` function. It provides for individual specific variables (the kind we use) and two kinds of alternative specific variables. Can provide 3 parts, separated by vertical bars. The first and third are alternative specific. If we stick to individual-specific vars, we can leave off the last, like this:

```
> simple0 = mlogit(passed ~ 0 | hsgpa, data=wide0); summary(simple0)

Call:
mlogit(formula = passed ~ 0 | hsgpa, data = wide0, method = "nr",
    print.level = 0)

Frequencies of alternatives:
     No     Yes
0.40102 0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000119
successive fonction values within tolerance limits

Coefficients :
                  Estimate Std. Error t-value  Pr(>|t|)
Yes:(intercept) -15.210112   1.998398 -7.6112 2.709e-14 ***
Yes:hsgpa         0.197734   0.025486  7.7587 8.660e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -221.72
McFadden R^2:  0.16436
Likelihood ratio test : chisq = 87.221 (p.value = < 2.22e-16)
```

(Repeating some output)

```
Coefficients :
                  Estimate Std. Error t-value  Pr(>|t|)
Yes:(intercept) -15.210112   1.998398 -7.6112 2.709e-14 ***
Yes:hsgpa         0.197734   0.025486  7.7587 8.660e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -221.72
McFadden R^2:  0.16436
Likelihood ratio test : chisq = 87.221 (p.value = < 2.22e-16)
```

```
> # Compare
> summary(glm(y~hsgpa,family=binomial,data=math))

Call:
glm(formula = y ~ hsgpa, family = binomial, data = math)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5152  -1.0209   0.4435   0.9321   2.1302

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -15.21013    1.99832  -7.611 2.71e-14 ***
hsgpa         0.19773    0.02548   7.759 8.56e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 530.66  on 393  degrees of freedom
Residual deviance: 443.43  on 392  degrees of freedom
AIC: 447.43
```

```
> anova(glm(y~hsgpa,family=binomial,data=math))
Terms added sequentially (first to last)


      Df Deviance Resid. Df Resid. Dev
NULL                    393     530.66
hsgpa  1   87.221       392     443.43
```

```
> # Excellent. Now try simple regression with a 3-category outcome.
> wide = mlogit.data(math,shape="wide",choice="outcome")
> wide[1:8,]
  hsgpa hsengl hscalc   course passed outcome chid         alt    idx
1  78.0     80    Yes Mainstrm     No   FALSE    1 Disappeared 1:ared
2  78.0     80    Yes Mainstrm     No    TRUE    1      Failed 1:iled
3  78.0     80    Yes Mainstrm     No   FALSE    1      Passed 1:ssed
4  66.0     75    Yes Mainstrm    Yes   FALSE    2 Disappeared 2:ared
5  66.0     75    Yes Mainstrm    Yes   FALSE    2      Failed 2:iled
6  66.0     75    Yes Mainstrm    Yes    TRUE    2      Passed 2:ssed
7  80.2     70    Yes Mainstrm    Yes   FALSE    3 Disappeared 3:ared
8  80.2     70    Yes Mainstrm    Yes   FALSE    3      Failed 3:iled

~~~ indexes ~~~~
  chid         alt
1    1 Disappeared
2    1      Failed
3    1      Passed
4    2 Disappeared
5    2      Failed
6    2      Passed
7    3 Disappeared
8    3      Failed
indexes:  1, 2

> head(math)
  hsgpa hsengl hscalc   course passed outcome
1  78.0     80    Yes Mainstrm     No  Failed
2  66.0     75    Yes Mainstrm    Yes  Passed
3  80.2     70    Yes Mainstrm    Yes  Passed
4  81.7     67    Yes Mainstrm    Yes  Passed
5  86.8     80    Yes Mainstrm    Yes  Passed
6  76.7     75    Yes Mainstrm    Yes  Passed
```

```
> # Note use of reflevel to put P(Failed) in the denominators
> simple1 = mlogit(outcome ~ 0 | hsgpa, reflevel = 'Failed', data=wide)
> summary(simple1)

Call:
mlogit(formula = outcome ~ 0 | hsgpa, data = wide, reflevel = "Failed",
    method = "nr")

Frequencies of alternatives:choice
     Failed Disappeared      Passed
    0.15482     0.24619     0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 1.09E-05
successive function values within tolerance limits

