Stat 536 HW1

Dongyang Wang

2022-09-29

Question 1

```
library(foreign)
df <- read.dta("MROZ.dta")
df <- df[,c(1,3:6,9:19,22)]
```

The wage of the non-working women is zero, hence the variables wage and lwage (columns 7 and 21) will contain missing values we eliminate these variables from the data. Also remove hours, repwage, and nwifeinc because they seem irrelevant to whether a woman will be in the labor force.

Question 2

```
df[,"lage"] = log(df[,"age"])
df[,"lhushrs"] = log(df[,"hushrs"])
df <-df[ , !(names(df) %in% c("age", "hushrs"))]
```

Transforming two variables and dropping original ones.

Question 3

```
for(i in 2:17)
{
    cat(colnames(df)[i],"[",i,"] = ",cor(df[,1],df[,i]),"\n");
}
```

```
## kidslt6 [ 2 ] =  -0.2137493
## kidsge6 [ 3 ] =  -0.002424231
## educ [ 4 ] =  0.1873528
## husage [ 5 ] =  -0.07282005
## huseduc [ 6 ] =  0.04591422
## huswage [ 7 ] =  -0.06947526
## faminc [ 8 ] =  0.09889538
## mtr [ 9 ] =  -0.1448255
## motheduc [ 10 ] =  0.09048973
## fatheduc [ 11 ] =  0.05771841
## unem [ 12 ] =  -0.02873489
## city [ 13 ] =  -0.006167593
## exper [ 14 ] =  0.3424847
## expersq [ 15 ] =  0.2607407
## lage [ 16 ] =  -0.07226078
## lhushrs [ 17 ] =  -0.06160755
```

Based on the results, the highest correlation turns out to be experience. I will implement a heuristic model based on exper.

Question 4

```
#the inverse of the logit function
inverseLogit <- function(x)
{
  return(exp(x)/(1+exp(x)));
}

#function for the computation of the Hessian
inverseLogit2 <- function(x)
{
  return(exp(x)/(1+exp(x))^2);
}

#computes pi_i = P(y_i = 1 | x_i)
getPi <- function(x,beta)
{
  x0 = cbind(rep(1,length(x)),x);
  return(inverseLogit(x0%*%beta));
}

#another function for the computation of the Hessian
getPi2 <- function(x,beta)
{
  x0 = cbind(rep(1,length(x)),x);
  return(inverseLogit2(x0%*%beta));
}

#logistic log-likelihood (formula (3) in your handout)
logisticLoglik <- function(y,x,beta)
{
  Pi = getPi(x,beta);
  return(sum(y*log(Pi))+sum((1-y)*log(1-Pi)));
}

#obtain the gradient for Newton-Raphson
getGradient <- function(y,x,beta)
{
  gradient = matrix(0,2,1);
  Pi = getPi(x,beta);

  gradient[1,1] = sum(y-Pi);
  gradient[2,1] = sum((y-Pi)*x);

  return(gradient);
}

#obtain the Hessian for Newton-Raphson
getHessian <- function(y,x,beta)
{
  hessian = matrix(0,2,2);
```

```r
  Pi2 = getPi2(x,beta);

  hessian[1,1] = sum(Pi2);
  hessian[1,2] = sum(Pi2*x);
  hessian[2,1] = hessian[1,2];
  hessian[2,2] = sum(Pi2*x^2);

  return(-hessian);
}

#this function implements our own Newton-Raphson procedure
getcoefNR <- function(response,explanatory,data)
{
  #2x1 matrix of coefficients`
  beta = matrix(0,2,1);
  y = data[,response];
  x = data[,explanatory];

  #current value of log-likelihood
  currentLoglik = logisticLoglik(y,x,beta);

  #infinite loop unless we stop it someplace inside
  while(1)
  {
    newBeta = beta - solve(getHessian(y,x,beta))%*%getGradient(y,x,beta);
    newLoglik = logisticLoglik(y,x,newBeta);

    #at each iteration the log-likelihood must increase
    if(newLoglik<currentLoglik)
    {
      cat("CODING ERROR!!\n");
      break;
    }
    beta = newBeta;
    #stop if the log-likelihood does not improve by too much
    if(newLoglik-currentLoglik<1e-6)
    {
      break;
    }
    currentLoglik = newLoglik;
  }

  return(beta);
}
```

```r
m_0 <- glm(inlf~exper,family=binomial(link=logit),data=df)

coef_nr = getcoefNR(1, "exper", df)
coef_mle = coef(m_0)
coef_nr
```

