Logistic Regression Project: Heart Disease

William Lin, Ishani Parekh, Kaiyu Yokoi

2023-05-18

### Problem statement and description of data

…

### Loading dataset

rm(list=ls()) # Clears objects from workspace  
#Load the data set   
#heart\_disease <-read.csv("/Users/ishani/Desktop/449project/heart\_2020\_cleaned.csv", stringsAsFactors=TRUE)   
heart\_disease <- read.csv("/Users/linw/Desktop/449project/heart\_2020\_cleaned.csv", stringsAsFactors = TRUE)  
  
head(heart\_disease)

## HeartDisease BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth  
## 1 No 16.60 Yes No No 3 30  
## 2 No 20.34 No No Yes 0 0  
## 3 No 26.58 Yes No No 20 30  
## 4 No 24.21 No No No 0 0  
## 5 No 23.71 No No No 28 0  
## 6 Yes 28.87 Yes No No 6 0  
## DiffWalking Sex AgeCategory Race Diabetic PhysicalActivity GenHealth  
## 1 No Female 55-59 White Yes Yes Very good  
## 2 No Female 80 or older White No Yes Very good  
## 3 No Male 65-69 White Yes Yes Fair  
## 4 No Female 75-79 White No No Good  
## 5 Yes Female 40-44 White No Yes Very good  
## 6 Yes Female 75-79 Black No No Fair  
## SleepTime Asthma KidneyDisease SkinCancer  
## 1 5 Yes No Yes  
## 2 7 No No No  
## 3 8 Yes No No  
## 4 6 No No Yes  
## 5 8 No No No  
## 6 12 No No No

#structure and summary of the data   
str(heart\_disease)

## 'data.frame': 319795 obs. of 18 variables:  
## $ HeartDisease : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...  
## $ BMI : num 16.6 20.3 26.6 24.2 23.7 ...  
## $ Smoking : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 2 1 2 1 1 ...  
## $ AlcoholDrinking : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Stroke : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ PhysicalHealth : num 3 0 20 0 28 6 15 5 0 0 ...  
## $ MentalHealth : num 30 0 30 0 0 0 0 0 0 0 ...  
## $ DiffWalking : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 2 1 2 1 2 ...  
## $ Sex : Factor w/ 2 levels "Female","Male": 1 1 2 1 1 1 1 1 1 2 ...  
## $ AgeCategory : Factor w/ 13 levels "18-24","25-29",..: 8 13 10 12 5 12 11 13 13 10 ...  
## $ Race : Factor w/ 6 levels "American Indian/Alaskan Native",..: 6 6 6 6 6 3 6 6 6 6 ...  
## $ Diabetic : Factor w/ 4 levels "No","No, borderline diabetes",..: 3 1 3 1 1 1 1 3 2 1 ...  
## $ PhysicalActivity: Factor w/ 2 levels "No","Yes": 2 2 2 1 2 1 2 1 1 2 ...  
## $ GenHealth : Factor w/ 5 levels "Excellent","Fair",..: 5 5 2 3 5 2 2 3 2 3 ...  
## $ SleepTime : num 5 7 8 6 8 12 4 9 5 10 ...  
## $ Asthma : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 2 1 1 ...  
## $ KidneyDisease : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...  
## $ SkinCancer : Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 2 1 1 1 ...

summary(heart\_disease)

## HeartDisease BMI Smoking AlcoholDrinking Stroke   
## No :292422 Min. :12.02 No :187887 No :298018 No :307726   
## Yes: 27373 1st Qu.:24.03 Yes:131908 Yes: 21777 Yes: 12069   
## Median :27.34   
## Mean :28.33   
## 3rd Qu.:31.42   
## Max. :94.85   
##   
## PhysicalHealth MentalHealth DiffWalking Sex   
## Min. : 0.000 Min. : 0.000 No :275385 Female:167805   
## 1st Qu.: 0.000 1st Qu.: 0.000 Yes: 44410 Male :151990   
## Median : 0.000 Median : 0.000   
## Mean : 3.372 Mean : 3.898   
## 3rd Qu.: 2.000 3rd Qu.: 3.000   
## Max. :30.000 Max. :30.000   
##   
## AgeCategory Race   
## 65-69 : 34151 American Indian/Alaskan Native: 5202   
## 60-64 : 33686 Asian : 8068   
## 70-74 : 31065 Black : 22939   
## 55-59 : 29757 Hispanic : 27446   
## 50-54 : 25382 Other : 10928   
## 80 or older: 24153 White :245212   
## (Other) :141601   
## Diabetic PhysicalActivity GenHealth   
## No :269653 No : 71838 Excellent: 66842   
## No, borderline diabetes: 6781 Yes:247957 Fair : 34677   
## Yes : 40802 Good : 93129   
## Yes (during pregnancy) : 2559 Poor : 11289   
## Very good:113858   
##   
##   
## SleepTime Asthma KidneyDisease SkinCancer   
## Min. : 1.000 No :276923 No :308016 No :289976   
## 1st Qu.: 6.000 Yes: 42872 Yes: 11779 Yes: 29819   
## Median : 7.000   
## Mean : 7.097   
## 3rd Qu.: 8.000   
## Max. :24.000   
##

attach(heart\_disease)