Coefficients :
                           Estimate Std. Error z-value      Pr(>|z|)
(Intercept):Disappeared   1.904226   2.744979  0.6937        0.4879
(Intercept):Passed      -13.393056   2.570453 -5.2104 0.00000018845 ***
hsgpa:Disappeared        -0.018816   0.035775 -0.5260        0.5989
hsgpa:Passed              0.186437   0.033018  5.6465 0.00000001637 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -326.96
McFadden R^2:  0.11801
Likelihood ratio test : chisq = 87.497 (p.value = < 0.000000000000000222)


> # Estimate probabilities for a student with HSGPA = 90
> betahat1 = coefficients(simple1); betahat1

(Intercept):Disappeared      (Intercept):Passed       hsgpa:Disappeared
             1.90422575            -13.39305637             -0.01881621
           hsgpa:Passed
             0.18643711
```

```
> # Estimate probabilities for a student with HSGPA = 90
```

$$\pi_1 = \frac{e^{L_1}}{1 + e^{L_1} + e^{L_2}}$$

$$\pi_2 = \frac{e^{L_2}}{1 + e^{L_1} + e^{L_2}}$$

$$\pi_3 = \frac{1}{1 + e^{L_1} + e^{L_2}}$$

```
>  betahat1
(Intercept):Disappeared      (Intercept):Passed      hsgpa:Disappeared
             1.90422575            -13.39305637            -0.01881621
          hsgpa:Passed
             0.18643711
> gpa = 90
> L1 = betahat1[1] + betahat1[3]*gpa  # Disappeared
> L2 = betahat1[2] + betahat1[4]*gpa  # Passed
> denom = 1+exp(L1)+exp(L2)
> pihat1 = exp(L1)/denom   # Disappeared
> pihat2 = exp(L2)/denom   # Passed
> pihat3 = 1/denom         # Failed
> rbind(pihat1,pihat2,pihat3)

       (Intercept):Disappeared
pihat1              0.03883621
pihat2              0.92970789
pihat3              0.03145590
```

```
> # # More interesting full model.
> fullmod = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc+course,
+                  reflevel = 'Failed', data=wide)
> summary(fullmod)

Call:
mlogit(formula = outcome ~ 0 | hsgpa + hsengl + hscalc + course,
    data = wide, reflevel = "Failed", method = "nr")

Frequencies of alternatives:choice
     Failed Disappeared      Passed
    0.15482     0.24619     0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000216
successive function values within tolerance limits

Coefficients :
                              Estimate  Std. Error z-value  Pr(>|z|)
(Intercept):Disappeared      2.5734410   2.8288386  0.9097   0.36297
(Intercept):Passed         -14.0411854   2.7005870 -5.1993 2.000e-07 ***
hsgpa:Disappeared           -0.0079779   0.0413277 -0.1930   0.84693
hsgpa:Passed                 0.2157706   0.0382179  5.6458 1.644e-08 ***
hsengl:Disappeared          -0.0067241   0.0251049 -0.2678   0.78882
hsengl:Passed               -0.0399811   0.0228733 -1.7479   0.08047 .
hscalcYes:Disappeared       -0.3902775   0.6742796 -0.5788   0.56272
hscalcYes:Passed             1.0009683   0.8215247  1.2184   0.22306
courseElite:Disappeared     -2.0666545   0.9836801 -2.1009   0.03565 *
courseElite:Passed           0.6032839   0.8044316  0.7500   0.45328
courseMainstrm:Disappeared  -0.6834686   0.5560854 -1.2291   0.21905
courseMainstrm:Passed        0.4086564   0.6339142  0.6447   0.51915
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -312.26
McFadden R^2:  0.15766
Likelihood ratio test : chisq = 116.89 (p.value = < 2.22e-16)

> # Want to make Mainstream (3d cat) the ref category for course
> # Without rubbing out the nice names.
```

```
> # Working within the data frame math
> contrasts(math$course)
Error in contrasts(math$course) : contrasts apply only to factors
> is.factor(math$course)
[1] FALSE
> math$course = factor(math$course)
> contrasts(math$course)
         Elite Mainstrm
Catch-up     0        0
Elite        1        0
Mainstrm     0        1
> contrasts(math$course) = contr.treatment(3, base = 3)
> contrasts(math$course)
         1 2
Catch-up 1 0
Elite    0 1
Mainstrm 0 0
> colnames(contrasts(math$course)) = c('Catch-up', 'Elite')
> contrasts(math$course)
         Catch-up Elite
Catch-up        1     0
Elite           0     1
Mainstrm        0     0
>
> # Now re-create the mlogit data set and fit the model again. Compare
> # LR test statistic of chisq = 116.89
> wide = mlogit.data(math,shape="wide",choice="outcome")
> fullmod = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc+course, data=wide)
> summary(fullmod)