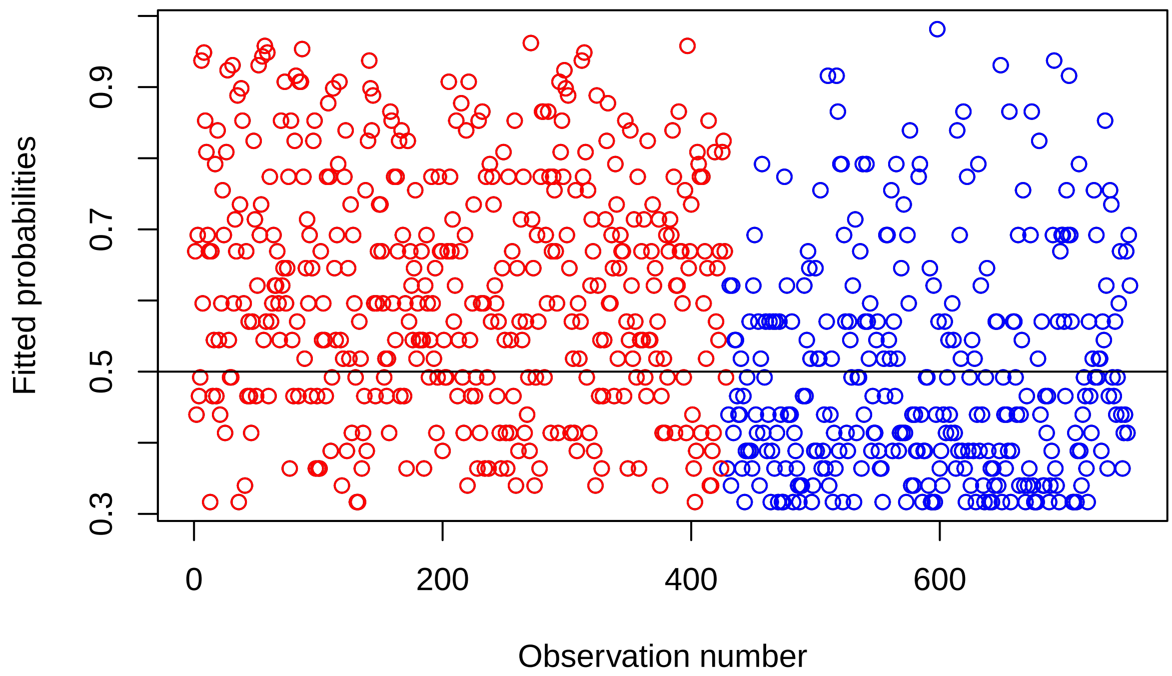
```
##            [,1]
## [1,] -0.7692075
## [2,]  0.1052530
```

```
coef_mle
```

```
## (Intercept)       exper
##  -0.7692075   0.1052530
```

Based on Newton-Raphson algorithm, the MLEs are calculated correctly. A formula could be $logit(inlf) = -0.7692075 + 0.1052530 * exper$. The validity will be shown in plots as follows.

```
myind = 1:length(df$inlf)
plot(myind,m_0$fitted.values,xlab="Observation number",ylab="Fitted probabilities")
points(myind[df$inlf==0],m_0$fitted.values[df$inlf==0],col="blue")
points(myind[df$inlf==1],m_0$fitted.values[df$inlf==1],col="red")
abline(h=0.5)
```