### Data cleaning

### Setting reference variables, ordering levels of ordinal variables  
heart\_disease$Race = relevel(heart\_disease$Race, ref = "White")  
heart\_disease$GenHealth = factor(heart\_disease$GenHealth, levels = c("Excellent", "Very good", "Good", "Fair", "Poor"))  
  
### Creating new quantitative variables for ordinal variables using scores  
# AgeScore = c(21,27,32,37,42,47,52,57,62,67,72,77,90)  
# DiabeticScore = c(0, 1, 2, 3) # (No, No borderline, Yes, Yes pregnant)   
# GenHeatlhScore = c(0, 1, 2, 3, 4) # (Excellent, Very good, Good, Fair, Poor)  
heart\_disease$Age=rep(0,319795)  
 heart\_disease$Age[heart\_disease$AgeCategory=="18-24"]=21  
 heart\_disease$Age[heart\_disease$AgeCategory=="25-29"]=27  
 heart\_disease$Age[heart\_disease$AgeCategory=="30-34"]=32  
 heart\_disease$Age[heart\_disease$AgeCategory=="35-39"]=37  
 heart\_disease$Age[heart\_disease$AgeCategory=="40-44"]=42  
 heart\_disease$Age[heart\_disease$AgeCategory=="45-49"]=47  
 heart\_disease$Age[heart\_disease$AgeCategory=="50-54"]=52  
 heart\_disease$Age[heart\_disease$AgeCategory=="55-59"]=57  
 heart\_disease$Age[heart\_disease$AgeCategory=="60-64"]=62  
 heart\_disease$Age[heart\_disease$AgeCategory=="65-69"]=67  
 heart\_disease$Age[heart\_disease$AgeCategory=="70-74"]=72  
 heart\_disease$Age[heart\_disease$AgeCategory=="75-79"]=77  
 heart\_disease$Age[heart\_disease$AgeCategory=="80 or older"]=87  
   
heart\_disease$DiabeticScore=rep(0,319795)  
 heart\_disease$DiabeticScore[heart\_disease$Diabetic=="No"]=0  
 heart\_disease$DiabeticScore[heart\_disease$Diabetic=="No, borderline diabetes"]=1  
 heart\_disease$DiabeticScore[heart\_disease$Diabetic=="Yes"]=2  
 heart\_disease$DiabeticScore[heart\_disease$Diabetic=="Yes (during pregnancy)"]=3  
   
heart\_disease$GenHealthScore=rep(0,319795)  
 heart\_disease$GenHealthScore[heart\_disease$GenHealth=="Excellent"]=0  
 heart\_disease$GenHealthScore[heart\_disease$GenHealth=="Very good"]=1  
 heart\_disease$GenHealthScore[heart\_disease$GenHealth=="Good"]=2  
 heart\_disease$GenHealthScore[heart\_disease$GenHealth=="Fair"]=3  
 heart\_disease$GenHealthScore[heart\_disease$GenHealth=="Poor"]=4  
  
originalvars = data.frame(heart\_disease$AgeCategory, heart\_disease$Diabetic,  
 heart\_disease$GenHealth)   
   
# Remove ordinal variables; replaced with scores  
heart\_disease = subset(heart\_disease, select = -c(AgeCategory, Diabetic, GenHealth))

### Checking proportions in response variable

summary(heart\_disease$HeartDisease)

## No Yes   
## 292422 27373

count.yes = nrow(heart\_disease[heart\_disease$HeartDisease == 'Yes', ])  
count.no = nrow(heart\_disease[heart\_disease$HeartDisease == 'No', ])  
# Proportion of yes  
(prop.yes = count.yes / (count.yes + count.no))

## [1] 0.08559546

The data is unbalanced; only 8.5% of the response variable are Yes’s. Training data is likely to be biased towards the No responses. For this project we will downsample the No class such that the proportion of No responses is 2/3.

# Downsampling majority class  
set.seed(123)  
yes.df <- heart\_disease[which(heart\_disease$HeartDisease == "Yes"), ]  
no.df <- heart\_disease[which(heart\_disease$HeartDisease == "No"), ]  
no.index <- as.numeric(rownames(no.df))  
no.sample <- sample(no.index, 2\*count.yes)  
no.sample.index <- data.frame(no.sample)  
no.sample.index <- no.sample.index[order(no.sample), ]  
no.downsample <- heart\_disease[no.sample.index, ]   
heart\_disease2 <- rbind(yes.df, no.downsample)  
  
summary(heart\_disease2)

