PA1_HW7_GROUP4

Jieda Li, Kristian Nikolov, Parth Patel, Kristiyan Dimitrov

11/18/2019

7.9 (a) What is the odds ratio fo coronary disea for males vs females. Calculate a 95% confidence internval for it.

```
beta_sex<- 0.035
se_beta_sex<-0.0148
ci_lower<-exp(beta_sex-1.96*se_beta_sex)
ci_higher<-exp(beta_sex+1.96*se_beta_sex)
ci_lower
```

```
## [1] 1.00601
```

```
ci_higher
```

```
## [1] 1.066101
```

(b) If the odds of coronary disease for a female with age =50 and cholesterol = 180 are 1 in 10, what are the odds for a male with age =60 and cholesterol = 200? What is the corresponding probability of coronary disease for that male?

```
# Based on the know odds of the female patient, we can ge the beta0 coefficient

beta_age = 0.0906
beta_choles = 0.0755
# The corresponding equation: ln(0.1) = beta0 + beta_age*50 + beta_choles*180
# solve for beta0
beta0 = log(0.1) -beta_age*50-beta_choles*180
beta0
```

```
## [1] -20.42259
```

```
#beta0=-18.02, compute the odds for a male with age =60 and choles = 200, and the corresponding probabi
odds_male = exp(beta0 + beta_age*60 + beta_choles*200 + beta_sex*1)
print("The odds of coronary disease for this male are:")
```

```
## [1] "The odds of coronary disease for this male are:"
```

```
odds_male
```

```
## [1] 1.159994
```

```r
# the corresponding probablity?
prob_male <-odds_male/(1+odds_male)
print("Probability is:")
```

```
## [1] "Probability is:"
```

```r
prob_male
```

```
## [1] 0.5370358
```

7.12(Pregnancy Duration)

7.12(a) Fit a nominal logistic regression model to the training set and make predictions for the test et using the maximum probability rule. What is the correct classification rate and how does it break down among the three categories?

```r
setwd("/Users/jiedali/Documents/courses/401_predictive_analysis/homeworks/hw7/")
pregnancy <-read.csv("Pregnancy.csv")
# Treat Age as categorical variable, using Age 2 as reference category
pregnancy$Age.factor <-as.factor(pregnancy$Age)
head(pregnancy)
```

```
##   Duration Nutrition Alcohol Smoking Age Age.factor
## 1        1       150       0       1   2          2
## 2        1       124       0       0   1          1
## 3        1       128       0       1   2          2
## 4        1       128       0       1   1          1
## 5        1       133       1       1   2          2
## 6        1       130       1       1   2          2
```

```r
#split into training and test set
#All odd-numbered observations into the training set and all even-numbered observations into test set
train_indices<-seq(1,nrow(pregnancy),by=2)
test_indices<-seq(2,nrow(pregnancy),by=2)
train <-pregnancy[train_indices, ]
test <-pregnancy[-train_indices, ]
# Fit a nominal logistic regression model to the training set
library(nnet)
fit_nominal <-multinom(Duration ~ Nutrition+Alcohol+Smoking+relevel(Age.factor, ref="2"), data=train, ma
```

```
## # weights:  21 (12 variable)
## initial  value 56.029227
## iter  10 value 42.573138
## final  value 42.534429
## converged
```

```r
summary(fit_nominal)
```

```
## Call:
## multinom(formula = Duration ~ Nutrition + Alcohol + Smoking +
##     relevel(Age.factor, ref = "2"), data = train, maxit = 1000)
##
## Coefficients:
##   (Intercept)  Nutrition   Alcohol    Smoking
## 2   -1.085473 0.01553680 -1.060328 -0.6273106
## 3   -1.862191 0.03713018 -2.113399 -2.7174780
##   relevel(Age.factor, ref = "2")1 relevel(Age.factor, ref = "2")3
## 2                        1.006311                       0.4746559
## 3                       -2.361983                      -0.7404409
##
## Std. Errors:
##   (Intercept)  Nutrition   Alcohol   Smoking
## 2    2.521758 0.01876176 0.8216689 0.8575421
## 3    2.657861 0.02022154 0.9752696 0.9989745
##   relevel(Age.factor, ref = "2")1 relevel(Age.factor, ref = "2")3
## 2                        1.025365                        1.075981
## 3                        1.504805                        1.329033
##
## Residual Deviance: 85.06886
## AIC: 109.0689
```

```
# Make predictions for the test set using the maximum probablity rule. What is the CCR and how does it
predicted=predict(fit_nominal, type='probs',newdata=test)
n=nrow(test)
Y.hat.1 = rep(0,n)
for(i in 1:nrow(test)) {if(max(predicted[i,])==predicted[i,1]){Y.hat.1[i]=1;}
  else if(max(predicted[i,])==predicted[i,2]) {Y.hat.1[i]=2;}
  else if(max(predicted[i,])==predicted[i,3]) {Y.hat.1[i]=3;}
}
Y.hat.1
```

```
##  [1] 2 2 1 2 3 1 1 3 3 2 1 2 2 3 2 2 2 1 2 3 2 3 2 2 2 1 3 2 1 1 3 3 3 3 3
## [36] 3 3 3 3 2 3 3 3 3 3 3 3 2 2 1 1
```

```
ctable1 = table(test$Duration,Y.hat.1)
ctable1
```

```
##    Y.hat.1
##      1  2  3
##   1  4  6  3
##   2  4  9  4
##   3  2  3 16
```

```
# calcualte correct classification rate
correct.rate1=sum(diag(ctable1)[1:3])/nrow(test);
print("The correct classification rate is:")
```

```
## [1] "The correct classification rate is:"
```

```r
correct.rate1
```

```
## [1] 0.5686275
```

```r
# Note: difference between R function fitted() and predict()
# https://stackoverflow.com/questions/12201439/is-there-a-difference-between-the-r-functions-fitted-and
```

