LogisticRegression

Nirmal Patel

October 14, 2018

Regression with binary outcomes

Logistic regression

This far we have used the lm' function to fit our regression models.lm' is great, but limitedâ in particular it only fits models for continuous dependent variables. For categorical dependent variables wecan 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("NatHealth2011.rds")
labs <- attributes(NH11)$labels
```

[CDC website] http://www.cdc.gov/nchs/nhis.htm

Logistic regression example

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

```
table(NH11$hypev)
```

```
## 
##              1 Yes              2 No         7 Refused 8 Not ascertained 
##              10672             22296                20                 0 
##       9 Don't know 
##                 26
```

```r
str(NH11$hypev) # check stucture of hypev
```

```
##  Factor w/ 5 levels "1 Yes","2 No",..: 2 2 1 2 2 1 2 2 1 2 ...
```

```r
levels(NH11$hypev) # check levels of hypev
```

```
## [1] "1 Yes"             "2 No"              "7 Refused"        
## [4] "8 Not ascertained" "9 Don't know"
```

collapse all missing values to NA

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

run our regression model

```r
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

```r
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
```

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 likelyis 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

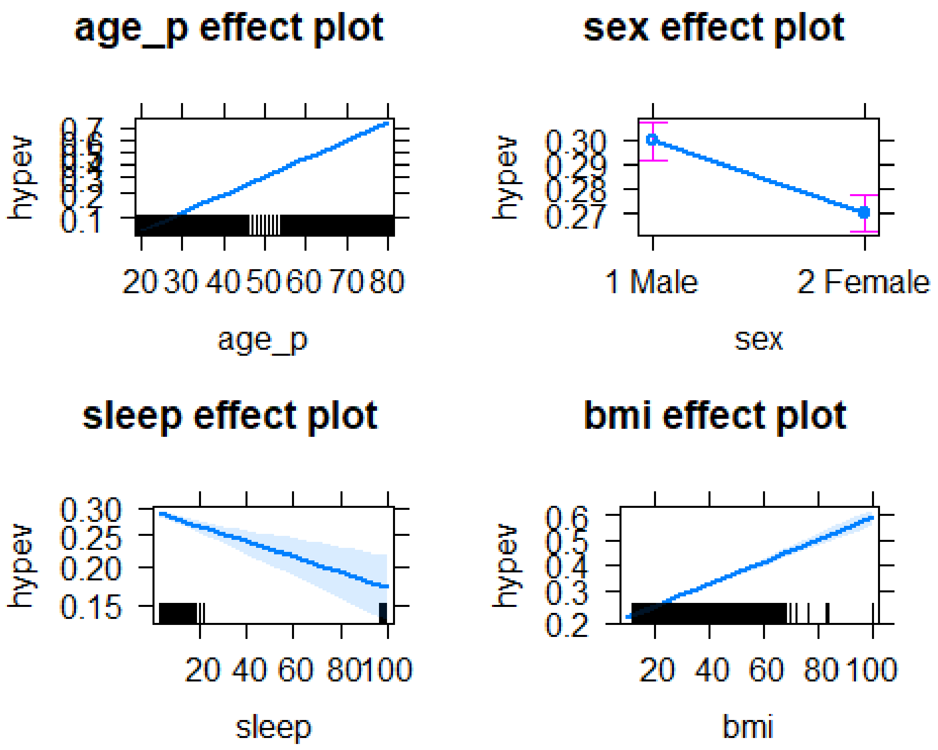
```
cbind(predDat, predict(hyp.out, type = "response",
                       se.fit = TRUE, interval="confidence",
                       newdata = predDat))

##   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).

```
plot(allEffects(hyp.out))
```



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).

```
table(NH11$everwrk)

## 
##               1 Yes                2 No          7 Refused 8 Not ascertained 
##               12153                1887                 17                 0 
##        9 Don't know 
##                   8