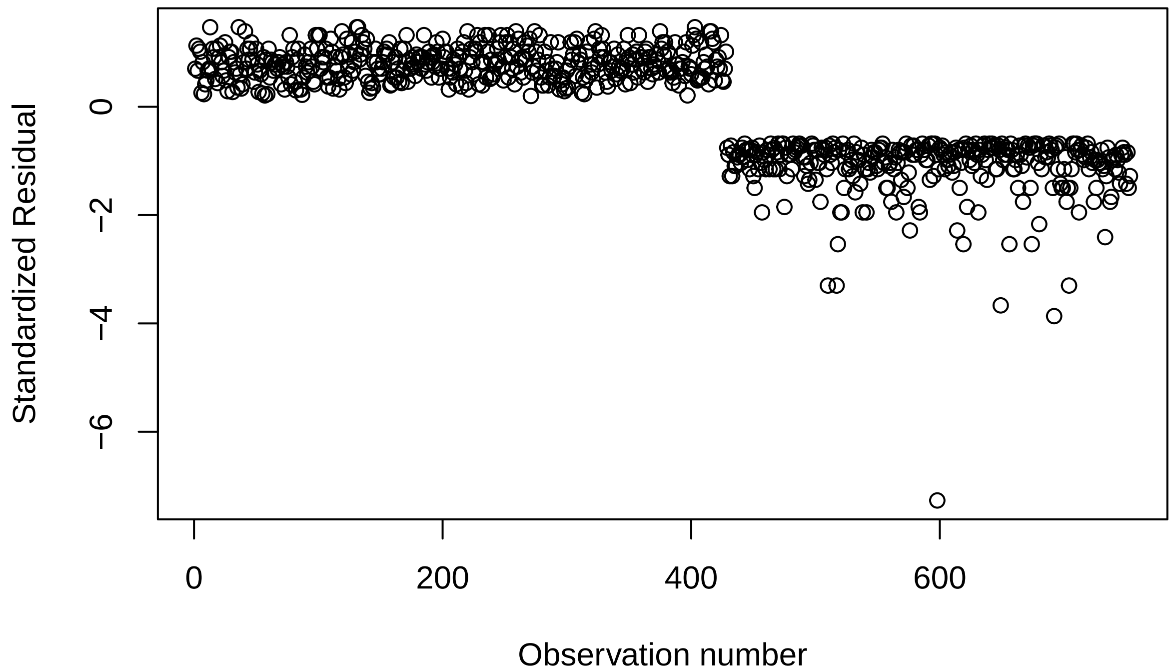



```
#determine the standardized residuals
myres = (df$inlf-m_0$fitted.values)/sqrt(m_0$fitted.values*(1-m_0$fitted.values))

#calculate the p-value for the chisq test
1-pchisq(sum(myres^2),length(df$inlf)-length(coef(m_0)))
```

```
## [1] 0.04661837
```

```
#make an index plot of standardized residuals against observation number
plot(1:length(df$inlf),myres,xlab="Observation number",ylab="Standardized Residual")
```



It turns out that the model does not fit the data well. For the fitted values, the prediction with a threshold of 0.5 show very slight predictive power. For the chi-squared test, the p value shows that the residuals tend to fall out in the extreme part of the distribution, therefore marking a bad fit. Easily seen from the plot, there are standarized residuals exceeding -2, indicating poor fit.

