BE7023 Homework 6

Mike Lape

Octobe 31, 2018

```
setwd("C:/Users/lapt3u/Box/UC/Fall_2018/BE7023_Adv_Biostats/adv_biostats/hw_6")
library(MASS)
```

1. Describe the data.
   The birthwt dataset contains 189 rows/observations and 10 columns/variables.
   Each row represents a birth at Baystate Medical Center in 1986. The variables include low an indicator if the birth weight was less than 2.5 kg, and data about the mother and her health history.
   Variables:
   Low: Indicator of birth weight < 2.5 kg [Categorical - binary]
   age: Age of mother in years [Numerical]
   lwt: Mother's weight in pounds at last menstrual period [Numerical]
   race: Mother's race, 1 = white, 2 = black, 3 = other [Categorical]
   smoke: Smoking status during pregnancy [Categorical - binary]
   ptl: Number of previous premature labors [Numerical]
   ht: history of hypertension [Categorical - binary]
   ui: Presence of uterine irritability [Categorical - binary]
   ftv: Number of physician visits during 1st trimester [Numerical]
   bwt: Birth weight in grams [Numerical]

```
# The birthwt dataframe has 10 columns/variables, and 189 rows/observations.
dim(birthwt)
```

```
## [1] 189  10
```

```
# Top 6 rows.
head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
## 91   0  21 124    3     0   0  0  0   0 2622
```

2. Create a new folder by omitting the last column.
   Convert race,smoke, and ht as factors.
   Obtain summary statistics.

```r
# Drop the last column
dat <- birthwt[, - length(birthwt)]

# Convert race, smoke, anad ht to factors
dat$race    <- as.factor(dat$race)
dat$smoke   <- as.factor(dat$smoke)
dat$ht      <- as.factor(dat$ht)

# Get summary stats
summary(dat)
```

```
##       low              age              lwt        race   smoke
##  Min.   :0.0000   Min.   :14.00   Min.   : 80.0   1:96   0:115
##  1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0   2:26   1: 74
##  Median :0.0000   Median :23.00   Median :121.0   3:67
##  Mean   :0.3122   Mean   :23.24   Mean   :129.8
##  3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0
##  Max.   :1.0000   Max.   :45.00   Max.   :250.0
##       ptl         ht            ui              ftv
##  Min.   :0.0000   0:177   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1: 12   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000           Median :0.0000   Median :0.0000
##  Mean   :0.1958           Mean   :0.1481   Mean   :0.7937
##  3rd Qu.:0.0000           3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :3.0000           Max.   :1.0000   Max.   :6.0000
```

3. Fit a logistic regression model with low as the response variable using the new folder and covariates the rest.
   Identify the significant predictors.
   Interpret carefully the coefficients associated with race.
   Check goodness-of-fit carefully defining what the null hypothesis is.

```r
# fit log reg on low
mod <- glm(low ~ ., data = dat, family = binomial)
summary(mod)
```

```
##
## Call:
## glm(formula = low ~ ., family = binomial, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8946  -0.8212  -0.5316   0.9818   2.2125
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.480623   1.196888   0.402  0.68801
## age         -0.029549   0.037031  -0.798  0.42489
```

```
## lwt         -0.015424   0.006919  -2.229  0.02580 *
## race2        1.272260   0.527357   2.413  0.01584 *
## race3        0.880496   0.440778   1.998  0.04576 *
## smoke1       0.938846   0.402147   2.335  0.01957 *
## ptl          0.543337   0.345403   1.573  0.11571
## ht1          1.863303   0.697533   2.671  0.00756 **
## ui           0.767648   0.459318   1.671  0.09467 .
## ftv          0.065302   0.172394   0.379  0.70484
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 201.28  on 179  degrees of freedom
## AIC: 221.28
##
## Number of Fisher Scoring iterations: 4
```

The significant predictors, those with a p-val < 0.05 are
lwt (P-val = 0.02580)
race2 (P-val = 0.01584)
race3 (P-val = 0.04576)
smoke1 (P-val = 0.01957)
ht1 (P-val = 0.00756)

In this model race1 [white] is held as the reference and p-values as well as coefficients are calculated for both race2 [black] (p-val = 0.01584, coeff = 1.272) and race3 [other] (p-val = 0.04576, coeff = 0.880). The positive coefficients indicate that mothers of race2 [black] and race3 [other] have increased odds of giving birth to an underweight baby as compared to race1 [white] mothers.

```r
# Check goodness of fit
res <- mod$deviance
dof <- mod$df.residual
p <- pchisq(res, dof, lower.tail = F)
round(p, 3)
```

```
## [1] 0.122
```

```r
# Our null hypothesis is that our multinomial logistic regression model
# adequately describes the data. Our p value is 0.122 and since it is
# greater than 0.05 we cannot reject the null hypothesis and thus our
# model is a good fit of the data.
```

