Multinomial Logit Models with R

The mlogit package has already been downloaded.

```
> library(mlogit)
Loading required package: Formula
Loading required package: statmod
Loading required package: lmtest
Loading required package: zoo

Attaching package: 'zoo'

The following object(s) are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: maxLik
Loading required package: miscTools
Loading required package: MASS
> math =
read.table("http://www.utstat.toronto.edu/~brunner/312f12/code_n_data/mathc
at.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 wide 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 long format work.

```
> # Try a simple logistic regression.
> math0 = math[,c(1,5)]; math0[1:3,]
  hsgpa passed
1  78.0     No
2  66.0    Yes
3  80.2    Yes
> # Make an mlogit data frame in long format
> long0 = mlogit.data(math0,shape="wide",choice="passed")
> head(long0)
      hsgpa passed chid alt
1.No   78.0   TRUE    1  No
1.Yes  78.0  FALSE    1 Yes
2.No   66.0  FALSE    2  No
2.Yes  66.0   TRUE    2 Yes
3.No   80.2  FALSE    3  No
3.Yes  80.2   TRUE    3 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=long0); summary(simple0)

Call:
mlogit(formula = passed ~ 0 | hsgpa, data = long0, 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(passed~hsgpa,family=binomial,data=math))

Call:
glm(formula = passed ~ 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(passed~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.

> # I think I have to make an mlogit data frame with just the vars I want.
> # First try to make reference category of outcome Failed.
> # Setting contrasts had no effect.
> # Change the alphabetical order
>
> outcome = as.character(math$outcome)
> for(j in 1:length(outcome))
+     {if(outcome[j]=='Disappeared') outcome[j]='Gone'}
> math$outcome = factor(outcome)
> math1 = math[,c(1,6)]
> long1 = mlogit.data(math1,shape="wide",choice="outcome")
> head(long1)
         hsgpa outcome chid    alt
1.Failed    78    TRUE    1 Failed
1.Gone      78   FALSE    1   Gone
1.Passed    78   FALSE    1 Passed
2.Failed    66   FALSE    2 Failed
2.Gone      66   FALSE    2   Gone
2.Passed    66    TRUE    2 Passed

> 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
```

```
> simple1 = mlogit(outcome ~ 0 | hsgpa, data=long1)
> summary(simple1)

Call:
mlogit(formula = outcome ~ 0 | hsgpa, data = long1, method = "nr",
    print.level = 0)

Frequencies of alternatives:
 Failed    Gone  Passed
0.15482 0.24619 0.59898

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

Coefficients :
                     Estimate Std. Error t-value  Pr(>|t|)
Gone:(intercept)     1.904226   2.744979  0.6937    0.4879
Passed:(intercept) -13.393056   2.570453 -5.2104 1.884e-07 ***
Gone:hsgpa          -0.018816   0.035775 -0.5260    0.5989
Passed:hsgpa         0.186437   0.033018  5.6465 1.637e-08 ***
---
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 = < 2.22e-16)
> # Estimate probabilities for a student with HSGPA = 90

>
> betahat1 = simple1$coefficients; betahat1
  Gone:(intercept) Passed:(intercept)         Gone:hsgpa       Passed:hsgpa
        1.90422575       -13.39305637        -0.01881621         0.18643711
attr(,"fixed")
  Gone:(intercept) Passed:(intercept)         Gone:hsgpa       Passed:hsgpa
             FALSE              FALSE              FALSE              FALSE
```

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

$$
\begin{aligned}
\pi_1 &= \frac{e^{L_1}}{1+e^{L_1}+e^{L_2}} \\
\pi_2 &= \frac{e^{L_2}}{1+e^{L_1}+e^{L_2}} \\
\pi_k &= \frac{1}{1+e^{L_1}+e^{L_2}}
\end{aligned}
$$

```
>  betahat1
  Gone:(intercept) Passed:(intercept)        Gone:hsgpa       Passed:hsgpa
        1.90422575       -13.39305637       -0.01881621         0.18643711
attr(,"fixed")
  Gone:(intercept) Passed:(intercept)        Gone:hsgpa       Passed:hsgpa
             FALSE              FALSE             FALSE              FALSE
> gpa = 90
> L1 = betahat1[1] + betahat1[3]*gpa  # Gone
> L2 = betahat1[2] + betahat1[4]*gpa  # Passed
> denom = 1+exp(L1)+exp(L2)
> pihat1 = exp(L1)/denom   # Gone
> pihat2 = exp(L2)/denom   # Passed
> pihat3 = 1/denom         # Failed
> rbind(pihat1,pihat2,pihat3)
       Gone:(intercept)
pihat1       0.03883621
pihat2       0.92970789
pihat3       0.03145590
```

```
> # More interesting full model. First the data frame, without passed.
> long = mlogit.data(math[,c(1:4,6)],shape="wide",choice="outcome")
> fullmod = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc+course, data=long)
> summary(fullmod)

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

Frequencies of alternatives:
 Failed    Gone  Passed
0.15482 0.24619 0.59898

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

Coefficients :
                       Estimate  Std. Error t-value  Pr(>|t|)
Gone:(intercept)      2.5734410   2.8288386  0.9097   0.36297
Passed:(intercept)  -14.0411854   2.7005870 -5.1993 2.000e-07 ***
Gone:hsgpa           -0.0079779   0.0413277 -0.1930   0.84693
Passed:hsgpa          0.2157706   0.0382179  5.6458 1.644e-08 ***
Gone:hsengl          -0.0067241   0.0251049 -0.2678   0.78882
Passed:hsengl        -0.0399811   0.0228733 -1.7479   0.08047 .
Gone:hscalcYes       -0.3902775   0.6742796 -0.5788   0.56272
Passed:hscalcYes      1.0009683   0.8215247  1.2184   0.22306
Gone:courseElite     -2.0666545   0.9836801 -2.1009   0.03565 *
Passed:courseElite    0.6032839   0.8044316  0.7500   0.45328
Gone:courseMainstrm  -0.6834686   0.5560854 -1.2291   0.21905
Passed:courseMainstrm 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)
> # Making Mainstream (3d cat) the ref category for course
> # rubs out the nice names, and all |Z|<2
```

```
> # Test Course controlling for HS variables
> nocourse = mlogit(outcome ~ 0 | hsgpa+hsengl+hscalc, data=long)
> summary(nocourse)

Call:
mlogit(formula = outcome ~ 0 | hsgpa + hsengl + hscalc, data = long,
    method = "nr", print.level = 0)

Frequencies of alternatives:
 Failed    Gone  Passed
0.15482 0.24619 0.59898

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

Coefficients :
                      Estimate  Std. Error t-value  Pr(>|t|)
Gone:(intercept)    2.3477e+00  2.7951e+00  0.8399   0.40094
Passed:(intercept) -1.3892e+01  2.6802e+00 -5.1830 2.183e-07 ***
Gone:hsgpa         -1.4534e-02  4.0858e-02 -0.3557   0.72205
Passed:hsgpa        2.1798e-01  3.8092e-02  5.7224 1.050e-08 ***
Gone:hsengl        -9.7165e-04  2.4331e-02 -0.0399   0.96815
Passed:hsengl      -4.1906e-02  2.2615e-02 -1.8530   0.06389 .
Gone:hscalcYes     -7.7280e-01  6.0002e-01 -1.2880   0.19776
Passed:hscalcYes    1.2320e+00  7.6885e-01  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 = < 2.22e-16)
> 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?
> coursegpa = mlogit(outcome ~ 0 | hsgpa+course, data=long)
> 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
```

Conclusion: Let's keep just `course` and `hsgpa`.