GLM Logistic Demo

Kim Johnson and Kyle Pitzer

February 19, 2020

Introduction

This demo will use the BRFSS2017_10percent_v2.csv dataset to ask and answer three questions:
1. Is BMI a risk factor for diabetes?
- What are the null and alternative hypotheses?
2. Is low income a risk factor for diabetes?
- What are the null and alternative hypotheses?
3. Are BMI and low income still risk factors after controlling for each other?

Load packages and read in data

```
#install.packages("DescTools")
#install.packages("lmtest")
library(tidyverse)
library(car)
library(DescTools)
library(lmtest) #for LR test

#loading csv from github
BRFSS <- read_csv("https://raw.githubusercontent.com/kijohnson/ADA_Spring_2019/master/BRFSS2017_10percen

#show the first part of the data
print(head(BRFSS))
```

```
## # A tibble: 6 x 13
##        X1  rowID X_AGE80 age_cat state employed income seatbelt diabetes   bmi
##     <dbl>  <dbl>   <dbl> <chr>   <chr> <chr>    <chr>  <chr>    <chr>    <dbl>
## 1 119484 119484      42 Age 40~ IN    Employe~ $20,0~ Always   No        39.6
## 2 167462 167462      60 Age 60~ LA    Retired  $50,0~ Always   No        24.4
## 3 257793 257793      28 Age 25~ NE    Employe~ $35,0~ Never    No        25.1
## 4 408706 408706      55 Age 55~ VA    Employe~ $75,0~ Always   No        27.1
## 5  90760  90760      42 Age 40~ GA    Employe~ < $10~ Refused  No        32.1
## 6 404286 404286      55 Age 55~ VT    Employe~ $75,0~ Always   No        28.6
## # ... with 3 more variables: wtkg <dbl>, ht_meters <dbl>, sex <chr>
```

Classify diabetes as a binary variable for logistic regression analyses.

Since we want to do a logistic regression, we need to make sure our outcome is binary.

```
#check type of variable
class(BRFSS$diabetes)
```

```
## [1] "character"
```

```
#look at number of observations per level
table(BRFSS$diabetes)
```

```
## 
##                         Don't know/Not Sure 
##                                          56 
##                                          No 
##                                       37719 
##     No, pre-diabetes or borderline diabetes 
##                                         844 
##                                     Refused 
##                                          16 
##                                         Yes 
##                                        6041 
## Yes, but female told only during pregnancy 
##                                         326
```

Here, we have clear no's, no's that are borderline, and clear yes's, and yes's where the female was only told during pregnancy. Let's combine the no's and yes's and exclude "Don't know/Not sure" and "Refused".

```
#make a binary diabetes variable categorizing diabetes into yes and no and excluding individuals with o
BRFSS$diabetes_binary[
  BRFSS$diabetes=="No"| BRFSS$diabetes=="No, pre-diabetes or borderline diabetes"]<-0 #Assign 0 to thos

BRFSS$diabetes_binary[
  BRFSS$diabetes=="Yes"|BRFSS$diabetes=="Yes, but female told only during pregnancy"]<-1 #Assign 1 to t

#check to make sure re-classification worked
table(BRFSS$diabetes_binary, BRFSS$diabetes)
```