Question 5

```
my_logit <- glm(inlf~.,family=binomial(link=logit),data=df)
my_model <- step(my_logit,trace=TRUE)
```

```
## Start:  AIC=761.22
## inlf ~ kidslt6 + kidsge6 + educ + husage + huseduc + huswage +
##     faminc + mtr + motheduc + fatheduc + unem + city + exper +
##     expersq + lage + lhushrs
##
##            Df Deviance    AIC
## - motheduc  1   727.22 759.22
## - city      1   727.22 759.22
## - fatheduc  1   727.23 759.23
## - unem      1   727.71 759.71
## - husage    1   727.93 759.93
## <none>          727.22 761.22
## - huseduc   1   729.65 761.65
## - faminc    1   730.39 762.39
## - kidsge6   1   732.22 764.22
## - lage      1   736.80 768.80
## - expersq   1   737.01 769.01
## - educ      1   740.59 772.59
## - mtr       1   752.02 784.02
## - exper     1   765.24 797.24
## - lhushrs   1   768.53 800.53
## - kidslt6   1   770.35 802.35
```

```
## - huswage   1   788.02 820.02
##
## Step:  AIC=759.22
## inlf ~ kidslt6 + kidsge6 + educ + husage + huseduc + huswage +
##     faminc + mtr + fatheduc + unem + city + exper + expersq +
##     lage + lhushrs
##
##            Df Deviance    AIC
## - city      1   727.22 757.22
## - fatheduc  1   727.23 757.23
## - unem      1   727.71 757.71
## - husage    1   727.94 757.94
## <none>          727.22 759.22
## - huseduc   1   729.65 759.65
## - faminc    1   730.39 760.39
## - kidsge6   1   732.23 762.23
## - lage      1   736.82 766.82
## - expersq   1   737.01 767.01
## - educ      1   741.27 771.27
## - mtr       1   752.06 782.06
## - exper     1   765.24 795.24
## - lhushrs   1   768.64 798.64
## - kidslt6   1   770.53 800.53
## - huswage   1   788.18 818.18
##
## Step:  AIC=757.22
## inlf ~ kidslt6 + kidsge6 + educ + husage + huseduc + huswage +
##     faminc + mtr + fatheduc + unem + exper + expersq + lage +
##     lhushrs
##
##            Df Deviance    AIC
## - fatheduc  1   727.23 755.23
## - unem      1   727.72 755.72
## - husage    1   727.94 755.94
## <none>          727.22 757.22
## - huseduc   1   729.67 757.67
## - faminc    1   730.39 758.39
## - kidsge6   1   732.23 760.23
## - lage      1   736.87 764.87
## - expersq   1   737.03 765.03
## - educ      1   741.28 769.28
## - mtr       1   752.06 780.06
## - exper     1   765.30 793.30
## - lhushrs   1   768.66 796.66
## - kidslt6   1   770.53 798.53
## - huswage   1   789.15 817.15
##
## Step:  AIC=755.23
## inlf ~ kidslt6 + kidsge6 + educ + husage + huseduc + huswage +
##     faminc + mtr + unem + exper + expersq + lage + lhushrs
##
##           Df Deviance    AIC
## - unem     1   727.72 753.72
## - husage   1   727.94 753.94
```

```