4. Fit a logistic regression model with low as the response variable and predictors lwt, race, smoke, and ht.

Write the prediction equation.
Identify the most significant predictor.
Find a way to plot P(low =1) as a function of lwt in the presence of various choices of race, smoke, and ht.
Check goodness-of-fit.
Obtain confusion matrix.
Calculate the misclassification rate.

```
mod2 <- glm(low ~ lwt + race + smoke + ht, data = dat, family = binomial)
summary(mod2)
```

```
##
## Call:
## glm(formula = low ~ lwt + race + smoke + ht, family = binomial,
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7751  -0.8747  -0.5712   0.9634   2.1131
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.352049   0.924438   0.381  0.70333
## lwt         -0.017907   0.006799  -2.634  0.00844 **
## race2        1.287662   0.521648   2.468  0.01357 *
## race3        0.943645   0.423382   2.229  0.02583 *
## smoke1       1.071566   0.387517   2.765  0.00569 **
## ht1          1.749163   0.690820   2.532  0.01134 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 208.25  on 183  degrees of freedom
## AIC: 220.25
##
## Number of Fisher Scoring iterations: 4
```

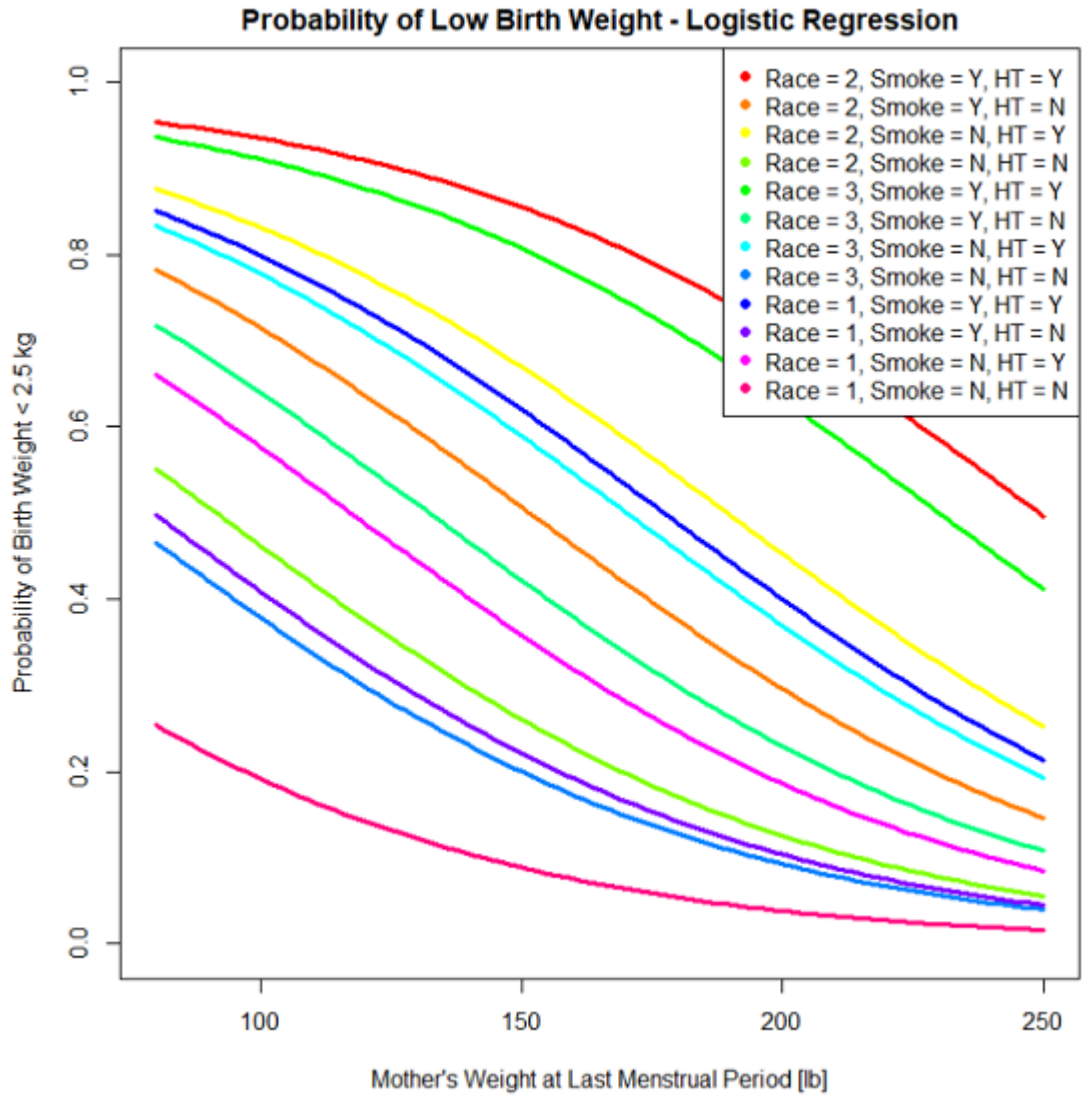
Prediction Equation:
Pr(low birth weight = Yes) = e^(Z) / (1 + e^(Z))

Where: Z = 0.3520 - (0.0179 * lwt) + (1.2877 * race2) + (0.9436 * race3) + (1.0716 * smoke1) + (1.749 * ht1)

All of the predictors are significant, the most significant predictor is smoke1 (Smoking = Yes) with a p-value of 0.00569.

plot P(low =1) as a function of lwt in the presence of various choices of race, smoke, and ht.