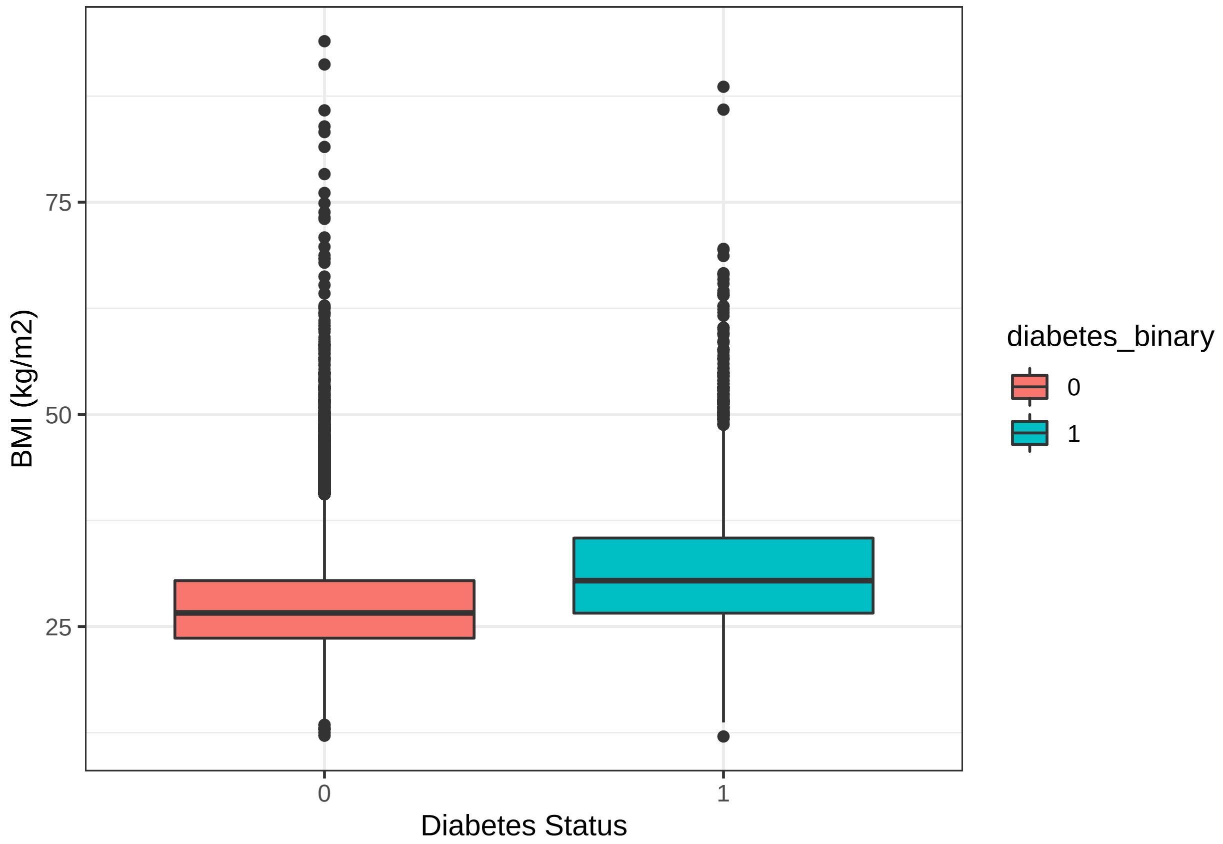
```
## 
##     Don't know/Not Sure    No No, pre-diabetes or borderline diabetes Refused
##   0                   0 37719                                     844       0
##   1                   0     0                                       0       0
## 
##       Yes Yes, but female told only during pregnancy
##   0     0                                          0
##   1  6041                                        326
```

Make a box plot to visualize whether there is a difference in the BMI distributions by diabetes status

Let's examine a boxplot to see if there are any potential differences in diabetes by BMI.

```
#BRFSS$bmi<-as.numeric(as.character(BRFSS$bmi)) #you may need this code
BRFSS$diabetes_binary<- as.factor(BRFSS$diabetes_binary)

#Drop NA's from diabetes binary and bmi variables and then plot the boxplots
BRFSS %>%
  drop_na(c(diabetes_binary, bmi)) %>%
ggplot(aes(x = diabetes_binary, y = bmi)) +
  geom_boxplot(aes(fill = diabetes_binary)) +
  labs(x = "Diabetes Status", y = "BMI (kg/m2)") +
  theme_bw()
```



What can you conclude from this boxplot?

Recode variables and create complete cases data set

In the models we plan to run, we will use the diabetes_binary variable as the outcome and a continuous and categorical version of BMI as well as a collapsed categorical variable for income as the predictors. Although you typically want to do data management on the front end to take care of NA's (e.g. recode Don't Know/Refused as NA), our recode here will force anything not recoded to NA for each variable. We will then create a complete cases data set for analysis so we have the same number of observations in each model. This step is important since we do some model comparison.

First, we will check our variables.

```
#checking summaries for each variable to get an idea of NA values
summary(BRFSS$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   12.05   23.86   27.16   28.15   31.19   93.97    3561
```

```
summary(as.factor(BRFSS$income))
```

```
##                    < $10,000 $10,000 to less than $15,000 
##                         1826                         1906 
## $15,000 to less than $20,000 $20,000 to less than $25,000 
##                         2759                         3349 
## $25,000 to less than $35,000 $35,000 to less than $50,000 
##                         4032                         5366
## $50,000 to less than $75,000              $75,000 or more
##                         5918                        12392
##          Don't know/Not sure                      Refused
##                         3353                         3780
##                         NA's
##                          321
```

```r
summary(BRFSS$diabetes_binary)
```

```
##     0     1  NA's
## 38563  6367    72
```

