Logistic Regression Excersie

Missy Lee
October 10, 2016

Set working directory getwd() [1] "C:/Users/mlee/Documents/GitHub/DataWranglingExercise1/Rstatistics" List files and load data list.files("dataSets") [1] "Exam.rds" "NatHealth2008MI" "NatHealth2011.rds" "states.dta" "states.rds"
NH11 <- readRDS("dataSets/NatHealth2011.rds")

```
NH11 <- readRDS("dataSets/NatHealth2011.rds")
str(NH11)
## 'data.frame':    33014 obs. of  36 variables:
##  $ fmx     : chr  "01" "01" "01" "01" ...
##  $ fpx     : chr  "03" "03" "01" "01" ...
##  $ wtia_sa : num  7521 5784 2512 3086 12530 ...
##  $ wtfa_sa : num  8814 10427 2791 3888 16609 ...
##  $ region  : num  3 3 1 3 3 1 3 3 3 3 ...
##  $ strat_p : num  223 201 3 166 125 31 190 190 217 173 ...
##  $ psu_p   : num  1 2 1 1 2 1 1 1 1 1 ...
##  $ sex     : Factor w/ 2 levels "1 Male","2 Female": 2 2 2 2 2 2 2 2 1 1 .
..
##  $ hispan_i: Factor w/ 13 levels "00 Multiple Hispanic",..: 13 13 13 13 13
13 7 13 13 13 ...
##  $ mracrpi2: Factor w/ 9 levels "01 White","02 Black/African American",..:
1 2 2 2 1 1 1 1 2 1 ...
##  $ age_p   : num  47 18 79 51 43 41 21 20 33 56 ...
##  $ r_maritl: Factor w/ 10 levels "0 Under 14 years",..: 6 8 5 7 2 2 8 8 8
2 ...
##  $ everwrk : Factor w/ 5 levels "1 Yes","2 No",..: NA NA 1 NA NA NA NA NA
1 1 ...
##  $ hypev   : Factor w/ 5 levels "1 Yes","2 No",..: 2 2 1 2 2 1 2 2 1 2 ...
##  $ aasmev  : Factor w/ 5 levels "1 Yes","2 No",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ aasmyr  : Factor w/ 5 levels "1 Yes","2 No",..: 1 NA NA NA NA NA NA NA
NA NA ...
##  $ dibev   : Factor w/ 6 levels "1 Yes","2 No",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ dibage  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ difage2 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ insln   : Factor w/ 5 levels "1 Yes","2 No",..: 2 NA NA NA NA NA NA NA
NA NA ...
##  $ dibpill : Factor w/ 5 levels "1 Yes","2 No",..: 2 NA NA NA NA NA NA NA
NA NA ...
##  $ arth1   : Factor w/ 5 levels "1 Yes","2 No",..: 1 2 1 2 2 1 2 2 1 2 ...
##  $ arthlmt : Factor w/ 5 levels "1 Yes","2 No",..: 2 NA 1 NA NA 2 NA 2 2 N
A ...
##  $ wkdayr  : num  3 0 NA 0 1 0 0 1 NA 0 ...
##  $ beddayr : num  3 0 0 0 1 0 0 0 0 0 ...
##  $ aflhca18: Factor w/ 5 levels "1 Mentioned",..: 2 NA 2 NA NA 2 2 NA 2 NA
...
##  $ aldura10: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ aldura17: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ aldura18: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ smkev   : Factor w/ 5 levels "1 Yes","2 No",..: 2 2 2 1 3 2 2 2 2 1 ...
##  $ cigsday : num  NA NA NA 5 NA NA NA NA NA NA ...
##  $ vigmin  : num  NA NA NA NA NA 60 120 30 NA 120 ...
##  $ modmin  : num  15 NA 10 NA NA 30 30 120 NA 45 ...
##  $ bmi     : num  100 21.6 32.3 100 100 ...
##  $ sleep   : num  6 8 6 8 9 8 7 6 10 8 ...
##  $ ausualpl: Factor w/ 6 levels "1 Yes","2 There is NO place",..: 1 2 1 2
1 1 1 2 1 1 ...
##  - attr(*, "labels")='data.frame':   36 obs. of  2 variables:
##   ..$ name : Factor w/ 591 levels "aaseryr1","aasmev",..: 452 453 590 589
538 567 534 541 455 520 ...
##   ..$ label: Factor w/ 590 levels " AAU.050_01.010: Doesn't need doctor/ha
ven't had problems",..: 359 472 534 533 483 480 479 497 400 481 ...
```