For some reason Rmarkdown was not including 4 of the plots so I took a snapshot of it working properly in R



and just included that here.

```
# cols <- palette(rainbow(12))
# # RACE = 2
# # Race = 2, smoke = 1, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#             (1.0716 * 1) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#             (1.0716 * 1) + (1.749 * 1))), from = 80, to = 250, lwd = 3,
#                 col = cols[1], ylim = c(0,1),
#           xlab = "Mother's Weight at Last Menstrual Period [lb]",
#           ylab = "Probability of Birth Weight < 2.5 kg",
#           main = "Probability of Low Birth Weight - Logistic Regression")
#
# # Race = 2, smoke = 1, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 1) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 1) + (1.749 * 0))), col = cols[2], lwd = 3,
#             add = T)
```

```
#
# # Race = 2, smoke = 0, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 1))), col = cols[3], lwd = 3,
#             add = T)
#
# # Race = 2, smoke = 0, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 1) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 0))), col = cols[4], lwd = 3,
#             add = T)
#
# ##RACE = 3
#
# # Race = 3, smoke = 1, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#             (1.0716 * 1) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#             (1.0716 * 1) + (1.749 * 1))), col = cols[5], lwd = 3,
#             add = T)
#
# # Race = 3, smoke = 1, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 1) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 1) + (1.749 * 0))), col = cols[6], lwd = 3,
#             add = T)
#
#
# # Race = 3, smoke = 0, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 0) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 0) + (1.749 * 1))), col = cols[7], lwd = 3,
#             add = T)
#
# # Race = 3, smoke = 0, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 0) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 1) +
#           (1.0716 * 0) + (1.749 * 0))), col = cols[8], lwd = 3,
#             add = T)
#
# # RACE = 1
# # Race = 3, smoke = 1, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#             (1.0716 * 1) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#             (1.0716 * 1) + (1.749 * 1))), col = cols[9], lwd = 3,
#             add = T)
```

```
#
#
# # Race = 0, smoke = 1, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#           (1.0716 * 1) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#           (1.0716 * 1) + (1.749 * 0))), col = cols[10], lwd = 3,
#             add = T)
#
# # Race = 0, smoke = 0, ht = 1
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 1)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#           (1.0716 * 0) + (1.749 * 1))), col = cols[11], lwd = 3,
#             add = T)
#
# # Race = 0, smoke = 0, ht = 0
# curve(exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#             (1.0716 * 0) + (1.749 * 0)) /
#         (1 + exp(0.3520 - (0.0179 * x) + (1.2877 * 0) + (0.9436 * 0) +
#             (1.0716 * 0) + (1.749 * 0))), col = cols[12], lwd = 3,
#             add = T)
#
# legend("topright", pch = rep(16,12), col = cols, legend =
#          c("Race = 2, Smoke = Y, HT = Y",
#            "Race = 2, Smoke = Y, HT = N",
#            "Race = 2, Smoke = N, HT = Y",
#            "Race = 2, Smoke = N, HT = N",
#            "Race = 3, Smoke = Y, HT = Y",
#            "Race = 3, Smoke = Y, HT = N",
#            "Race = 3, Smoke = N, HT = Y",
#            "Race = 3, Smoke = N, HT = N",
#            "Race = 1, Smoke = Y, HT = Y",
#            "Race = 1, Smoke = Y, HT = N",
#            "Race = 1, Smoke = N, HT = Y",
#            "Race = 1, Smoke = N, HT = N"))

# Check goodness-of-fit.
res2 <- mod2$deviance
dof2 <- mod2$df.residual
p2 <- pchisq(res2, dof2, lower.tail = F)
round(p2,3)
```

```
## [1] 0.097
```

```
# Our p-value is 0.097, so we cannot reject the null hypothesis and
# thus our logistic regression model fits this data well.

# Obtain confusion matrix.
# Running the training data through the model to get some predictions
pred <- predict.glm(mod2, type = "response" )

# Classify the prediction, and build confusion matrix.
```

```r
pred_class <- ifelse(pred >= 0.5, 1,0)
conf <- table(dat$low, pred_class)
rownames(conf) = c('Obs_Normal', 'Obs_Low')
colnames(conf) = c('Pred_Normal', 'Pred_Low')
conf
```

```
##             pred_class
##              Pred_Normal Pred_Low
##   Obs_Normal         123        7
##   Obs_Low             43       16
```

```r
# Calculate misclassification rate using confusion matrix.
miss <- round(((conf[2] + conf[3]) / (conf[1] + conf[2] + conf[3] + conf[4])) * 100,2)


cat("The misclassification rate is: ", miss, "%")
```

```
## The misclassification rate is:  26.46 %
```