Let's create a categorical BMI variable according to underweight (<18.5 kg/m2) normal (18.5 to <25 kg/m2), overweight (25 to <30 kg/m2), and obese (30 kg/m2 and above) categories.

```r
#recoding BMI to 4 categories
BRFSS$bmi_cat[
  (BRFSS$bmi>0 & BRFSS$bmi<18.5)]<-0
BRFSS$bmi_cat[
  (BRFSS$bmi>=18.5 & BRFSS$bmi<25)]<-1
BRFSS$bmi_cat[
  (BRFSS$bmi>=25 & BRFSS$bmi<30)]<-2
BRFSS$bmi_cat[
  (BRFSS$bmi>=30)]<-3

#checking to make sure recode worked
summary(BRFSS$bmi_cat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.962   3.000   3.000    3561
```

```r
by(BRFSS$bmi, BRFSS$bmi_cat, summary)
```

```
## BRFSS$bmi_cat: 0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.05   16.76   17.68   17.21   18.07   18.48
## ------------------------------------------------------------
## BRFSS$bmi_cat: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.52   21.29   22.78   22.52   23.91   24.98
## ------------------------------------------------------------
## BRFSS$bmi_cat: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.00   25.96   27.32   27.31   28.48   29.99
## ------------------------------------------------------------
## BRFSS$bmi_cat: 3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.00   31.61   33.89   35.42   37.49   93.97
```

Let's also create a variable for income with three levels: less than 25K, 25 to <75K, >75K, and exclude others from analysis.

```r
#checking class and values of income variable
class(BRFSS$income)
```

```
## [1] "character"
```

```r
table(BRFSS$income)
```

```
##
##                    < $10,000 $10,000 to less than $15,000
##                         1826                         1906
## $15,000 to less than $20,000 $20,000 to less than $25,000
##                         2759                         3349
## $25,000 to less than $35,000 $35,000 to less than $50,000
##                         4032                         5366
## $50,000 to less than $75,000              $75,000 or more
##                         5918                        12392
##          Don't know/Not sure                      Refused
##                         3353                         3780
```

```
#recoding income to three categories
BRFSS$income_3L[
  BRFSS$income=="< $10,000"|
  BRFSS$income=="$10,000 to less than $15,000"|
  BRFSS$income=="$15,000 to less than $20,000"|
  BRFSS$income=="$20,000 to less than $25,000"]<-2

BRFSS$income_3L[
  BRFSS$income=="$25,000 to less than $35,000"|
  BRFSS$income=="$35,000 to less than $50,000"]<-1

BRFSS$income_3L[
  BRFSS$income=="$50,000 to less than $75,000"|
  BRFSS$income=="$75,000 or more"]<-0

#checking to make sure recode worked
table(BRFSS$income_3L, BRFSS$income)
```

```
##
##     < $10,000 $10,000 to less than $15,000 $15,000 to less than $20,000
##   0         0                            0                            0
##   1         0                            0                            0
##   2      1826                         1906                         2759
##
##     $20,000 to less than $25,000 $25,000 to less than $35,000
##   0                            0                            0
##   1                            0                         4032
##   2                         3349                            0
##
##     $35,000 to less than $50,000 $50,000 to less than $75,000 $75,000 or more
##   0                            0                         5918           12392
##   1                         5366                            0               0
##   2                            0                            0               0
##
##     Don't know/Not sure Refused
##   0                   0       0
##   1                   0       0
##   2                   0       0
```

Finally, let's create a data set with only valid data for each variable used in our models.

```
#defining variables to include in the complete data set
myvars <- c("rowID", "diabetes_binary", "bmi", "bmi_cat", "income_3L")
```

```r
#subsetting by those variables
BRFSS_cc<-BRFSS[myvars]

#omitting NA's in the data set
BRFSS_cc<-na.omit(BRFSS_cc)

#checking to make sure there are no NA's
summary(BRFSS_cc)
```

```
##      rowID        diabetes_binary      bmi           bmi_cat
##  Min.   :     4   0:30518         Min.   :12.14   Min.   :0.000
##  1st Qu.:112212   1: 5072         1st Qu.:24.03   1st Qu.:1.000
##  Median :225502                   Median :27.29   Median :2.000
##  Mean   :225390                   Mean   :28.27   Mean   :1.978
##  3rd Qu.:337038                   3rd Qu.:31.32   3rd Qu.:3.000
##  Max.   :450008                   Max.   :93.97   Max.   :3.000
##    income_3L
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.7648
##  3rd Qu.:2.0000
##  Max.   :2.0000
```