## HeartDisease BMI Smoking AlcoholDrinking Stroke   
## No :54746 Min. :12.02 No :44446 No :77108 No :76283   
## Yes:27373 1st Qu.:24.28 Yes:37673 Yes: 5011 Yes: 5836   
## Median :27.44   
## Mean :28.59   
## 3rd Qu.:31.83   
## Max. :93.86   
## PhysicalHealth MentalHealth DiffWalking Sex   
## Min. : 0.000 Min. : 0.000 No :65740 Female:40550   
## 1st Qu.: 0.000 1st Qu.: 0.000 Yes:16379 Male :41569   
## Median : 0.000 Median : 0.000   
## Mean : 4.566 Mean : 4.103   
## 3rd Qu.: 3.000 3rd Qu.: 3.000   
## Max. :30.000 Max. :30.000   
## Race PhysicalActivity SleepTime   
## White :64180 No :21471 Min. : 1.000   
## American Indian/Alaskan Native: 1415 Yes:60648 1st Qu.: 6.000   
## Asian : 1712 Median : 7.000   
## Black : 5749 Mean : 7.104   
## Hispanic : 6328 3rd Qu.: 8.000   
## Other : 2735 Max. :24.000   
## Asthma KidneyDisease SkinCancer Age DiabeticScore   
## No :70085 No :77129 No :72508 Min. :21.0 Min. :0.0000   
## Yes:12034 Yes: 4990 Yes: 9611 1st Qu.:47.0 1st Qu.:0.0000   
## Median :62.0 Median :0.0000   
## Mean :58.8 Mean :0.4069   
## 3rd Qu.:72.0 3rd Qu.:0.0000   
## Max. :87.0 Max. :3.0000   
## GenHealthScore  
## Min. :0.00   
## 1st Qu.:1.00   
## Median :2.00   
## Mean :1.63   
## 3rd Qu.:2.00   
## Max. :4.00

### Training and test sets

library(ggplot2)  
set.seed(1)  
  
train = sample(1:nrow(heart\_disease2), 0.8\*nrow(heart\_disease2)) # 80/20 split  
test = (-train)  
y.test = heart\_disease2$HeartDisease[test]  
  
# Dataframes  
train.df = data.frame(heart\_disease2[train,])  
test.df = data.frame(heart\_disease2[test,])

### Null model

glm.null = glm(HeartDisease~1, data=heart\_disease2, subset=train,  
 family=binomial)

### Fit a logistic regression model with all predictors

# AgeCategory, Diabetic, GenHealth replaced with scores  
glm.fit=glm(HeartDisease ~ ., data=heart\_disease2, subset = train,  
 family=binomial)  
summary(glm.fit)

##   
## Call:  
## glm(formula = HeartDisease ~ ., family = binomial, data = heart\_disease2,   
## subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.8980 -0.7138 -0.3600 0.7560 3.1481   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.0388315 0.0936400 -64.490 < 2e-16 \*\*\*  
## BMI 0.0118250 0.0016776 7.049 1.81e-12 \*\*\*  
## SmokingYes 0.3860132 0.0206389 18.703 < 2e-16 \*\*\*  
## AlcoholDrinkingYes -0.2400483 0.0459991 -5.219 1.80e-07 \*\*\*  
## StrokeYes 1.1683305 0.0396899 29.437 < 2e-16 \*\*\*  
## PhysicalHealth 0.0037299 0.0012813 2.911 0.003601 \*\*   
## MentalHealth 0.0061786 0.0013234 4.669 3.03e-06 \*\*\*  
## DiffWalkingYes 0.2551703 0.0277258 9.203 < 2e-16 \*\*\*  
## SexMale 0.7500774 0.0210167 35.690 < 2e-16 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.0611385 0.0758883 0.806 0.420451   
## RaceAsian -0.4207052 0.0892276 -4.715 2.42e-06 \*\*\*  
## RaceBlack -0.2422708 0.0414717 -5.842 5.16e-09 \*\*\*  
## RaceHispanic -0.1159969 0.0432923 -2.679 0.007376 \*\*   
## RaceOther 0.0218306 0.0577385 0.378 0.705360   
## PhysicalActivityYes 0.0458036 0.0238888 1.917 0.055191 .   
## SleepTime -0.0242179 0.0065171 -3.716 0.000202 \*\*\*  
## AsthmaYes 0.2817197 0.0287569 9.797 < 2e-16 \*\*\*  
## KidneyDiseaseYes 0.6174908 0.0413905 14.919 < 2e-16 \*\*\*  
## SkinCancerYes 0.1433762 0.0297403 4.821 1.43e-06 \*\*\*  
## Age 0.0526551 0.0007741 68.024 < 2e-16 \*\*\*  
## DiabeticScore 0.2499736 0.0123329 20.269 < 2e-16 \*\*\*  
## GenHealthScore 0.5126802 0.0120933 42.394 < 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: 83691 on 65694 degrees of freedom  
## Residual deviance: 60464 on 65673 degrees of freedom  
## AIC: 60508  
##   
## Number of Fisher Scoring iterations: 5

We see that PhysicalActivity is not statistically significant at the 0.05 level with an associated p-value of 0.0551. Thus, we will fit a model with PhysicalActivity dropped.

### Select the best subset of variables. Perform a diagnostic on the best model. Perform all possible inferences you can think about.

# Model 2: p-value selection  
# Remove non-significant variable: PhysicalActivity  
glm.fit2 = update(glm.fit, ~ . -PhysicalActivity)   
summary(glm.fit2)