Regression with binary outcomes

Logistic regression

This far we have used the lm' function to fit our regression models. ##lm' is great, but limitedâ models for continuous dependent variables. For categorical dependent variables we can use the `glm()' function. For these models we will use a different dataset, drawn from the National Health Interview Survey. From the [CDC website http://www.cdc.gov/nchs/nhis.htm]: The National Health Interview Survey (NHIS) has monitored the health of the nation since 1957. NHIS data on a broad range of health topics are collected through personal household interviews. For over 50 years, the U.S. Census Bureau has been the data collection agent for the National Health Interview Survey. Survey results have been instrumental in providing data to track health status, health care access, and progress toward achieving national health objectives.

Load the National Health Interview Survey data:

NH11 <- readRDS("dataSets/NatHealth2011.rds") labs <- attributes(NH11)$labels

Logistic regression example

Let's predict the probability of being diagnosed with hypertension based on age, sex, sleep, and bmi

check structure of hypev

```
str(NH11$hypev)
##  Factor w/ 5 levels "1 Yes","2 No",..: 2 2 1 2 2 1 2 2 1 2 ...
```

```
levels(NH11$hypev)
## [1] "1 Yes"             "2 No"              "7 Refused"
## [4] "8 Not ascertained" "9 Don't know"
```

Collapse all missing values to NA

```
NH11$hypev <- factor(NH11$hypev, levels=c("2 No", "1 Yes"))
```

Run our regression model

```
hyp.out <- glm(hypev~age_p+sex+sleep+bmi,
              data=NH11, family="binomial")

coef(summary(hyp.out))

##                 Estimate   Std. Error    z value     Pr(>|z|)
## (Intercept) -4.269466028 0.0564947294 -75.572820 0.000000e+00
## age_p        0.060699303 0.0008227207  73.778743 0.000000e+00
## sex2 Female -0.144025092 0.0267976605  -5.374540 7.677854e-08
## sleep       -0.007035776 0.0016397197  -4.290841 1.779981e-05
## bmi          0.018571704 0.0009510828  19.526906 6.485172e-85
```

Logistic regression coefficients

Generalized linear models use link functions, so raw coefficients are difficult to interpret. For example, the age coefficient of .06 in the previous model tells us that for every one unit increase in age, the log odds of hypertension diagnosis increases by 0.06. Since most of us are not used to thinking in log odds this is not too helpful! One solution is to transform the coefficients to make them easier to interpret

```
hyp.out.tab <- coef(summary(hyp.out))
hyp.out.tab[, "Estimate"] <- exp(coef(hyp.out))
hyp.out.tab

##               Estimate   Std. Error    z value     Pr(>|z|)
## (Intercept) 0.01398925 0.0564947294 -75.572820 0.000000e+00
## age_p       1.06257935 0.0008227207  73.778743 0.000000e+00
## sex2 Female 0.86586602 0.0267976605  -5.374540 7.677854e-08
## sleep       0.99298892 0.0016397197  -4.290841 1.779981e-05
## bmi         1.01874523 0.0009510828  19.526906 6.485172e-85
          Estimate   Std. Error    z value     Pr(>|z|)
```

Generating predicted values

In addition to transforming the log-odds produced by glm' to odds, we ##   can use the predict()' function to make direct statements about the predictors in our model. For example, we can ask "How much more likely is a 63 year old female to have hypertension compared to a 33 year old female?".