Test assumptions of linearity and influence

Linearity

To do the Box Tidwell test, we need to create a term for the predictor*log(predictor) and then run a logistic regression with that term. Remember, a significant coefficient means the assumption is violated.

```r
#linearity
bmi.times.logbmi <- BRFSS_cc$bmi * log(BRFSS_cc$bmi)#create term to test linearity

boxTidwellBMI <- glm(diabetes_binary ~ bmi + bmi.times.logbmi, data=BRFSS_cc, family="binomial") #Box T

summary(boxTidwellBMI)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ bmi + bmi.times.logbmi, family = "binomial",
##     data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0414  -0.5897  -0.4693  -0.3530   2.9600
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -10.05281    0.44840  -22.42   <2e-16 ***
## bmi                0.86351    0.06036   14.31   <2e-16 ***
## bmi.times.logbmi  -0.17138    0.01332  -12.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

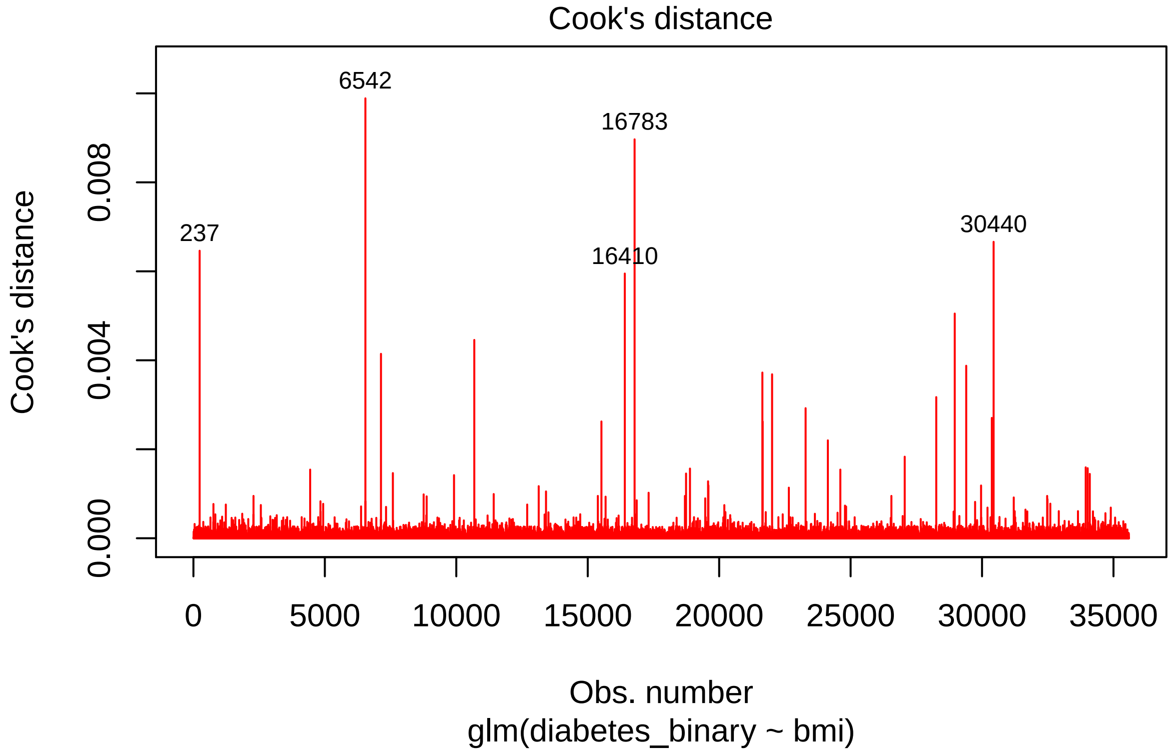
```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 27383  on 35587  degrees of freedom
## AIC: 27389
##
## Number of Fisher Scoring iterations: 5
```

What would your conclusion be about the linearity assumption?

Influence

Here, we check for influential data using Cook's Distance.