##   
## Call:  
## glm(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +   
## Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +   
## Race + SleepTime + Asthma + KidneyDisease + SkinCancer +   
## Age + DiabeticScore + GenHealthScore, family = binomial,   
## data = heart\_disease2, subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9024 -0.7139 -0.3600 0.7555 3.1484   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.9830230 0.0889318 -67.277 < 2e-16 \*\*\*  
## BMI 0.0115774 0.0016725 6.922 4.44e-12 \*\*\*  
## SmokingYes 0.3842467 0.0206177 18.637 < 2e-16 \*\*\*  
## AlcoholDrinkingYes -0.2408655 0.0460045 -5.236 1.64e-07 \*\*\*  
## StrokeYes 1.1675848 0.0396865 29.420 < 2e-16 \*\*\*  
## PhysicalHealth 0.0035364 0.0012773 2.769 0.005629 \*\*   
## MentalHealth 0.0061228 0.0013230 4.628 3.70e-06 \*\*\*  
## DiffWalkingYes 0.2472590 0.0274155 9.019 < 2e-16 \*\*\*  
## SexMale 0.7521179 0.0209918 35.829 < 2e-16 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.0593601 0.0758835 0.782 0.434066   
## RaceAsian -0.4203880 0.0891970 -4.713 2.44e-06 \*\*\*  
## RaceBlack -0.2434697 0.0414627 -5.872 4.31e-09 \*\*\*  
## RaceHispanic -0.1197466 0.0432327 -2.770 0.005609 \*\*   
## RaceOther 0.0224874 0.0577338 0.390 0.696905   
## SleepTime -0.0243946 0.0065157 -3.744 0.000181 \*\*\*  
## AsthmaYes 0.2819941 0.0287570 9.806 < 2e-16 \*\*\*  
## KidneyDiseaseYes 0.6160914 0.0413814 14.888 < 2e-16 \*\*\*  
## SkinCancerYes 0.1454886 0.0297218 4.895 9.83e-07 \*\*\*  
## Age 0.0525515 0.0007719 68.084 < 2e-16 \*\*\*  
## DiabeticScore 0.2492847 0.0123272 20.222 < 2e-16 \*\*\*  
## GenHealthScore 0.5101072 0.0120163 42.451 < 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: 83691 on 65694 degrees of freedom  
## Residual deviance: 60468 on 65674 degrees of freedom  
## AIC: 60510  
##   
## Number of Fisher Scoring iterations: 5

## Checking model goodness-of-fit  
# drop1(glm.fit, test="LRT")  
anova(glm.fit2, glm.fit, test="LRT") # likelihood-ratio test comparing models

## Analysis of Deviance Table  
##   
## Model 1: HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke + PhysicalHealth +   
## MentalHealth + DiffWalking + Sex + Race + SleepTime + Asthma +   
## KidneyDisease + SkinCancer + Age + DiabeticScore + GenHealthScore  
## Model 2: HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke + PhysicalHealth +   
## MentalHealth + DiffWalking + Sex + Race + PhysicalActivity +   
## SleepTime + Asthma + KidneyDisease + SkinCancer + Age + DiabeticScore +   
## GenHealthScore  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 65674 60468   
## 2 65673 60464 1 3.683 0.05497 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We fail to reject H0, so it is safe to assume that the models are the same. Now we use backward subset selection to find the best selection of variables and compare with the previous model.

# Backward subset selection  
library(MASS)  
stepAIC(glm.fit) # stepwise backward selection using AIC

## Start: AIC=60508.28  
## HeartDisease ~ BMI + Smoking + AlcoholDrinking + Stroke + PhysicalHealth +   
## MentalHealth + DiffWalking + Sex + Race + PhysicalActivity +   
## SleepTime + Asthma + KidneyDisease + SkinCancer + Age + DiabeticScore +   
## GenHealthScore  
##   
## Df Deviance AIC  
## <none> 60464 60508  
## - PhysicalActivity 1 60468 60510  
## - PhysicalHealth 1 60473 60515  
## - SleepTime 1 60478 60520  
## - MentalHealth 1 60486 60528  
## - SkinCancer 1 60487 60529  
## - AlcoholDrinking 1 60492 60534  
## - BMI 1 60514 60556  
## - Race 5 60526 60560  
## - DiffWalking 1 60549 60591  
## - Asthma 1 60560 60602  
## - KidneyDisease 1 60694 60736  
## - Smoking 1 60814 60856  
## - DiabeticScore 1 60873 60915  
## - Stroke 1 61412 61454  
## - Sex 1 61775 61817  
## - GenHealthScore 1 62340 62382  
## - Age 1 66041 66083

##   
## Call: glm(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +   
## Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +   
## Race + PhysicalActivity + SleepTime + Asthma + KidneyDisease +   
## SkinCancer + Age + DiabeticScore + GenHealthScore, family = binomial,   
## data = heart\_disease2, subset = train)  
##   
## Coefficients:  
## (Intercept) BMI   
## -6.038831 0.011825   
## SmokingYes AlcoholDrinkingYes   
## 0.386013 -0.240048   
## StrokeYes PhysicalHealth   
## 1.168330 0.003730   
## MentalHealth DiffWalkingYes   
## 0.006179 0.255170   
## SexMale RaceAmerican Indian/Alaskan Native   
## 0.750077 0.061139   
## RaceAsian RaceBlack   
## -0.420705 -0.242271   
## RaceHispanic RaceOther   
## -0.115997 0.021831   
## PhysicalActivityYes SleepTime   
## 0.045804 -0.024218   
## AsthmaYes KidneyDiseaseYes   
## 0.281720 0.617491   
## SkinCancerYes Age   
## 0.143376 0.052655   
## DiabeticScore GenHealthScore   
## 0.249974 0.512680   
##   
## Degrees of Freedom: 65694 Total (i.e. Null); 65673 Residual  
## Null Deviance: 83690   
## Residual Deviance: 60460 AIC: 60510