Call:
mlogit(formula = outcome ~ 0 | hsgpa + hsengl + hscalc + course,
    data = wide, method = "nr")

Frequencies of alternatives:choice
 Failed    Gone  Passed
0.15482 0.24619 0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000216
successive function values within tolerance limits

Coefficients :
                        Estimate  Std. Error z-value      Pr(>|z|)
(Intercept):Gone       2.5734410   2.8288386  0.9097       0.36297
(Intercept):Passed   -14.0411854   2.7005870 -5.1993 0.00000020003 ***
hsgpa:Gone            -0.0079779   0.0413277 -0.1930       0.84693
hsgpa:Passed           0.2157706   0.0382179  5.6458 0.00000001644 ***
hsengl:Gone           -0.0067241   0.0251049 -0.2678       0.78882
hsengl:Passed         -0.0399811   0.0228733 -1.7479       0.08047 .
hscalcYes:Gone        -0.3902775   0.6742796 -0.5788       0.56272
hscalcYes:Passed       1.0009683   0.8215247  1.2184       0.22306
courseElite:Gone      -2.0666545   0.9836801 -2.1009       0.03565 *
courseElite:Passed     0.6032839   0.8044316  0.7500       0.45328
courseMainstrm:Gone   -0.6834686   0.5560854 -1.2291       0.21905
courseMainstrm:Passed  0.4086564   0.6339142  0.6447       0.51915
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -312.26
McFadden R^2:  0.15766
Likelihood ratio test : chisq = 116.89 (p.value = < 0.00000000000000222)
>
```

```
> # Astonishingly, it did NOT use the contrasts defined in math.
> contrasts(wide$course)
         Elite Mainstrm
Catch-up     0        0
Elite        1        0
Mainstrm     0        1

> # Fix it up within the mlogit data set wide
> contrasts(wide$course) = contr.treatment(3, base = 3)
> colnames(contrasts(wide$course)) = c('Catch-up', 'Elite')
> contrasts(wide$course)

         Catch-up Elite
Catch-up        1     0
Elite           0     1
Mainstrm        0     0

> # Fit the full model again
> fullmod = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc+course,
+                  reflevel = 'Failed', data=wide)
> summary(fullmod)

Call:
mlogit(formula = outcome ~ 0 | hsgpa + hsengl + hscalc + course,
    data = wide, reflevel = "Failed", method = "nr")

Frequencies of alternatives:choice
     Failed Disappeared      Passed
    0.15482     0.24619     0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000216
successive function values within tolerance limits

Coefficients :
                              Estimate  Std. Error z-value      Pr(>|z|)
(Intercept):Disappeared      1.8899724   2.8959975  0.6526       0.51400
(Intercept):Passed         -13.6325290   2.7562888 -4.9460 0.00000075765 ***
hsgpa:Disappeared           -0.0079779   0.0413277 -0.1930       0.84693
hsgpa:Passed                 0.2157706   0.0382179  5.6458 0.00000001644 ***
hsengl:Disappeared          -0.0067241   0.0251049 -0.2678       0.78882
hsengl:Passed               -0.0399811   0.0228733 -1.7479       0.08047 .
hscalcYes:Disappeared       -0.3902775   0.6742796 -0.5788       0.56272
hscalcYes:Passed             1.0009683   0.8215247  1.2184       0.22306
courseCatch-up:Disappeared   0.6834686   0.5560854  1.2291       0.21905
courseCatch-up:Passed       -0.4086564   0.6339142 -0.6447       0.51915
courseElite:Disappeared     -1.3831859   0.8700722 -1.5897       0.11189
courseElite:Passed           0.1946275   0.5664312  0.3436       0.73114
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -312.26
McFadden R^2:  0.15766
Likelihood ratio test : chisq = 116.89 (p.value = < 0.000000000000000222)
```

```
> # Test Course controlling for HS variables
> nocourse = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc,
+            reflevel = 'Failed', data=wide)
> anova(nocourse,fullmod)
Error in UseMethod("anova") :
  no applicable method for 'anova' applied to an object of class "mlogit"
> summary(nocourse)