```
#logistic model with bmi as a predictor
bmiLogitCD <- glm(diabetes_binary ~ bmi, data=BRFSS_cc, family="binomial")
#influence plot - Cook's D plot-identifies observation number in parent dataset
  plot(bmiLogitCD, which=4, id.n=5, col="red")
```



Note that testing for multicollinearity are not necessary because we only have one predictor.

Because linearity assumption was violated with BMI, let's use the categorical variable according to underweight (<18.5 kg/m2) normal (18.5 to <25 kg/m2), overweight (25 to <30 kg/m2), and obese (30 kg/m2 and above) for running in models below as well.

Run logistic models for both BMI and BMI_cat

BMI

```
#bmi logistic model
bmiLogit <- glm(diabetes_binary ~ bmi, data=BRFSS_cc, family="binomial")
```

```r
  summary(bmiLogit)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ bmi, family = "binomial", data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8023  -0.5612  -0.4758  -0.4023   2.5412
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.399285   0.070264  -62.61   <2e-16 ***
## bmi          0.088387   0.002245   39.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 27561  on 35588  degrees of freedom
## AIC: 27565
##
## Number of Fisher Scoring iterations: 4
```

```r
#calculate and print ORs and 95% CIs
  ORbmi<-exp(cbind(OR = coef(bmiLogit), confint(bmiLogit))) #calculate ORs and 95% CIs
```

```
## Waiting for profiling to be done...
```

```r
  ORbmi #print ORs and 95% CIs
```

```
##                     OR      2.5 %     97.5 %
## (Intercept) 0.01228612 0.01070043 0.01409393
## bmi         1.09241082 1.08762297 1.09723919
```

```r
#another way! Use Dr. Harris' odds.n.ends package!

#install.packages("odds.n.ends")
library(odds.n.ends)
odds.n.ends(bmiLogit)
```

```
## Waiting for profiling to be done...
```

```
## $`Logistic regression model significance`
## Chi-squared        d.f.           p
##      1586.7         1.0         0.0
##
## $`Contingency tables (model fit): percent predicted`
##                  Percent observed
## Percent predicted           1           0         Sum
##               1   0.003118854 0.004917112 0.008035965
##               0   0.139393088 0.852570947 0.991964035
##               Sum 0.142511942 0.857488058 1.000000000
##
## $`Contingency tables (model fit): frequency predicted`
##                  Number observed
## Number predicted     1     0   Sum
##               1    111   175   286
##               0   4961 30343 35304
##               Sum 5072 30518 35590
##
## $`Predictor odds ratios and 95% CI`
##                     OR      2.5 %     97.5 %
## (Intercept) 0.01228612 0.01070043 0.01409393
## bmi         1.09241082 1.08762297 1.09723919
##
## $`Model sensitivity`
## [1] 0.02188486
##
## $`Model specificity`
## [1] 0.9942657
```

How do we interpret the results?

BMI_cat

```
#bmi_cat logistic model
bmi_catLogit <- glm(diabetes_binary ~as.factor(bmi_cat), data=BRFSS_cc, family="binomial")
  summary(bmi_catLogit)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ as.factor(bmi_cat), family = "binomial",
##     data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7477  -0.5110  -0.5110  -0.3706   2.4164
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.8641     0.1877 -15.259  < 2e-16 ***
## as.factor(bmi_cat)1   0.2202     0.1916   1.149     0.25
## as.factor(bmi_cat)2   0.8940     0.1896   4.715 2.42e-06 ***
## as.factor(bmi_cat)3   1.7323     0.1890   9.167  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 27627  on 35586  degrees of freedom
## AIC: 27635
##
## Number of Fisher Scoring iterations: 5
```

```
#calculate and print ORs and 95% CIs
  ORbmi_cat<-exp(cbind(OR = coef(bmi_catLogit), confint(bmi_catLogit))) #calculate ORs and 95% CIs
```

```
## Waiting for profiling to be done...
```

```
  ORbmi_cat #print ORs and 95% CIs
```

```
##                              OR      2.5 %     97.5 %
## (Intercept)          0.05703422 0.03863799 0.08085627
## as.factor(bmi_cat)1 1.24634746 0.87169220 1.85234466
## as.factor(bmi_cat)2 2.44481926 1.71735274 3.62092082
## as.factor(bmi_cat)3 5.65391706 3.97703938 8.36456090
```

How do we interpret the results?

Checking model fits for BMI and BMI_cat

Let's check the log likelihood and sensitivity and specificity for the BMI model.

Log Likelihood for BMI and BMI_cat

```
#Log Likelihood for BMI
logLik(bmiLogit)
```

```
## 'log Lik.' -13780.66 (df=2)
```

We will use this to compare to the model with two predictors below.

Sensitivity and Specificity

```
#check percent correctly predicted (example of how to do this)
xt <- addmargins(table(round(predict(bmiLogit, type="response")), bmiLogit$model$diabetes_binary))
  xt #Note the Gold standard (reporting by participant) is the column variable and the model prediction
```

```
##
##           0     1   Sum
##   0   30343  4961 35304
##   1     175   111   286
##   Sum 30518  5072 35590
```

```
#Can you calculate sensitivity and specificity of the model for predicting diabetes?