7.12(b) Repeat the above exercise by fitting an ordinal logistic regression model. Do you get better predictions?

```r
library(ordinal)
pregnancy$Duration.ordered = ordered(pregnancy$Duration,levels=c(1,2,3), labels=c(1,2,3))
pregnancy$Duration.ordered = as.ordered(pregnancy$Duration)
pregnancy$Age.factor <-as.factor(pregnancy$Age)
# Now that the original datafram is modified, we redefine train and test set
train <-pregnancy[train_indices, ]
test <-pregnancy[-train_indices, ]
#fit the ordinal logistic regression model
fit_ordinal <- clm(Duration.ordered ~ Nutrition+Alcohol+Smoking+Age.factor, data=train)
summary(fit_ordinal)
```

```
## formula: Duration.ordered ~ Nutrition + Alcohol + Smoking + Age.factor
## data:    train
##
##  link  threshold nobs logLik AIC    niter max.grad cond.H
##  logit flexible  51   -45.65 105.30 5(0)  1.34e-12 1.5e+06
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## Nutrition    0.02954    0.01383   2.136  0.03266 *
## Alcohol     -1.62121    0.64727  -2.505  0.01226 *
## Smoking     -1.80054    0.61530  -2.926  0.00343 **
## Age.factor2  1.37253    0.74461   1.843  0.06529 .
## Age.factor3  0.82917    0.92807   0.893  0.37163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 1|2    1.908      1.810   1.054
## 2|3    3.934      1.874   2.099
```

```r
# Now make predictions using the fitted model of ordinal logsitic regression
Y.prob.2 = predict(fit_ordinal, newdata=test[,c(2,3,4,6)])$fit;
# now get the predicted class using the maximum probability rule
n=nrow(test)
Y.hat.1 = rep(0,n)
for(i in 1:nrow(test)) {if(max(Y.prob.2[i,])==Y.prob.2[i,1]){Y.hat.1[i]=1;}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,2]) {Y.hat.1[i]=2;}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,3]) {Y.hat.1[i]=3;}
```

```r
}
Y.hat.1
```

```
##  [1] 3 1 1 2 2 1 1 3 3 2 1 2 2 3 2 2 2 2 3 2 3 2 2 2 2 1 3 2 1 1 3 3 3 3 3
## [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1
```

```r
ctable1 = table(test$Duration,Y.hat.1)
ctable1
```

```
##    Y.hat.1
##      1  2  3
##   1  5  5  3
##   2  3 10  4
##   3  2  0 19
```

```r
# calcualte correct classification rate
correct.rate1=sum(diag(ctable1)[1:3])/nrow(test);
print("The correct classification rate for the ordinal logistic regression model is:")
```

```
## [1] "The correct classification rate for the ordinal logistic regression model is:"
```

```r
correct.rate1
```

```
## [1] 0.6666667
```