Call:
mlogit(formula = outcome ~ 0 | hsgpa + hsengl + hscalc, data = wide,
    reflevel = "Failed", method = "nr")

Frequencies of alternatives:choice
     Failed Disappeared      Passed
    0.15482     0.24619     0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 1.83E-05
successive function values within tolerance limits

Coefficients :
                           Estimate   Std. Error z-value     Pr(>|z|)
(Intercept):Disappeared   2.34769119   2.79507093  0.8399      0.40094
(Intercept):Passed      -13.89165347   2.68022281 -5.1830 0.0000002183 ***
hsgpa:Disappeared        -0.01453386   0.04085813 -0.3557      0.72205
hsgpa:Passed              0.21797643   0.03809182  5.7224 0.0000000105 ***
hsengl:Disappeared       -0.00097165   0.02433134 -0.0399      0.96815
hsengl:Passed            -0.04190565   0.02261526 -1.8530      0.06389 .
hscalcYes:Disappeared    -0.77280145   0.60001813 -1.2880      0.19776
hscalcYes:Passed          1.23197874   0.76884765  1.6024      0.10907
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -318.19
McFadden R^2:  0.14166
Likelihood ratio test : chisq = 105.03 (p.value = < 0.000000000000000222)
> 116.89-105.03 # Diff between Likelihood ratio tests, df=4
[1] 11.86
> # Better
> nocourse$logLik
'log Lik.' -318.1931 (df=8)
> fullmod$logLik
'log Lik.' -312.2625 (df=12)
> G2 = -2 * as.numeric(nocourse$logLik - fullmod$logLik); G2
[1] 11.86122
> pval = 1-pchisq(G2,df=4) # Two betas for each dummy variable.
> pval
[1] 0.01841369
> # Let's keep course and hsgpa.  Do we need hsengl and hscalc?
```

```
> # Let's keep course and hsgpa.  Do we need hsengl and hscalc?
> coursegpa = mlogit(outcome ~ 0 | hsgpa+course, reflevel = 'Failed', data=wide)
>
> G2 = -2 * as.numeric(coursegpa$logLik - fullmod$logLik); G2
[1] 8.457276
> pval = 1-pchisq(G2,df=4) # df=4 again
> pval
[1] 0.07619288

> # It's pretty close. What if we discarded just HScalc?
> coursegpaEng = mlogit(outcome ~ 0 | hsgpa+course+hsengl, reflevel = 'Failed',
data=wide)
> G2 = -2 * as.numeric(coursegpaEng$logLik - fullmod$logLik); G2
[1] 4.238631
> pval = 1-pchisq(G2,df=2) # Two regression coefficients for HS calculus
> pval
[1] 0.1201138

> # Test HS English the same way
> coursegpaCalc = mlogit(outcome ~ 0 | hsgpa+course+hscalc, reflevel = 'Failed',
data=wide)
> G2 = -2 * as.numeric(coursegpaCalc$logLik - fullmod$logLik); G2
[1] 4.354343
> pval = 1-pchisq(G2,df=2) # Two regression coefficients for HS English
> pval
[1] 0.1133617

> # Conclusion: Let's keep just course and hsgpa.
> summary(coursegpa)
Call:
mlogit(formula = outcome ~ 0 | hsgpa + course, data = wide, reflevel = "Failed",
    method = "nr")

Frequencies of alternatives:choice
     Failed Disappeared      Passed
    0.15482     0.24619     0.59898

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.00016
successive function values within tolerance limits

Coefficients :
                             Estimate Std. Error z-value      Pr(>|z|)
(Intercept):Disappeared      1.363464   2.818403  0.4838       0.62855
(Intercept):Passed         -13.264833   2.611242 -5.0799 0.00000037765 ***
hsgpa:Disappeared           -0.012635   0.036570 -0.3455       0.72971
hsgpa:Passed                 0.184952   0.033426  5.5332 0.00000003144 ***
courseCatch-up:Disappeared   0.838789   0.501793  1.6716       0.09461 .
courseCatch-up:Passed       -0.707533   0.590061 -1.1991       0.23049
courseElite:Disappeared     -1.322791   0.857624 -1.5424       0.12298
courseElite:Passed           0.357558   0.549697  0.6505       0.51539
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -316.49
McFadden R^2:  0.14625
Likelihood ratio test : chisq = 108.43 (p.value = < 0.00000000000000222)