## <none>         727.23 755.23
## - huseduc  1   729.68 755.68
## - faminc   1   730.39 756.39
## - kidsge6  1   732.24 758.24
## - expersq  1   737.03 763.03
## - lage     1   737.06 763.06
## - educ     1   742.65 768.65
## - mtr      1   752.09 778.09
## - exper    1   765.31 791.31
## - lhushrs  1   768.66 794.66
## - kidslt6  1   770.53 796.53
## - huswage  1   789.15 815.15
##
## Step:  AIC=753.72
## inlf ~ kidslt6 + kidsge6 + educ + husage + huseduc + huswage +
##     faminc + mtr + exper + expersq + lage + lhushrs
##
##           Df Deviance    AIC
## - husage   1   728.38 752.38
## <none>         727.72 753.72
## - huseduc  1   730.15 754.15
## - faminc   1   730.96 754.96
## - kidsge6  1   732.55 756.55
## - expersq  1   737.17 761.17
## - lage     1   738.02 762.02
## - educ     1   742.86 766.86
## - mtr      1   752.82 776.82
## - exper    1   765.33 789.33
## - lhushrs  1   768.69 792.69
## - kidslt6  1   771.19 795.19
## - huswage  1   791.60 815.60
##
## Step:  AIC=752.38
## inlf ~ kidslt6 + kidsge6 + educ + huseduc + huswage + faminc +
##     mtr + exper + expersq + lage + lhushrs
##
##           Df Deviance    AIC
## <none>         728.38 752.38
## - huseduc  1   730.62 752.62
## - faminc   1   731.54 753.54
## - kidsge6  1   733.54 755.54
## - expersq  1   737.69 759.69
## - educ     1   743.54 765.54
## - mtr      1   753.64 775.64
## - exper    1   766.08 788.08
## - lhushrs  1   768.97 790.97
## - kidslt6  1   771.19 793.19
## - lage     1   773.19 795.19
## - huswage  1   792.26 814.26
```

```
summary(my_model)
```

```
##
## Call:
## glm(formula = inlf ~ kidslt6 + kidsge6 + educ + huseduc + huswage +
```

```
##     faminc + mtr + exper + expersq + lage + lhushrs, family = binomial(link = logit),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5044  -0.7847   0.3402   0.7462   2.6714
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.478e+01  5.666e+00   7.903 2.72e-15 ***
## kidslt6     -1.307e+00  2.126e-01  -6.146 7.92e-10 ***
## kidsge6      1.825e-01  8.089e-02   2.256  0.02407 *
## educ         2.051e-01  5.361e-02   3.826  0.00013 ***
## huseduc     -6.278e-02  4.213e-02  -1.490  0.13617
## huswage     -3.550e-01  5.007e-02  -7.091 1.33e-12 ***
## faminc       3.280e-05  1.830e-05   1.792  0.07308 .
## mtr         -1.475e+01  3.058e+00  -4.824 1.41e-06 ***
## exper        2.079e-01  3.425e-02   6.071 1.27e-09 ***
## expersq     -3.376e-03  1.095e-03  -3.084  0.00204 **
## lage        -4.131e+00  6.516e-01  -6.339 2.31e-10 ***
## lhushrs     -2.642e+00  4.548e-01  -5.808 6.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.75  on 752  degrees of freedom
## Residual deviance:  728.38  on 741  degrees of freedom
## AIC: 752.38
##
## Number of Fisher Scoring iterations: 5
```