Create a dataset with predictors set at desired levels

```
predDat <- with(NH11,
                expand.grid(age_p = c(33, 63),
                            sex = "2 Female",
                            bmi = mean(bmi, na.rm = TRUE),
                            sleep = mean(sleep, na.rm = TRUE)))
```

Predict hypertension at those levels

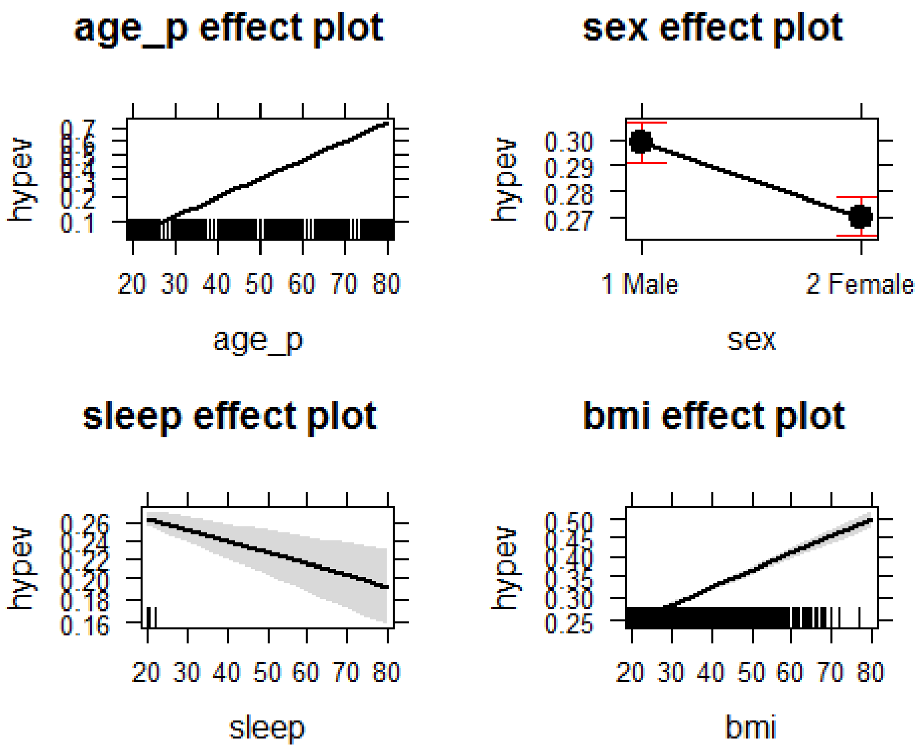
```
cbind(predDat, predict(hyp.out, type = "response",
                       se.fit = TRUE, interval="confidence",
                       newdata = predDat))
##   age_p      sex      bmi   sleep       fit      se.fit residual.scale
## 1    33 2 Female 29.89565 7.86221 0.1289227 0.002849622              1
## 2    63 2 Female 29.89565 7.86221 0.4776303 0.004816059              1
```

This tells us that a 33 year old female has a 13% probability of having been diagnosed with hypertension, while and 63 year old female has a 48% probability of having been diagnosed.

Packages for computing and graphing predicted values

Instead of doing all this ourselves, we can use the effects package to compute quantities of interest for us (cf. the Zelig package).

```
library(effects) plot(allEffects(hyp.o ut))
```



Exercise: logistic regression

Use the NH11 data set that we loaded earlier.
1. Use glm to conduct a logistic regression to predict ever worked (everwrk) using age (age_p) and marital status (r_maritl).
2. Predict the probability of working for each level of marital status.
Note that the data is not perfectly clean and ready to be modeled. You will need to clean up at least some of the variables before fitting the model.