7.13 (Mammography testing history)

7/13(a) Fit a nominal logistic regression model to the training set and make predictions for the test set using the maximum probability rule. What is the correct classification rate and how does it break down among the threee categories>?

```r
#import the data
setwd("/Users/jiedali/Documents/courses/401_predictive_analysis/homeworks/hw7/")
mam <-read.csv("./Mammography.csv")
head(mam)
```

```
##   OBS ME PB HIST
## 1   1  0  7    0
## 2   2  0 11    0
## 3   3  0  8    1
## 4   4  2 11    0
## 5   5  1  7    0
## 6   6  0  7    0
```

```r
#All odd-numbered observations into the training set and all even-numbered observations into test set
train_indices<-seq(1,nrow(mam),by=2)
test_indices<-seq(2,nrow(mam),by=2)
train <-mam[train_indices, ]
```

```r
test <-mam[-train_indices, ]
#Fit nominal logistic regression model
library(nnet)
fit_nominal <-multinom(ME ~ PB+HIST, data=train, maxit=1000)
```

```
## # weights:  12 (6 variable)
## initial  value 226.314131
## iter  10 value 181.623142
## final  value 181.623113
## converged
```

```r
summary(fit_nominal)
```

```
## Call:
## multinom(formula = ME ~ PB + HIST, data = train, maxit = 1000)
##
## Coefficients:
##   (Intercept)         PB     HIST
## 1   0.2424622 -0.2250368 1.064849
## 2   1.0356672 -0.2996926 1.655385
##
## Std. Errors:
##   (Intercept)         PB      HIST
## 1   0.7918821 0.10642547 0.6300210
## 2   0.7153200 0.09834596 0.5267861
##
## Residual Deviance: 363.2462
## AIC: 375.2462
```

```r
# Make predictions for the test set using the maximum probablity rule. What is the CCR and how does it
predicted=predict(fit_nominal, type='probs',newdata=test)
n=nrow(test)
Y.hat.1 = rep(0,n)
for(i in 1:nrow(test)) {if(max(predicted[i,])==predicted[i,1]){Y.hat.1[i]=0;}
  else if(max(predicted[i,])==predicted[i,2]) {Y.hat.1[i]=1;}
  else if(max(predicted[i,])==predicted[i,3]) {Y.hat.1[i]=2;}
}
Y.hat.1
```

```
##   [1] 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
ctable1 = table(test$ME,Y.hat.1)
ctable1
```

```
##    Y.hat.1
##       0   2
```

```
##   0 106   4
##   1  37   4
##   2  48   7
```

```r
# calcualte correct classification rate
correct.rate1=(106+7)/nrow(test);
print("The correct classification rate is:")
```

```
## [1] "The correct classification rate is:"
```

```r
correct.rate1
```

```
## [1] 0.5485437
```

```r
# How does the CCR break down among the three categories?
print("print the confusion matrix")
```

```
## [1] "print the confusion matrix"
```

```r
print(ctable1)
```

```
##    Y.hat.1
##       0   2
##   0 106   4
##   1  37   4
##   2  48   7
```

```r
# Based on the confusion matrix
print("CCR for outcome 0")
```

```
## [1] "CCR for outcome 0"
```

```r
print(106/(106+4))
```

```
## [1] 0.9636364
```

```r
print("CCR for outcome 1")
```

```
## [1] "CCR for outcome 1"
```

```r
print(0/(37+4))
```

```
## [1] 0
```

```r
print("CCR for outcome 2")
```

```
## [1] "CCR for outcome 2"
```

```r
print(7/(48+7))
```

```
## [1] 0.1272727
```