sapply(list(glm.fit, glm.fit2), AIC) # compare AIC of models

## [1] 60508.28 60509.96

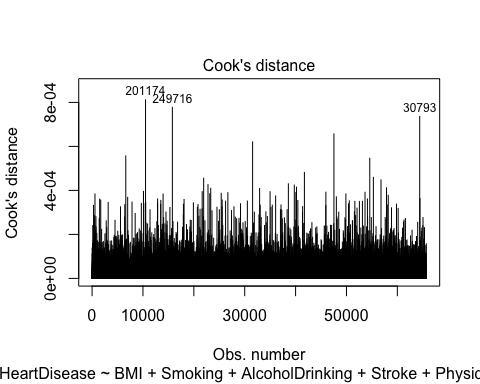
After applying backward subset selection, we find that no variables were dropped from the full model containing 17 predictors. Then we compare model 1 and 2 and select the model with the lower AIC. The AICs associated with the all-predictors model and model 2 are 145546 and 145547. However, since the difference in AIC 2, we will choose the simpler model: the model with PhysicalActivity dropped.

# Model diagnostic checking  
## Checking correlation between predictors  
cor(heart\_disease[,c("BMI", "PhysicalHealth", "MentalHealth", "SleepTime", "Age","DiabeticScore", "GenHealthScore")])

## BMI PhysicalHealth MentalHealth SleepTime Age  
## BMI 1.00000000 0.10978754 0.06413057 -0.051822254 -0.00987085  
## PhysicalHealth 0.10978754 1.00000000 0.28798667 -0.061386632 0.10992440  
## MentalHealth 0.06413057 0.28798667 1.00000000 -0.119716788 -0.15543072  
## SleepTime -0.05182225 -0.06138663 -0.11971679 1.000000000 0.10684451  
## Age -0.00987085 0.10992440 -0.15543072 0.106844508 1.00000000  
## DiabeticScore 0.20247228 0.15136118 0.03294478 0.000449238 0.19037529  
## GenHealthScore 0.23071978 0.48269718 0.24162528 -0.063071012 0.18860567  
## DiabeticScore GenHealthScore  
## BMI 0.202472284 0.23071978  
## PhysicalHealth 0.151361181 0.48269718  
## MentalHealth 0.032944777 0.24162528  
## SleepTime 0.000449238 -0.06307101  
## Age 0.190375286 0.18860567  
## DiabeticScore 1.000000000 0.26801834  
## GenHealthScore 0.268018341 1.00000000

There is no apparent multicollinearity in the predictors; each correlation coefficient is less than 0.75.

## Check for influential values  
plot(glm.fit2, which = 4, id.n = 3)



library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

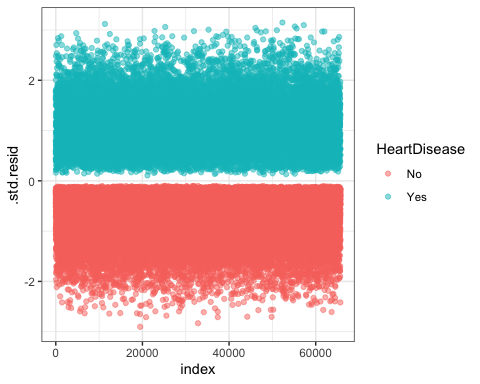
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(broom)  
# Extract model results  
model.data <- augment(glm.fit2) %>%   
 mutate(index = 1:n())   
# Top 3 influential values  
model.data %>% top\_n(3, .cooksd)

## # A tibble: 3 × 25  
## .rownames HeartDisease BMI Smoking AlcoholDrinking Stroke PhysicalHealth  
## <chr> <fct> <dbl> <fct> <fct> <fct> <dbl>  
## 1 201174 Yes 83.3 No No Yes 0  
## 2 249716 No 93.9 Yes Yes No 30  
## 3 30793 Yes 21.3 Yes Yes Yes 29  
## # ℹ 18 more variables: MentalHealth <dbl>, DiffWalking <fct>, Sex <fct>,  
## # Race <fct>, SleepTime <dbl>, Asthma <fct>, KidneyDisease <fct>,  
## # SkinCancer <fct>, Age <dbl>, DiabeticScore <dbl>, GenHealthScore <dbl>,  
## # .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,  
## # .std.resid <dbl>, index <int>

ggplot(model.data, aes(index, .std.resid)) +   
 geom\_point(aes(color = HeartDisease), alpha = .5) +  
 theme\_bw()

