logistic\_regression.R

stevehaunguyen

Sun Jun 10 07:27:42 2018

## Regression with binary outcomes  
##  
  
## Logistic regression  
##  
  
## This far we have used the `lm' function to fit our regression models.  
## `lm' is great, but limited in particular it only fits 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]:  
  
## 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:  
setwd("~/Springboard Projects/Chapter 7 - Machine Learning/logistic\_regression/logistic\_regression")  
NH11 <- readRDS("dataSets/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  
  
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 ...

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  
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

## 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))

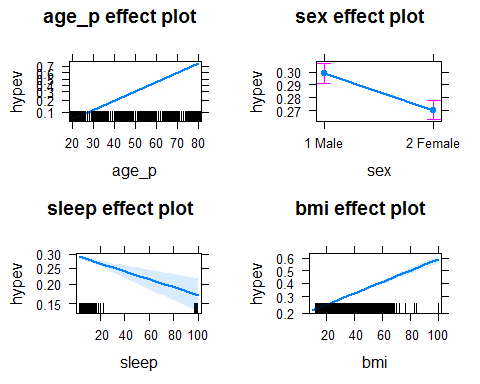
## age\_p sex bmi sleep fit se.fit residual.scale  
## 1 33 2 Female 29.89565 7.86221 0.1289227 0.002849622 1  
## 2 63 2 Female 29.89565 7.86221 0.4776303 0.004816059 1

## 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)

## Loading required package: carData

## lattice theme set by effectsTheme()  
## See ?effectsTheme for details.

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).  
  
str(NH11$everwrk) #Check structure

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

levels(NH11$everwrk) #5 levels

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

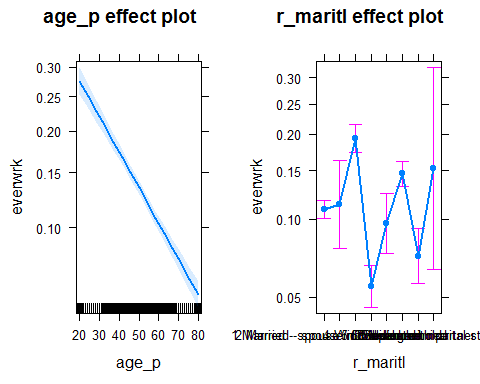
nh11.wrk.age.mar <- subset(NH11, select = c("everwrk", "age\_p", "r\_maritl"))  
NH11 <- transform(NH11,  
 everwrk = factor(everwrk,  
 levels = c("1 Yes", "2 No")),  
 r\_maritl = droplevels(r\_maritl))  
  
mod.wk.age.mar <- glm(everwrk ~ age\_p + r\_maritl, data = NH11,  
 family = "binomial")  
  
mod.wk.age.mar.tab <- coef(summary(mod.wk.age.mar))  
mod.wk.age.mar.tab[, "Estimate"] <- exp(coef(mod.wk.age.mar))  
mod.wk.age.mar.tab #For easier interpretation, instead of one unit increase, increase/decrease log odds of

## Estimate Std. Error  
## (Intercept) 0.6438770 0.093537691  
## age\_p 0.9706278 0.001645433  
## r\_maritl2 Married - spouse not in household 1.0509300 0.217309587  
## r\_maritl4 Widowed 1.9810316 0.084335382  
## r\_maritl5 Divorced 0.4818536 0.111680788  
## r\_maritl6 Separated 0.8797735 0.151366140  
## r\_maritl7 Never married 1.4100296 0.069222260  
## r\_maritl8 Living with partner 0.6417330 0.137769623  
## r\_maritl9 Unknown marital status 1.4850962 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

#ever working by coefficient (in the mod.wk.age.mar),   
 #it is one unit increase, increase/decrease odds of ever working by coefficient  
 #(in the mod.wk.age.mar.tab)  
  
summary(mod.wk.age.mar)

##   
## Call:  
## glm(formula = everwrk ~ age\_p + r\_maritl, family = "binomial",   
## data = NH11)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0436 -0.5650 -0.4391 -0.3370 2.7308   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -0.440248 0.093538 -4.707  
## age\_p -0.029812 0.001645 -18.118  
## r\_maritl2 Married - spouse not in household 0.049675 0.217310 0.229  
## r\_maritl4 Widowed 0.683618 0.084335 8.106  
## r\_maritl5 Divorced -0.730115 0.111681 -6.538  
## r\_maritl6 Separated -0.128091 0.151366 -0.846  
## r\_maritl7 Never married 0.343611 0.069222 4.964  
## r\_maritl8 Living with partner -0.443583 0.137770 -3.220  
## r\_maritl9 Unknown marital status 0.395480 0.492967 0.802  
## Pr(>|z|)   
## (Intercept) 2.52e-06 \*\*\*  
## age\_p < 2e-16 \*\*\*  
## r\_maritl2 Married - spouse not in household 0.81919   
## r\_maritl4 Widowed 5.23e-16 \*\*\*  
## r\_maritl5 Divorced 6.25e-11 \*\*\*  
## r\_maritl6 Separated 0.39742   
## r\_maritl7 Never married 6.91e-07 \*\*\*  
## r\_maritl8 Living with partner 0.00128 \*\*   
## r\_maritl9 Unknown marital status 0.42241   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 11082 on 14039 degrees of freedom  
## Residual deviance: 10309 on 14031 degrees of freedom  
## (18974 observations deleted due to missingness)  
## AIC: 10327  
##   
## Number of Fisher Scoring iterations: 5

library(effects)  
plot(allEffects(mod.wk.age.mar))



## 2. Predict the probability of working for each level of marital  
## status.  
  
library(effects)  
data.frame(Effect("r\_maritl", mod.wk.age.mar))

## r\_maritl fit se lower  
## 1 1 Married - spouse in household 0.10822000 0.04413754 0.10014980  
## 2 2 Married - spouse not in household 0.11310823 0.21326041 0.07746061  
## 3 4 Widowed 0.19381087 0.06806325 0.17381358  
## 4 5 Divorced 0.05524394 0.10272953 0.04562877  
## 5 6 Separated 0.09646417 0.14579706 0.07426824  
## 6 7 Never married 0.14611000 0.05978759 0.13208775  
## 7 8 Living with partner 0.07224958 0.13285112 0.05662466  
## 8 9 Unknown marital status 0.15270076 0.49100994 0.06440837  
## upper  
## 1 0.11685606  
## 2 0.16227532  
## 3 0.21550873  
## 4 0.06674358  
## 5 0.12440219  
## 6 0.16134411  
## 7 0.09176661  
## 8 0.32055728