7.13(b) Repeat the above exercise by fitting an ordinal logistic regression model. Make sure that you order the responses so that 0<2<1. Do you get better predictions?

```r
#Order the responses so that 0<2<1
mam$ME.ordered = ordered(mam$ME, levels=c(0,2,1))
#All odd-numbered observations into the training set and all even-numbered observations into test set
train_indices<-seq(1,nrow(mam),by=2)
test_indices<-seq(2,nrow(mam),by=2)
train <-mam[train_indices, ]
test <-mam[-train_indices, ]
#
library(ordinal)
#fit the ordinal logistic regression model
fit_ordinal <- clm(ME.ordered ~ PB+HIST, data=train, maxit=1000)
summary(fit_ordinal)
```

```
## formula: ME.ordered ~ PB + HIST
## data:    train
##
##  link  threshold nobs logLik  AIC    niter max.grad cond.H
##  logit flexible  206  -185.13 378.27 4(0)  7.74e-08 1.8e+03
##
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## PB   -0.23590    0.07667  -3.077  0.00209 **
## HIST  0.91997    0.39039   2.357  0.01845 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 0|2  -1.1809     0.5744  -2.056
## 2|1   0.1461     0.5745   0.254
```

```r
# Now make predictions using the fitted model of ordinal logsitic regression
Y.prob.2 = predict(fit_ordinal, newdata=test[,3:4])$fit;
# now get the predicted class using the maximum probability rule
n=nrow(test)
Y.hat.1 = rep(0,n)
for(i in 1:nrow(test)) {if(max(Y.prob.2[i,])==Y.prob.2[i,1]){Y.hat.1[i]=0;}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,2]) {Y.hat.1[i]=2;}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,3]) {Y.hat.1[i]=1;}
}
Y.hat.1
```

```
##   [1] 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
ctable1 = table(test$ME.ordered,Y.hat.1)
ctable1
```

```
##    Y.hat.1
##       0   1
##   0 107   3
##   2  52   3
##   1  39   2
```

```r
# calcualte correct classification rate
correct.rate1=sum(107+2)/nrow(test);
print("The correct classification rate for the ordinal logistic regression model is:")
```

```
## [1] "The correct classification rate for the ordinal logistic regression model is:"
```

```r
correct.rate1
```

```
## [1] 0.5291262
```

Comment: the CCR for nominal logistic regression is 54.85%, CCR for ordinal logistic regression is 52.91%.

7.14 (Program choices by high school students)