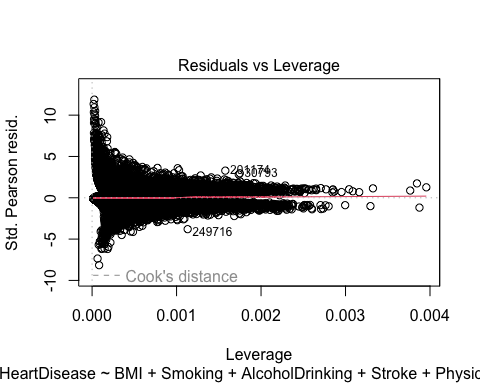
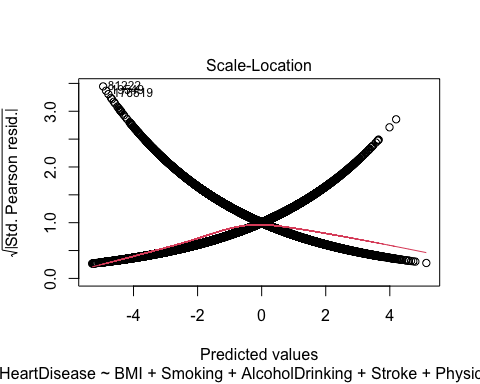
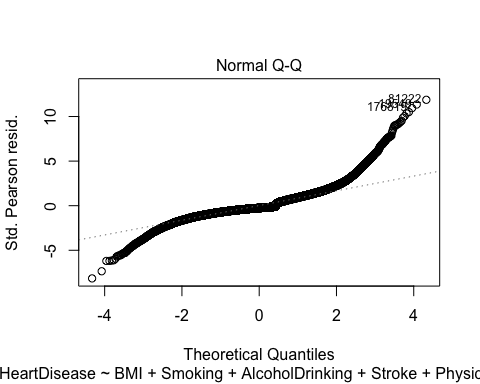
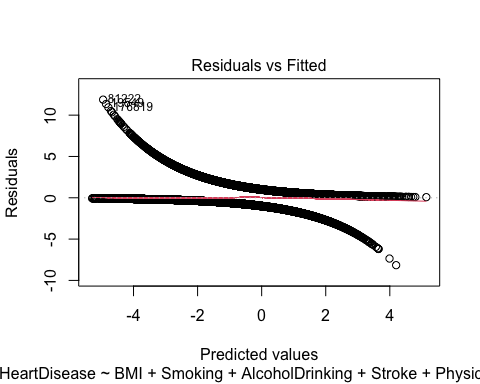


# Filter potential influential data points with abs(.std.res) > 3  
model.data %>%   
 filter(abs(.std.resid) > 3)

## # A tibble: 8 × 25  
## .rownames HeartDisease BMI Smoking AlcoholDrinking Stroke PhysicalHealth  
## <chr> <fct> <dbl> <fct> <fct> <fct> <dbl>  
## 1 19549 Yes 18.7 No No No 0  
## 2 25932 Yes 20.8 No No No 2  
## 3 237018 Yes 19.3 No No No 7  
## 4 52137 Yes 21.7 No No No 0  
## 5 81222 Yes 27.4 No Yes No 0  
## 6 249926 Yes 44.7 No No No 0  
## 7 58040 Yes 19.8 No No No 0  
## 8 176819 Yes 35.3 No No No 0  
## # ℹ 18 more variables: MentalHealth <dbl>, DiffWalking <fct>, Sex <fct>,  
## # Race <fct>, SleepTime <dbl>, Asthma <fct>, KidneyDisease <fct>,  
## # SkinCancer <fct>, Age <dbl>, DiabeticScore <dbl>, GenHealthScore <dbl>,  
## # .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,  
## # .std.resid <dbl>, index <int>

There are 8 observations that are very influential in the logistic regression model.

## Check the linear relationship between   
## continuous predictor variables and the logit of the outcome   
  
plot(glm.fit2)



There is an S-shaped pattern in the deviance residuals plot. This suggests a violation of the linearity assumption between the predictors and the logit of the response.

# Conducting Inference  
  
# Wald test  
summary(glm.fit2)

##   
## Call:  
## glm(formula = HeartDisease ~ BMI + Smoking + AlcoholDrinking +   
## Stroke + PhysicalHealth + MentalHealth + DiffWalking + Sex +   
## Race + SleepTime + Asthma + KidneyDisease + SkinCancer +   
## Age + DiabeticScore + GenHealthScore, family = binomial,   
## data = heart\_disease2, subset = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9024 -0.7139 -0.3600 0.7555 3.1484   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.9830230 0.0889318 -67.277 < 2e-16 \*\*\*  
## BMI 0.0115774 0.0016725 6.922 4.44e-12 \*\*\*  
## SmokingYes 0.3842467 0.0206177 18.637 < 2e-16 \*\*\*  
## AlcoholDrinkingYes -0.2408655 0.0460045 -5.236 1.64e-07 \*\*\*  
## StrokeYes 1.1675848 0.0396865 29.420 < 2e-16 \*\*\*  
## PhysicalHealth 0.0035364 0.0012773 2.769 0.005629 \*\*   
## MentalHealth 0.0061228 0.0013230 4.628 3.70e-06 \*\*\*  
## DiffWalkingYes 0.2472590 0.0274155 9.019 < 2e-16 \*\*\*  
## SexMale 0.7521179 0.0209918 35.829 < 2e-16 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.0593601 0.0758835 0.782 0.434066   
## RaceAsian -0.4203880 0.0891970 -4.713 2.44e-06 \*\*\*  
## RaceBlack -0.2434697 0.0414627 -5.872 4.31e-09 \*\*\*  
## RaceHispanic -0.1197466 0.0432327 -2.770 0.005609 \*\*   
## RaceOther 0.0224874 0.0577338 0.390 0.696905   
## SleepTime -0.0243946 0.0065157 -3.744 0.000181 \*\*\*  
## AsthmaYes 0.2819941 0.0287570 9.806 < 2e-16 \*\*\*  
## KidneyDiseaseYes 0.6160914 0.0413814 14.888 < 2e-16 \*\*\*  
## SkinCancerYes 0.1454886 0.0297218 4.895 9.83e-07 \*\*\*  
## Age 0.0525515 0.0007719 68.084 < 2e-16 \*\*\*  
## DiabeticScore 0.2492847 0.0123272 20.222 < 2e-16 \*\*\*  
## GenHealthScore 0.5101072 0.0120163 42.451 < 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: 83691 on 65694 degrees of freedom  
## Residual deviance: 60468 on 65674 degrees of freedom  
## AIC: 60510  
##   
## Number of Fisher Scoring iterations: 5