> # Test course controlling for just hsgpa. coursegpa is the full model now
> justgpa = mlogit(outcome ~ 0 | hsgpa, reflevel = 'Failed', data=wide)
> G2 = -2 * as.numeric(justgpa$logLik - coursegpa$logLik); G2
[1] 20.93569
> pval = 1-pchisq(G2,df=4) # df=4
> pval
[1] 0.0003261024
```

```
> # Wald test of the same null hypothesis:
> # beta21 = beta31  =  beta22 = beta32 = 0
> # Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
> source("http://www.utstat.utoronto.ca/brunner/Rfunctions/Wtest.txt")
> betahat = coefficients(coursegpa); betahat

     (Intercept):Gone    (Intercept):Passed            hsgpa:Gone          hsgpa:Passed
           1.36346362          -13.26483339           -0.01263518            0.18495194
  courseCatch-up:Gone courseCatch-up:Passed      courseElite:Gone    courseElite:Passed
           0.83878891           -0.70753322           -1.32279088            0.35755823

> Lcourse = rbind(c(0,0,0,0,1,0,0,0),
+                 c(0,0,0,0,0,1,0,0),
+                 c(0,0,0,0,0,0,1,0),
+                 c(0,0,0,0,0,0,0,1)   )
> Vhat = vcov(coursegpa)
> Wtest(Lcourse, betahat, Vhat) # Compare G-squared = 20.94
           W           df      p-value
17.246117481  4.000000000  0.001731276

> # Custom questions are easier with Wald tests.
> # For example, controlling for HS GPA, do the three courses differ in the
> # probability a student will disappear (as opposed to failing)?
```

```
> # Custom questions are easier with Wald tests.
> # For example, controlling for HS GPA, do the three courses differ in the
> # probability a student will disappear (as opposed to failing)?
>
> L = rbind(c(0,0,0,0,1,0,0,0),
+           c(0,0,0,0,0,0,1,0)   )
> Wtest(L, betahat, Vhat)
         W         df    p-value
5.64697548 2.00000000 0.05939841
> # If test had been significant, proceed to pairwise comparisons



> # Prediction
> # Let's get estimated probabilities for a student with an average hsgpa
> xbar = mean(math$hsgpa); xbar
[1] 79.73756
>
> cup = c(1,xbar,1,0)
> et  = c(1,xbar,0,1)
> ms  = c(1,xbar,0,0)
> betahat  = coefficients(coursegpa); betahat
     (Intercept):Gone    (Intercept):Passed            hsgpa:Gone          hsgpa:Passed
           1.36346362          -13.26483339           -0.01263518            0.18495194
  courseCatch-up:Gone courseCatch-up:Passed      courseElite:Gone    courseElite:Passed
           0.83878891           -0.70753322           -1.32279088            0.35755823
> betahat1 = betahat[c(1,3,5,7)] # Gone
> betahat2 = betahat[c(2,4,6,8)] # Passed
>
> # Catch-up course
> L1 = sum(cup*betahat1); L2 = sum(cup*betahat2)
> denom = 1+exp(L1)+exp(L2)
> pi1 = exp(L1)/(denom); pi2 = exp(L2)/(denom); pi3 = 1/denom
> estimate = c(pi1,pi2,pi3); names(estimate) = c('Disappeared','Passed','Failed')
> estimate # For the catch-up course

Disappeared      Passed      Failed
  0.5101646   0.3353685   0.1544669
>
> # Elite course
> L1 = sum(et*betahat1); L2 = sum(et*betahat2)
> denom = 1+exp(L1)+exp(L2)
> pi1 = exp(L1)/(denom); pi2 = exp(L2)/(denom); pi3 = 1/denom
> estimate = c(pi1,pi2,pi3); names(estimate) = c('Disappeared','Passed','Failed')
> estimate # For the elite course
Disappeared      Passed      Failed
 0.04952327  0.82025110  0.13022563

> # Mainsteam course
> L1 = sum(ms*betahat1); L2 = sum(ms*betahat2)
> denom = 1+exp(L1)+exp(L2)
> pi1 = exp(L1)/(denom); pi2 = exp(L2)/(denom); pi3 = 1/denom
> estimate = c(pi1,pi2,pi3); names(estimate) = c('Disappeared','Passed','Failed')
> estimate # For the mainstream course
Disappeared      Passed      Failed
  0.2089287   0.6447173   0.1463540
```