The field r_marital contains no data in the version of the data used for this exercise. I will use the variable sex (sex) instead.

```
str(NH11$everwrk)
##  Factor w/ 5 levels "1 Yes","2 No",..: NA NA 1 NA NA NA NA NA 1 1 ...
str(NH11$sex)
##  Factor w/ 2 levels "1 Male","2 Female": 2 2 2 2 2 2 2 2 1 1 ...
```

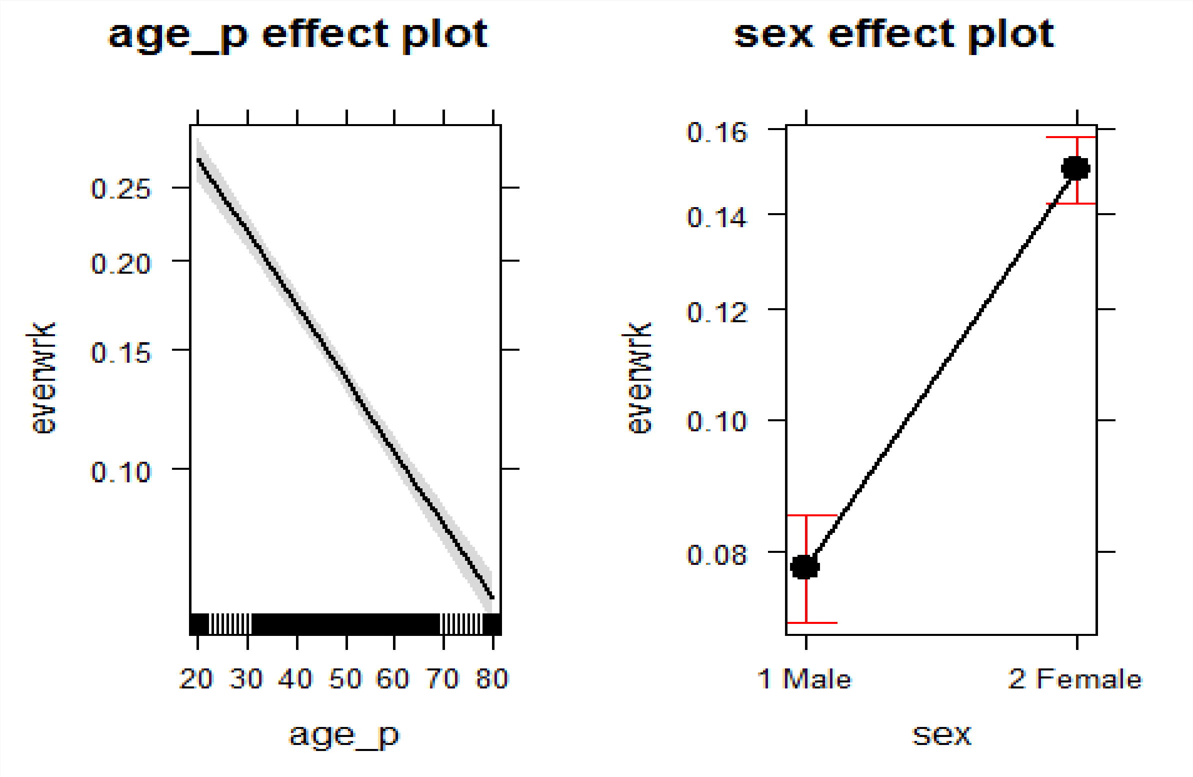
Run the regression model

```
mod.work <- glm(everwrk~age_p+sex,data=NH11, family=binomial)

coef(summary(mod.work))

##                Estimate  Std. Error   z value      Pr(>|z|)
## (Intercept) -0.87811468 0.075701824 -11.59965  4.137653e-31
## age_p       -0.02849825 0.001245026 -22.88967  5.888051e-116
## sex2 Female  0.73997897 0.057258130  12.92356  3.314369e-38
```

Transform the coefficients

```
mod.work.tab <- coef(summary(mod.work))
mod.work.tab[, "Estimate"] <- exp(coef(mod.work))
mod.work.tab

##              Estimate  Std. Error   z value      Pr(>|z|)
## (Intercept) 0.4155656 0.075701824 -11.59965  4.137653e-31
## age_p       0.9719040 0.001245026 -22.88967 5.888051e-116
## sex2 Female 2.0958914 0.0572581mod.work30  12.92356   3.314369e-38
```

Plot the results



According to this data, you are less likely to work as you age, but women are more likely to have worked than men.