# Likelihood ratio test  
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

Anova(glm.fit2)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: HeartDisease  
## LR Chisq Df Pr(>Chisq)   
## BMI 47.8 1 4.756e-12 \*\*\*  
## Smoking 346.9 1 < 2.2e-16 \*\*\*  
## AlcoholDrinking 28.0 1 1.193e-07 \*\*\*  
## Stroke 946.5 1 < 2.2e-16 \*\*\*  
## PhysicalHealth 7.7 1 0.0056406 \*\*   
## MentalHealth 21.3 1 3.856e-06 \*\*\*  
## DiffWalking 81.1 1 < 2.2e-16 \*\*\*  
## Sex 1321.1 1 < 2.2e-16 \*\*\*  
## Race 62.6 5 3.505e-12 \*\*\*  
## SleepTime 14.0 1 0.0001816 \*\*\*  
## Asthma 95.5 1 < 2.2e-16 \*\*\*  
## KidneyDisease 228.9 1 < 2.2e-16 \*\*\*  
## SkinCancer 23.9 1 9.963e-07 \*\*\*  
## Age 5583.4 1 < 2.2e-16 \*\*\*  
## DiabeticScore 406.4 1 < 2.2e-16 \*\*\*  
## GenHealthScore 1881.2 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Confidence Intervals  
  
# Wald confidence intervals for the multiplicative effect on odds  
exp(confint.default(glm.fit2))

## 2.5 % 97.5 %  
## (Intercept) 0.002117909 0.003001269  
## BMI 1.008333905 1.014966269  
## SmokingYes 1.410348540 1.529065269  
## AlcoholDrinkingYes 0.718181649 0.860107121  
## StrokeYes 2.973681164 3.474216326  
## PhysicalHealth 1.001033481 1.006058129  
## MentalHealth 1.003535906 1.008754015  
## DiffWalkingYes 1.213520507 1.351199101  
## SexMale 2.035974683 2.210593672  
## RaceAmerican Indian/Alaskan Native 0.914508237 1.231322675  
## RaceAsian 0.551446193 0.782262388  
## RaceBlack 0.722718815 0.850267363  
## RaceHispanic 0.815070136 0.965593722  
## RaceOther 0.913320165 1.145273615  
## SleepTime 0.963517056 0.988443129  
## AsthmaYes 1.253113839 1.402640782  
## KidneyDiseaseYes 1.707423138 2.008117264  
## SkinCancerYes 1.091153051 1.225982096  
## Age 1.052363571 1.055552447  
## DiabeticScore 1.252477830 1.314485897  
## GenHealthScore 1.626703856 1.705159507

### Use the new model to make predictions.

# Making predictions  
contrasts(heart\_disease2$HeartDisease)

## Yes  
## No 0  
## Yes 1

glm.probs=predict(glm.fit2, test.df, type="response")  
predicted.classes <- ifelse(glm.probs > 0.5, "Yes", "No")  
  
# Assessing model accuracy  
(mean(predicted.classes == test.df$HeartDisease))

## [1] 0.7722236

### Use different pi\_0 as a cut-off point and create a confusion table.

# pi\_0 = 0.2  
glm.pred=rep("No", nrow(test.df)) # generate a vector with each element as "No", name this vector glm.pred  
glm.pred[glm.probs>0.2]="Yes" #if the predicted probability of heart disease > 0.5, assign predicted direction as "up"  
table(glm.pred,test.df$HeartDisease) # produce the confusion matrix

##   
## glm.pred No Yes  
## No 6356 463  
## Yes 4636 4969

mean(glm.pred==test.df$HeartDisease)

## [1] 0.6895397

# pi\_0 = 0.3  
glm.pred=rep("No", nrow(test.df)) # generate a vector with each element as "No", name this vector glm.pred  
glm.pred[glm.probs>0.3]="Yes" #if the predicted probability of heart disease > 0.5, assign predicted direction as "up"  
table(glm.pred,test.df$HeartDisease) # produce the confusion matrix

##   
## glm.pred No Yes  
## No 7758 1031  
## Yes 3234 4401

mean(glm.pred==test.df$HeartDisease)

## [1] 0.740319

# pi\_0 = 0.4  
glm.pred=rep("No", nrow(test.df)) # generate a vector with each element as "No", name this vector glm.pred  
glm.pred[glm.probs>0.4]="Yes" #if the predicted probability of heart disease > 0.5, assign predicted direction as "up"  
table(glm.pred,test.df$HeartDisease) # produce the confusion matrix

##   
## glm.pred No Yes  
## No 8821 1670  
## Yes 2171 3762

mean(glm.pred==test.df$HeartDisease)

## [1] 0.7661349

### Perform visualization of data and models.

library(ggplot2)  
library(broom)  
  
library(lessR)