The final model based on AIC is shown above.

Question 6

```r
# retain all variables from previously chosen step model
vars <- c("inlf", "kidslt6", "kidsge6", "educ", "huseduc", "huswage", "faminc",
          "mtr", "exper", "expersq", "lage", "lhushrs")
len_vars <-length(vars)
vars_ind <- c(2:len_vars)
df1 = df[, vars]

# initialize some vectors
p_val = c()
model_bic = c()
model_brier = c()

for (i in 1:(len_vars - 1)){
  vars_ind1 = vars_ind[-i]
  model_temp = glm(inlf ~ as.matrix(df[, vars_ind1]), family=binomial(link=logit),data=df1)

  # calculate the p-value for the likelihood ratio test
  p_val = c(p_val, 1-pchisq(model_temp$deviance - my_model$deviance , 1))
```

```
  # BIC
  model_bic = c(model_bic, model_temp$deviance+model_temp$rank*log(length(df1$inlf)))

  # determine the residuals
  my_res = df1$inlf-model_temp$fitted.values

  # Brier
  model_brier = c(model_brier, sum(my_res^2))
}

p_val
```

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

Since all the p-values from the likelihood ratio tests are 0, we favor the more complex model, which we have determined using the step function.

Question 7

```
model_bic = c(model_bic, my_model$deviance+my_model$rank*log(length(df$inlf)))

model_bic
```

```
##  [1] 1005.7721  952.9126  976.2644  976.2718  958.4562  989.9190  953.0082
##  [8]  968.6053  952.9173  953.1183  952.9190  807.8680
```

Still, the original step model contains lowest BIC.

Question 8

```
# determine the standardized residuals
my_res = df$inlf-my_model$fitted.values

# Brier
model_brier = c(model_brier, sum(my_res^2))

model_brier
```

```
##  [1] 161.2013 149.6329 154.6715 155.1498 150.5562 158.3018 149.6564 153.6521
##  [9] 149.6404 149.7619 149.6227 118.7603
```

Still, the original step model contains lowest Brier number, indicating the best fit among the models.

```
model_res = ifelse(my_model$fitted.values>=0.5, 1, 0)
sum(model_res == df$inlf)/length(df$inlf)
```

```
## [1] 0.7715803
```

The accuracy (predictive error) of my preferred model is 0.7715803.

Question 9

```
summary(my_model)
```

```
##
## Call:
```

```
## glm(formula = inlf ~ kidslt6 + kidsge6 + educ + huseduc + huswage +
##     faminc + mtr + exper + expersq + lage + lhushrs, family = binomial(link = logit),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5044  -0.7847   0.3402   0.7462   2.6714
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.478e+01  5.666e+00   7.903 2.72e-15 ***
## kidslt6     -1.307e+00  2.126e-01  -6.146 7.92e-10 ***
## kidsge6      1.825e-01  8.089e-02   2.256  0.02407 *
## educ         2.051e-01  5.361e-02   3.826  0.00013 ***
## huseduc     -6.278e-02  4.213e-02  -1.490  0.13617
## huswage     -3.550e-01  5.007e-02  -7.091 1.33e-12 ***
## faminc       3.280e-05  1.830e-05   1.792  0.07308 .
## mtr         -1.475e+01  3.058e+00  -4.824 1.41e-06 ***
## exper        2.079e-01  3.425e-02   6.071 1.27e-09 ***
## expersq     -3.376e-03  1.095e-03  -3.084  0.00204 **
## lage        -4.131e+00  6.516e-01  -6.339 2.31e-10 ***
## lhushrs     -2.642e+00  4.548e-01  -5.808 6.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.75  on 752  degrees of freedom
## Residual deviance:  728.38  on 741  degrees of freedom
## AIC: 752.38
##
## Number of Fisher Scoring iterations: 5
```

One thing I learned from this analysis is that the step function can actually provide a good model in terms of AIC, BIC, and Brier score. I gained knowledge in variable selection. $logit(inlf) = 44.78 - 1.307 * kidslt6 + 0.1825 * kidsge6 + 0.2051 * educ - 0.06278 * huseduc - 0.3550 * huswage + 3.280 * 10^{-5} * faminc - 14.75 * mtr + 0.2079 * exper - 3.376 * 10^{-3} * expersq - 4.131 * lage - 2.642 * lhushrs$.

The interpretations are as follows.

The logit of inlf will change by -1.307 with one more kid under 6; The logit of inlf will change by 0.1825 with one more kid from 6-18; The logit of inlf will change by 0.2051 with one more year of schooling; The logit of inlf will change by -0.06278 with one more hour worked by husband.

The logit of inlf will change by -0.3550 with one dollar increase in husband's hourly wage; The logit of inlf will change by 3.280 * 10^{-5} with one dollar increase in family income; The logit of inlf will change by -14.75 with one additional fed. marginal tax rate facing woman.

The logit of inlf will change by 0.2079 with one more year of experience; The logit of inlf will change by - 3.376 * 10^{-3} with one unit increase in squared experience; The logit of inlf will change by - 4.131 with one unit increase in log wage; The logit of inlf will change by -2.642 with one unit increase in log husband hours.

The significant coefficients include kidslt6, kidsge6, educ, huswage, mtr, exper, expersq, lage, lhushrs.