#Sensitivity
111/5072
```

```
## [1] 0.02188486
```

```
#Specificity
30343/30518
```

```
## [1] 0.9942657
```

```
#Total predicted correctly
30454/35590
```

```
## [1] 0.8556898
```

Run logistic model for income_3L

First, make a bivariate table and calculate proportions at each income_3L level that have diabetes (gives insight into what is expected from the model)

```
xt<-table(BRFSS_cc$income_3L, BRFSS_cc$diabetes_binary)
  xt
```

```
##
##         0     1
##   0 15798  1737
##   1  7466  1426
##   2  7254  1909
```

```
  prop.table(xt, 1)
```

```
##
##              0          1
##   0 0.90094098 0.09905902
##   1 0.83963113 0.16036887
##   2 0.79166212 0.20833788
```

Now, let's change the reference group and run the model.

```
#set reference at low income
BRFSS_cc$income_3L <- relevel(as.factor(BRFSS_cc$income_3L), ref=3)

#income logistic model
incLogit <- glm(diabetes_binary ~ as.factor(income_3L), data=BRFSS_cc, family="binomial")
  summary(incLogit)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ as.factor(income_3L), family = "binomial",
##     data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6835  -0.5913  -0.4568  -0.4568   2.1504
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.33497    0.02572 -51.897   <2e-16 ***
## as.factor(income_3L)0 -0.87275    0.03606 -24.200   <2e-16 ***
## as.factor(income_3L)1 -0.32051    0.03869  -8.284   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 28536  on 35587  degrees of freedom
## AIC: 28542
##
## Number of Fisher Scoring iterations: 4
```

```
#calculate and print ORs and 95% CIs
ORincome <- exp(cbind(OR = coef(incLogit), confint(incLogit))) #calculate ORs and 95% CIs
```

```
## Waiting for profiling to be done...
```

```
  ORincome #print ORs and 95% CIs
```

```
##                              OR     2.5 %    97.5 %
## (Intercept)           0.2631652 0.2501634 0.2767056
## as.factor(income_3L)0 0.4178009 0.3892721 0.4483886
## as.factor(income_3L)1 0.7257769 0.6727095 0.7828841
```

What can we conclude about the relationship between income and diabetes?

##Multivariate model with diabetes as the dependent variable and income and bmi/bmi_cat as the independent variables

BMI continuous

```
#income and bmi logistic model
bmiIncLogit <- glm(diabetes_binary ~ as.factor(income_3L) + bmi, data=BRFSS_cc, family="binomial")
  summary(bmiIncLogit)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ as.factor(income_3L) + bmi, family = "binomial",
##     data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6729  -0.5700  -0.4582  -0.3686   2.5393
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.868810   0.074741 -51.763  < 2e-16 ***
## as.factor(income_3L)0 -0.777459   0.037127 -20.941  < 2e-16 ***
## as.factor(income_3L)1 -0.275784   0.039928  -6.907 4.95e-12 ***
## bmi                    0.084273   0.002258  37.319  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 27102  on 35586  degrees of freedom
## AIC: 27110
##
## Number of Fisher Scoring iterations: 5
```

```
#calculate and print ORs and 95% CIs
ORmodel<-exp(cbind(OR = coef(bmiIncLogit), confint(bmiIncLogit))) #calculate ORs and 95% CIs
```

```
## Waiting for profiling to be done...
```

```
  ORmodel #print ORs and 95% CIs
```

```
##                               OR      2.5 %     97.5 %
## (Intercept)           0.02088321 0.01802909 0.02416672
## as.factor(income_3L)0 0.45957207 0.42730852 0.49425382
## as.factor(income_3L)1 0.75897712 0.70178601 0.82069565
## bmi                   1.08792545 1.08313061 1.09276139
```

BMI categorical

```
#income and bmi cat logistic model
bmi_catIncLogit <- glm(diabetes_binary ~ as.factor(income_3L) + as.factor(bmi_cat), data=BRFSS_cc, famil
  summary(bmi_catIncLogit)
```