7.14 (a) Fit a nominal logistic regression model and use it to calculate the probabilities of three program choices for a male student from high ses and a private scholl with median scores on four tests: reading=50, writing=54, math=52 and science =53 (note that some of these predictors may not be in the final model). Which choice is this student likely to make?

```r
program <-read.csv("./program.csv")
head(program)
```

```
##    id gender    ses schtyp     prog read write math science
## 1  45 female    low public vocation   34    35   41      29
## 2 108   male middle public  general   34    33   41      36
## 3  15   male   high public vocation   39    39   44      26
## 4  67   male    low public vocation   37    37   42      33
## 5 153   male middle public vocation   39    31   40      39
## 6  51 female   high public  general   42    36   42      31
```

```
library(nnet)
fit_nominal <-multinom(prog ~ factor(gender)+factor(ses)+factor(schtyp)+read+write+math+science, data=pr
```

```
## # weights:  30 (18 variable)
## initial  value 219.722458
## iter  10 value 180.870401
## iter  20 value 159.648982
## final  value 159.589251
## converged
```

```
summary(fit_nominal)
```

```
## Call:
## multinom(formula = prog ~ factor(gender) + factor(ses) + factor(schtyp) +
##     read + write + math + science, data = program, maxit = 1000)
##
## Coefficients:
##          (Intercept) factor(gender)male factor(ses)low factor(ses)middle
## general     2.741092         -0.1327268      1.0320833          0.684584
## vocation    6.352510         -0.3429153      0.3442851          1.158928
##          factor(schtyp)public        read       write        math
## general             0.5501422 -0.05224702 -0.03412062 -0.09883759
## vocation            1.8902942 -0.05630321 -0.06191444 -0.12284845
##             science
## general  0.10016269
## vocation 0.06042398
##
## Std. Errors:
##          (Intercept) factor(gender)male factor(ses)low factor(ses)middle
## general     1.689618          0.4506350      0.5693775         0.4975331
## vocation    1.944370          0.4843302      0.6550376         0.5510320
##          factor(schtyp)public       read      write       math    science
## general             0.5504279 0.02910921 0.03145610 0.03379963 0.03067642
## vocation            0.8107251 0.03234417 0.03269864 0.03733624 0.03155476
##
## Residual Deviance: 319.1785
## AIC: 355.1785
```

```
# Run stepwise logistic regression to select the best model
fit_nominal<-step(fit_nominal, scope=~ prog ~ factor(gender)+factor(ses)+factor(schtyp)+read+write+math+
```

```
## Start:  AIC=355.18
## prog ~ factor(gender) + factor(ses) + factor(schtyp) + read +
##     write + math + science
##
## trying - factor(gender)
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 180.818124
## iter  20 value 159.843366
## final  value 159.841915
## converged
```

```
## trying - factor(ses)
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 178.290978
## iter  20 value 164.598770
## final  value 164.598741
## converged
## trying - factor(schtyp)
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 189.011967
## iter  20 value 163.336192
## final  value 163.332286
## converged
## trying - read
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 170.515601
## iter  20 value 161.861355
## final  value 161.858610
## converged
## trying - write
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 170.903728
## iter  20 value 161.490061
## final  value 161.476595
## converged
## trying - math
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 175.175435
## iter  20 value 167.196573
## final  value 167.194098
## converged
## trying - science
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 177.505367
## iter  20 value 165.646971
## final  value 165.645292
## converged
##                  Df      AIC
## - factor(gender) 16 351.6838
## - write          16 354.9532
## <none>           18 355.1785
## - read           16 355.7172
## - factor(ses)    14 357.1975
## - factor(schtyp) 16 358.6646
## - science        16 363.2906
## - math           16 366.3882
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 180.818124
```

```
## iter  20 value 159.843366
## final  value 159.841915
## converged
##
## Step:  AIC=351.68
## prog ~ factor(ses) + factor(schtyp) + read + write + math + science
##
## trying - factor(ses)
## # weights:  21 (12 variable)
## initial  value 219.722458
## iter  10 value 177.373108
## final  value 164.924274
## converged
## trying - factor(schtyp)
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 188.653507
## iter  20 value 163.564164
## final  value 163.564109
## converged
## trying - read
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 170.613054