str(NH11$everwrk) # check stucture of everwrk

##  Factor w/ 5 levels "1 Yes","2 No",..: NA NA 1 NA NA NA NA NA 1 1 ...

str(NH11$age_p) # check stucture of age

##  num [1:33014] 47 18 79 51 43 41 21 20 33 56 ...

levels(NH11$everwrk) # check levels of everwrk

## [1] "1 Yes"             "2 No"              "7 Refused"        
## [4] "8 Not ascertained" "9 Don't know"
```

```
levels(NH11$r_maritl) # check levels of r_maritl

##  [1] "0 Under 14 years"
##  [2] "1 Married - spouse in household"
##  [3] "2 Married - spouse not in household"
##  [4] "3 Married - spouse in household unknown"
##  [5] "4 Widowed"
##  [6] "5 Divorced"
##  [7] "6 Separated"
##  [8] "7 Never married"
##  [9] "8 Living with partner"
## [10] "9 Unknown marital status"
```

collapse all missing values to NA

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

run our regression model

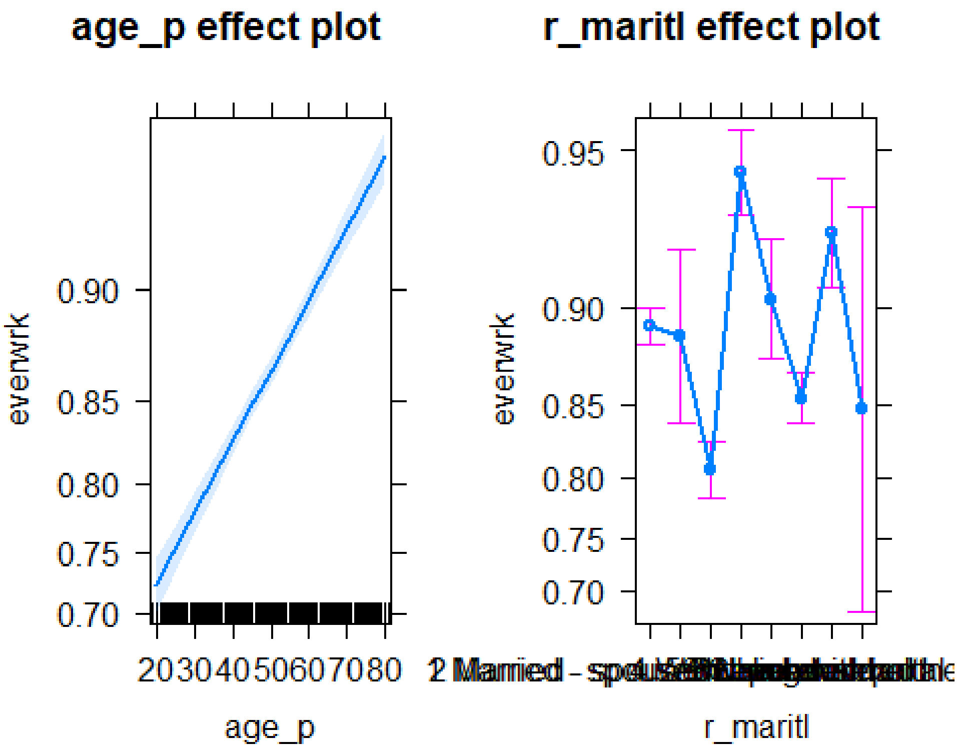
```
everwrks <- glm(everwrk~age_p+r_maritl,
               data=NH11, family="binomial")
coef(summary(everwrks))

##                                                Estimate  Std. Error
## (Intercept)                                  0.44024757 0.093537691
## age_p                                        0.02981220 0.001645433
## r_maritl2 Married - spouse not in household -0.04967549 0.217309587
## r_maritl4 Widowed                           -0.68361771 0.084335382
## r_maritl5 Divorced                           0.73011485 0.111680788
## r_maritl6 Separated                          0.12809081 0.151366140
## r_maritl7 Never married                     -0.34361068 0.069222260
## r_maritl8 Living with partner                0.44358296 0.137769623
## r_maritl9 Unknown marital status            -0.39547953 0.492966577
##                                                z value     Pr(>|z|)
## (Intercept)                                  4.7066328 2.518419e-06
## age_p                                       18.1181481 2.291800e-73
## r_maritl2 Married - spouse not in household -0.2285932 8.191851e-01
## r_maritl4 Widowed                           -8.1059419 5.233844e-16
## r_maritl5 Divorced                           6.5375152 6.254929e-11
## r_maritl6 Separated                          0.8462316 3.974236e-01
## r_maritl7 Never married                     -4.9638756 6.910023e-07
## r_maritl8 Living with partner                3.2197443 1.283050e-03
## r_maritl9 Unknown marital status            -0.8022441 4.224118e-01
```

2. Predict the probability of working for each level of marital status.

```
levels(NH11$r_maritl) # check levels of r_maritl

##  [1] "0 Under 14 years"
##  [2] "1 Married - spouse in household"
##  [3] "2 Married - spouse not in household"
##  [4] "3 Married - spouse in household unknown"
##  [5] "4 Widowed"
##  [6] "5 Divorced"
##  [7] "6 Separated"
##  [8] "7 Never married"
##  [9] "8 Living with partner"
## [10] "9 Unknown marital status"
```

gives plots of work and age and work and marital status

```
plot(allEffects(everwrks))
```


age_p effect plot
everwrk
age_p
r_maritl effect plot
everwrk
2 Married - spouse
r_maritl