```
##
## Call:
## glm(formula = diabetes_binary ~ as.factor(income_3L) + as.factor(bmi_cat),
##     family = "binomial", data = BRFSS_cc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8898  -0.6224  -0.4306  -0.3088   2.6107
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.55337    0.18884 -13.521  < 2e-16 ***
## as.factor(income_3L)0 -0.82097    0.03692 -22.234  < 2e-16 ***
## as.factor(income_3L)1 -0.30827    0.03970  -7.765 8.15e-15 ***
## as.factor(bmi_cat)1    0.35522    0.19229   1.847   0.0647 .
## as.factor(bmi_cat)2    1.04293    0.19033   5.480 4.26e-08 ***
## as.factor(bmi_cat)3    1.83123    0.18965   9.656  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 29148  on 35589  degrees of freedom
## Residual deviance: 27112  on 35584  degrees of freedom
## AIC: 27124
##
## Number of Fisher Scoring iterations: 5
```

```
#calculate and print ORs and 95% CIs
ORmodel<-exp(cbind(OR = coef(bmi_catIncLogit), confint(bmi_catIncLogit))) #calculate ORs and 95% CIs
```

```
## Waiting for profiling to be done...
```

```
  ORmodel #print ORs and 95% CIs
```

```
##                               OR      2.5 %    97.5 %
## (Intercept)           0.07781908 0.05261434 0.1105979
## as.factor(income_3L)0 0.44000304 0.40927089 0.4730140
## as.factor(income_3L)1 0.73471664 0.67965335 0.7941008
## as.factor(bmi_cat)1   1.42649392 0.99622978 2.1226193
## as.factor(bmi_cat)2   2.83752589 1.99013718 4.2078220
## as.factor(bmi_cat)3   6.24154834 4.38403296 9.2447742
```

How can we answer question number 3 based on the model results?

Check model fit for full models

Let's check the log likelihood and sensitivity and specificity for the full BMI model.

Log Likelihood

```r
#Log Likelihood for full model
logLik(bmiIncLogit)
```

```
## 'log Lik.' -13550.93 (df=4)
```

```r
#compare models with just bmi to that with bmi and income using LR test
lrtest(bmiLogit, bmiIncLogit)
```

```
## Likelihood ratio test
##
## Model 1: diabetes_binary ~ bmi
## Model 2: diabetes_binary ~ as.factor(income_3L) + bmi
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   2 -13781
## 2   4 -13551  2 459.46  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How does the log likelihood compare to the BMI only model, and what can we conclude with the LR test?

Sensitivity and Specificity

```r
#check percent correctly predicted (example of how to do this) for bmi continuous
xt <- addmargins(table(round(predict(bmiIncLogit, type="response")), bmiIncLogit$model$diabetes_binary)
  xt #Note the Gold standard (reporting by participant) is the column variable and the model prediction
```

```
##
##           0     1   Sum
##   0   30312  4929 35241
##   1     206   143   349
##   Sum 30518  5072 35590
```

```r
     #Can you calculate sensitivity and specificity of the model for predicting diabetes?

#Sensitivity
143/5072
```

```
## [1] 0.02819401
```

```r
#Specificity
30312/30518
```

```
## [1] 0.9932499
```

```r
#Total predicted correctly
30455/35590
```