## iter  20 value 162.298847
## final  value 162.298822
## converged
## trying - write
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 170.620432
## iter  20 value 161.495973
## final  value 161.495961
## converged
## trying - math
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 175.317462
## iter  20 value 167.613407
## final  value 167.613391
## converged
## trying - science
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 177.432541
## iter  20 value 165.939693
## final  value 165.939607
## converged
## trying + factor(gender)
## # weights:  30 (18 variable)
## initial  value 219.722458
## iter  10 value 180.870401
## iter  20 value 159.648982
## final  value 159.589251
```

```
## converged
##                  Df      AIC
## - write           14 350.9919
## <none>            16 351.6838
## - read            14 352.5976
## - factor(ses)     12 353.8485
## - factor(schtyp)  14 355.1282
## + +factor(gender) 18 355.1785
## - science         14 359.8792
## - math            14 363.2268
## # weights:  24 (14 variable)
## initial  value 219.722458
## iter  10 value 170.620432
## iter  20 value 161.495973
## final  value 161.495961
## converged
##
## Step:  AIC=350.99
## prog ~ factor(ses) + factor(schtyp) + read + math + science
##
## trying - factor(ses)
## # weights:  18 (10 variable)
## initial  value 219.722458
## iter  10 value 168.461544
## final  value 166.832303
## converged
## trying - factor(schtyp)
## # weights:  21 (12 variable)
## initial  value 219.722458
## iter  10 value 172.228881
## final  value 165.864396
## converged
## trying - read
## # weights:  21 (12 variable)
## initial  value 219.722458
## iter  10 value 167.071471
## final  value 165.099779
## converged
## trying - math
## # weights:  21 (12 variable)
## initial  value 219.722458
## iter  10 value 173.351020
## final  value 172.156655
## converged
## trying - science
## # weights:  21 (12 variable)
## initial  value 219.722458
## iter  10 value 170.853755
## final  value 166.993157
## converged
## trying + factor(gender)
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 170.903728
```

```
## iter  20 value 161.490061
## final  value 161.476595
## converged
## trying + write
## # weights:  27 (16 variable)
## initial  value 219.722458
## iter  10 value 180.818124
## iter  20 value 159.843366
## final  value 159.841915
## converged
##                   Df      AIC
## <none>            14 350.9919
## + +write          16 351.6838
## - factor(ses)     10 353.6646
## - read            12 354.1996
## + +factor(gender) 16 354.9532
## - factor(schtyp)  12 355.7288
## - science         12 357.9863
## - math            12 368.3133
```

```r
# print summary of model after stepwise regression
summary(fit_nominal)
```

```
## Call:
## multinom(formula = prog ~ factor(ses) + factor(schtyp) + read +
##     math + science, data = program, maxit = 1000)
##
## Coefficients:
##          (Intercept) factor(ses)low factor(ses)middle factor(schtyp)public
## general     2.101981      1.0357881          0.712673            0.6082977
## vocation    5.302734      0.3344098          1.194434            2.0165158
##                 read       math    science
## general  -0.05912833 -0.1074597 0.09077515
## vocation -0.07131321 -0.1382363 0.04306943
##
## Std. Errors:
##          (Intercept) factor(ses)low factor(ses)middle factor(schtyp)public
## general     1.578024      0.5648603         0.4975808            0.5484780
## vocation    1.818154      0.6434123         0.5453637            0.8100331
##                read       math    science
## general  0.02807080 0.03270789 0.02859517
## vocation 0.03106518 0.03601493 0.02920538
##
## Residual Deviance: 322.9919
## AIC: 350.9919
```

```r
# Make predictions for the test set using the maximum probablity rule. What is the CCR and how does it
predicted=predict(fit_nominal, type='probs',newdata=program)
n=nrow(program)
Y.hat.1_nominal = rep(0,n)
for(i in 1:nrow(program)) {if(max(predicted[i,])==predicted[i,1]){Y.hat.1_nominal[i]='academic';}
  else if(max(predicted[i,])==predicted[i,2]) {Y.hat.1_nominal[i]='general';}
  else if(max(predicted[i,])==predicted[i,3]) {Y.hat.1_nominal[i]='vocation';}
```

```
}
ctable1_nominal = table(program$prog,Y.hat.1_nominal)
ctable1_nominal
```

```
##           Y.hat.1_nominal
##            academic general vocation
##   academic       87       8       10
##   general        26       8       11
##   vocation       17       4       29
```

```
# calcualte correct classification rate
correct.rate1_nominal=sum(diag(ctable1_nominal)[1:3])/nrow(program);
print("The correct classification rate is:")
```

```
## [1] "The correct classification rate is:"
```

```
correct.rate1_nominal
```

```
## [1] 0.62
```

