## 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:

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 thepredict()’ 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)

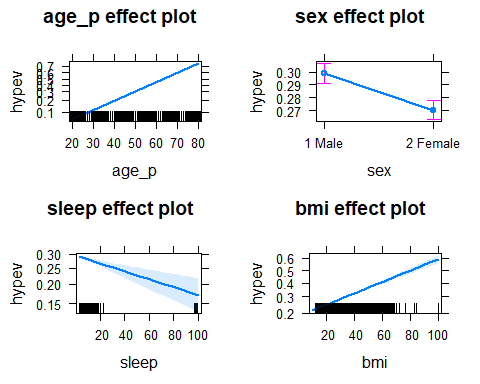
## Warning: package 'effects' was built under R version 3.4.4

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4

## 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).

# First, I noticed that (everwrk) has many NA values. I will use the ‘mice’ library to fill in any missing values.

library(mice)

## Warning: package 'mice' was built under R version 3.4.4

## Loading required package: lattice

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

set.seed(123)  
simple = NH11[c("age\_p","r\_maritl","everwrk")]  
imputed = complete(mice(simple))

##   
## iter imp variable  
## 1 1 everwrk  
## 1 2 everwrk  
## 1 3 everwrk  
## 1 4 everwrk  
## 1 5 everwrk  
## 2 1 everwrk  
## 2 2 everwrk  
## 2 3 everwrk  
## 2 4 everwrk  
## 2 5 everwrk  
## 3 1 everwrk  
## 3 2 everwrk  
## 3 3 everwrk  
## 3 4 everwrk  
## 3 5 everwrk  
## 4 1 everwrk  
## 4 2 everwrk  
## 4 3 everwrk  
## 4 4 everwrk  
## 4 5 everwrk  
## 5 1 everwrk  
## 5 2 everwrk  
## 5 3 everwrk  
## 5 4 everwrk  
## 5 5 everwrk

## Warning: Number of logged events: 25

NH11$everwrk = imputed$everwrk

# Now I can build my model with the new values added back to the original dataset. Training will consist of 70% of the data, and the other 30% will be for testing.

index = sample(1:nrow(NH11), 0.7\*nrow(NH11))  
training = NH11[index, ]  
testing = NH11[-index, ]

# Model

model = glm(everwrk~age\_p+r\_maritl, data=training, family="binomial")  
summary(model)

##   
## Call:  
## glm(formula = everwrk ~ age\_p + r\_maritl, family = "binomial",   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0024 -0.6210 -0.4862 -0.3520 2.7014   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -0.51369 0.06898 -7.447  
## age\_p -0.02795 0.00138 -20.263  
## r\_maritl2 Married - spouse not in household -0.03096 0.15824 -0.196  
## r\_maritl4 Widowed 0.61823 0.07901 7.824  
## r\_maritl5 Divorced -0.81659 0.08232 -9.919  
## r\_maritl6 Separated 0.03888 0.10580 0.367  
## r\_maritl7 Never married 0.37926 0.04696 8.075  
## r\_maritl8 Living with partner -0.55522 0.09187 -6.044  
## r\_maritl9 Unknown marital status 0.62431 0.34610 1.804  
## Pr(>|z|)   
## (Intercept) 9.58e-14 \*\*\*  
## age\_p < 2e-16 \*\*\*  
## r\_maritl2 Married - spouse not in household 0.8449   
## r\_maritl4 Widowed 5.10e-15 \*\*\*  
## r\_maritl5 Divorced < 2e-16 \*\*\*  
## r\_maritl6 Separated 0.7133   
## r\_maritl7 Never married 6.74e-16 \*\*\*  
## r\_maritl8 Living with partner 1.51e-09 \*\*\*  
## r\_maritl9 Unknown marital status 0.0713 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 19842 on 23108 degrees of freedom  
## Residual deviance: 18673 on 23100 degrees of freedom  
## AIC: 18691  
##   
## Number of Fisher Scoring iterations: 5

### From the model you can see that age and the majority of marital statuses had a very high impact on ever worked.

# Baseline Model

table(NH11$everwrk)

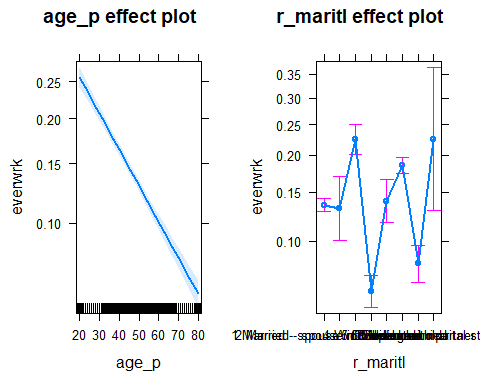
##   
## 1 Yes 2 No 7 Refused 8 Not ascertained   
## 27998 4955 44 0   
## 9 Don't know   
## 17

### From the table output, out of 33,014 responses 84.8% (27,998) of participants have worked, 15% (4,955) have not ever worked, and < 1% have either refused (44) or don’t know (17).

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

## status.

pred1 = predict(model, newdata=testing, type="response")  
final = table(testing$r\_maritl, pred1)  
plot(allEffects(model))



## 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.