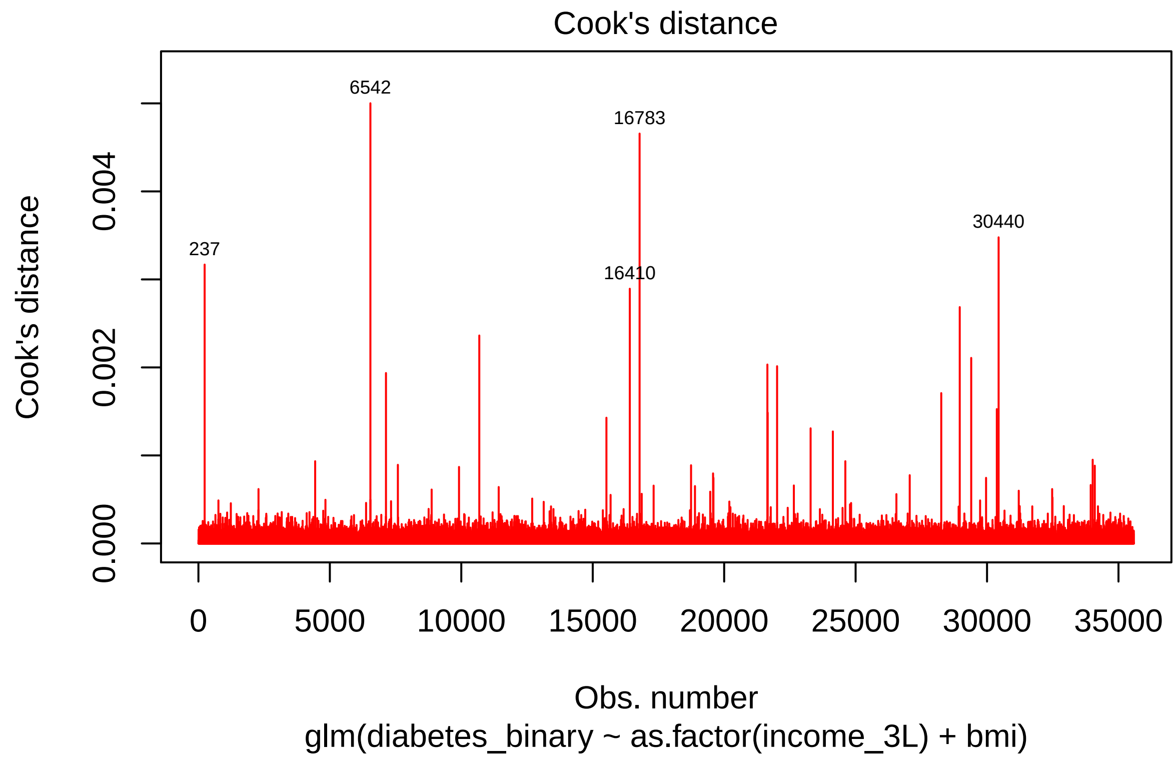
```
## [1] 0.8557179
```

Look at assumptions of influence and multicollinearity

Finally, we will check for influential data in the full model and multicollinearity between our predictors.

Influence

```r
#Cook's D plot
plot(bmiIncLogit, which=4, id.n=5, col="red", cex.id=0.60)
```



```
#identify observations with a Cook's D greater than 0.0015
y<-as.data.frame(cooks.distance(bmiIncLogit))
colnames(y)[1]<-"CD"
y$obs_no<-rownames(y)
z<-y[which(y$CD>0.0015),]
z$obs_no
```

```
##  [1] "237"   "6542"  "7135"  "10685" "16410" "16783" "21644" "22013" "28258"
## [10] "28961" "29398" "30372" "30440"
```

Multicollinearity

```
#Variance Inflation Factors
vif(bmiIncLogit)
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## as.factor(income_3L) 1.002988  2        1.000746
## bmi                  1.002988  1        1.001493
```

Exclude influential observations and compare Betas

Let's exclude the values shown in the Cook's D plot, and see how the models compare.

```
#car library needed for compareCoefs (notice the Camelcase!)
#dropping obs with CD>0.0015
bmiIncLogit.modex <- update(bmiIncLogit,subset=c(-237, -6542, -7135, -10685, -16410, -16783, -21644, -2
                                                 -28961, -29398, -30372, -30440))
#compare coefficients between models with and without influential observations, #caveat model number of
compareCoefs(bmiIncLogit, bmiIncLogit.modex)
```

```
## Calls:
```

```
## 1: glm(formula = diabetes_binary ~ as.factor(income_3L) + bmi, family =
##   "binomial", data = BRFSS_cc)
## 2: glm(formula = diabetes_binary ~ as.factor(income_3L) + bmi, family =
##   "binomial", data = BRFSS_cc, subset = c(-237, -6542, -7135, -10685, -16410,
##   -16783, -21644, -22013, -28258, -28961, -29398, -30372, -30440))
##
##                        Model 1 Model 2
## (Intercept)           -3.8688 -3.9566
## SE                     0.0747  0.0754
##
## as.factor(income_3L)0 -0.7775 -0.7764
## SE                     0.0371  0.0372
##
## as.factor(income_3L)1 -0.2758 -0.2761
## SE                     0.0399  0.0400
##
## bmi                   0.08427 0.08724
## SE                    0.00226 0.00228
##
```

Did removing influential data affect the coefficients?

Interpretation and conclusions (Discussion)

For fun:

1. The BMI linearity assumption was violated, if you remove influential observations is it still violated using the Box Tidwell method?
2. Calculate the sensitivity and specificity of the model for predicting reported diabetes