calculate the probabilities of three program choices for a male student from high ses and a private scholl with median scores on four tests: reading=50, writing=54, math=52 and science =53, which program is he likely to choose?

```
#
predicted=predict(fit_nominal,type='probs',newdata=data.frame(gender='male',ses='high',schtyp='private'
print("the predicted probabilities are:")
```

```
## [1] "the predicted probabilities are:"
```

```
print(predicted)
```

```
##   academic    general   vocation
## 0.80791865 0.15809598 0.03398537
```

Comment: based on the maximum probability rule, the choice that he is likely to make is "academic".

7.14(b) Repeat the above for the ordinal logistic regression model. Compare the results for the two models, in particular, with respect to the predictors in the final model and their interpretations.

```
#reimport data
program <-read.csv("./program.csv")
head(program)
```

```
##    id gender    ses schtyp     prog read write math science
## 1  45 female    low public vocation   34    35   41      29
## 2 108   male middle public  general   34    33   41      36
## 3  15   male   high public vocation   39    39   44      26
## 4  67   male    low public vocation   37    37   42      33
## 5 153   male middle public vocation   39    31   40      39
## 6  51 female   high public  general   42    36   42      31
```

```
#
library(ordinal)
#fit the ordinal logistic regression model
fit_ordinal <- clm(prog ~ factor(gender)+factor(ses)+factor(schtyp)+read+write+math+science, data=progra
#Run stepwise regression
fit_ordinal<-step(fit_ordinal, scope=~ prog ~ factor(gender)+factor(ses)+factor(schtyp)+read+write+math+
```

```
## Start:  AIC=348.18
## prog ~ factor(gender) + factor(ses) + factor(schtyp) + read +
##     write + math + science
##
##                  Df    AIC
## - factor(gender)  1 346.67
## <none>              348.18
## - factor(ses)     2 349.85
## - read            1 350.35
## - write           1 350.40
## - factor(schtyp)  1 352.51
## - science         1 352.74
## - math            1 362.60
##
## Step:  AIC=346.67
## prog ~ factor(ses) + factor(schtyp) + read + write + math + science
##
##                  Df    AIC
## <none>              346.67
## + factor(gender)  1 348.18
## - factor(ses)     2 348.44
## - write           1 348.46
## - read            1 349.52
## - science         1 350.76
## - factor(schtyp)  1 350.86
## - math            1 361.21
```

```
summary(fit_ordinal)
```

```
## formula:
## prog ~ factor(ses) + factor(schtyp) + read + write + math + science
## data:    program
##
##  link  threshold nobs logLik  AIC    niter max.grad cond.H
##  logit flexible  200  -164.33 346.67 5(0)  2.34e-12 1.6e+06
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
```

```
## factor(ses)low        0.42054    0.44497   0.945 0.344609
## factor(ses)middle     0.89023    0.38802   2.294 0.021774 *
## factor(schtyp)public  1.12568    0.47497   2.370 0.017787 *
## read                 -0.04873    0.02238  -2.177 0.029493 *
## write                -0.04064    0.02093  -1.942 0.052155 .
## math                 -0.09883    0.02551  -3.874 0.000107 ***
## science               0.05306    0.02193   2.420 0.015539 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                   Estimate Std. Error z value
## academic|general    -5.490      1.308  -4.197
## general|vocation    -4.102      1.278  -3.211
```

```
# Now make predictions using the fitted model of ordinal logsitic regression
Y.prob.2 = predict(fit_ordinal, newdata=program[,c(3,4,6,7,8,9)])$fit;
# now get the predicted class using the maximum probability rule
n=nrow(program)
Y.hat.1 = rep(0,n)
for(i in 1:nrow(program)) {if(max(Y.prob.2[i,])==Y.prob.2[i,1]){Y.hat.1[i]='academic';}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,2]) {Y.hat.1[i]='general';}
  else if(max(Y.prob.2[i,])==Y.prob.2[i,3]) {Y.hat.1[i]='vocation';}
}
ctable1_ord = table(program$prog,Y.hat.1)
ctable1_ord
```

```
##           Y.hat.1
##            academic vocation
##   academic       94       11
##   general        26       19
##   vocation       17       33
```

```
# calcualte correct classification rate
correct.rate1_ord=sum(94+33)/nrow(program);
print("The correct classification rate for the ordinal logistic regression model is:")
```

```
## [1] "The correct classification rate for the ordinal logistic regression model is:"
```

```
correct.rate1_ord
```

```
## [1] 0.635
```

Commet: the ordinal logisitc regression with stepwise regression has eliminated the "gender" predictor variable.

Commet: Comparing the two models given by nominal logistic regression and ordinal logistic regression, nominal logistic regression dropped predictor variable: factor(gender), write; Ordinal logistic regression model dropped only factor(gender). The resuling CCR, ordinal logistic regression (0.635) is a bit higher than nominal logistic regression (0.62). It makes sense since the ordinal model has one more predictor variable than the nominal model.

##7.14(c) Compute the classification matrices for the two models and the CCRs. Which model gives higher CCR.

```
#
print("print the confusion matrix for nominal case:")
```

```
## [1] "print the confusion matrix for nominal case:"
```

```
ctable1_nominal
```

```
##           Y.hat.1_nominal
##            academic general vocation
##   academic       87       8       10
##   general        26       8       11
##   vocation       17       4       29
```

```
# calcualte correct classification rate
correct.rate1_nominal=sum(diag(ctable1_nominal)[1:3])/nrow(program);
print("CCR for nominal case is:")
```

```
## [1] "CCR for nominal case is:"
```

```
correct.rate1_nominal
```

```
## [1] 0.62
```

```
print("print the confusion matrix for ordinal case:")
```

```
## [1] "print the confusion matrix for ordinal case:"
```

```
ctable1_ord
```

```
##           Y.hat.1
##            academic vocation
##   academic       94       11
##   general        26       19
##   vocation       17       33
```

```r
# calcualte correct classification rate
correct.rate1_ord=sum(94+33)/nrow(program);
print("The correct classification rate for the ordinal logistic regression model is:")
```

```
## [1] "The correct classification rate for the ordinal logistic regression model is:"
```

```r
correct.rate1_ord
```

```
## [1] 0.635
```

commmets: The resuling CCR, ordinal logistic regression (0.635) is a bit higher than nominal logistic regression (0.62). It makes sense since the ordinal model has one more predictor variable than the nominal model.