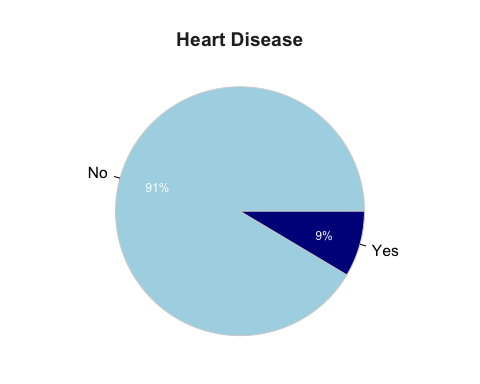
##   
## lessR 4.2.9 feedback: gerbing@pdx.edu   
## --------------------------------------------------------------  
## > d <- Read("") Read text, Excel, SPSS, SAS, or R data file  
## d is default data frame, data= in analysis routines optional  
##   
## Learn about reading, writing, and manipulating data, graphics,  
## testing means and proportions, regression, factor analysis,  
## customization, and descriptive statistics from pivot tables  
## Enter: browseVignettes("lessR")  
##   
## View changes in this and recent versions of lessR  
## Enter: news(package="lessR")  
##   
## Interactive data analysis  
## Enter: interact()

##   
## Attaching package: 'lessR'

## The following objects are masked from 'package:car':  
##   
## bc, recode, sp

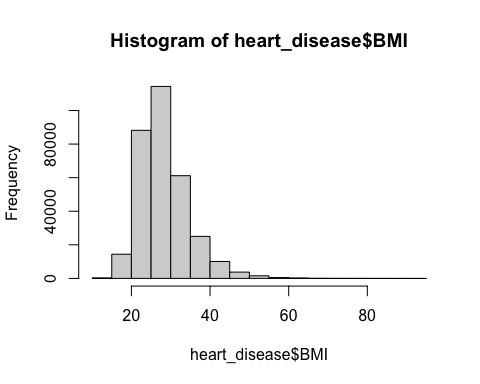
## The following objects are masked from 'package:dplyr':  
##   
## recode, rename

#Visual representation of proportion tables  
par(mfrow=c(1,1))  
  
PieChart(HeartDisease, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "Heart Disease")

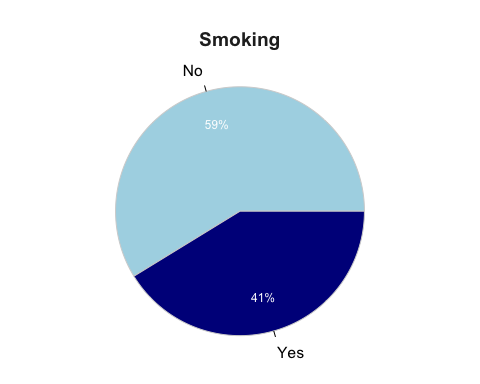


## >>> suggestions  
## PieChart(HeartDisease, hole=0) # traditional pie chart  
## PieChart(HeartDisease, values="%") # display %'s on the chart  
## PieChart(HeartDisease) # bar chart  
## Plot(HeartDisease) # bubble plot  
## Plot(HeartDisease, values="count") # lollipop plot   
##   
## --- HeartDisease ---   
##   
## No Yes Total   
## Frequencies: 292422 27373 319795   
## Proportions: 0.914 0.086 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 219675.018, df = 1, p-value = 0.000

hist(heart\_disease$BMI)

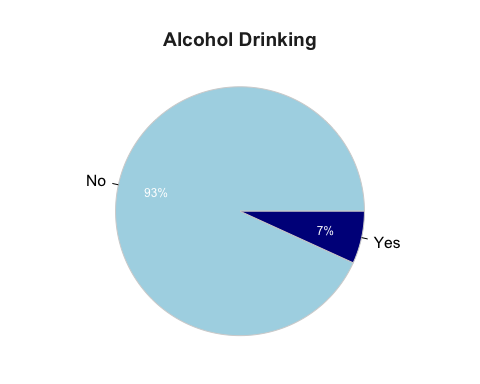


PieChart(Smoking, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "Smoking")



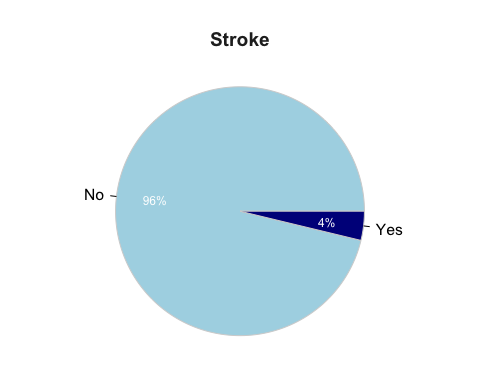
## >>> suggestions  
## PieChart(Smoking, hole=0) # traditional pie chart  
## PieChart(Smoking, values="%") # display %'s on the chart  
## PieChart(Smoking) # bar chart  
## Plot(Smoking) # bubble plot  
## Plot(Smoking, values="count") # lollipop plot   
##   
## --- Smoking ---   
##   
## No Yes Total   
## Frequencies: 187887 131908 319795   
## Proportions: 0.588 0.412 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 9798.929, df = 1, p-value = 0.000

PieChart(AlcoholDrinking, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "Alcohol Drinking")



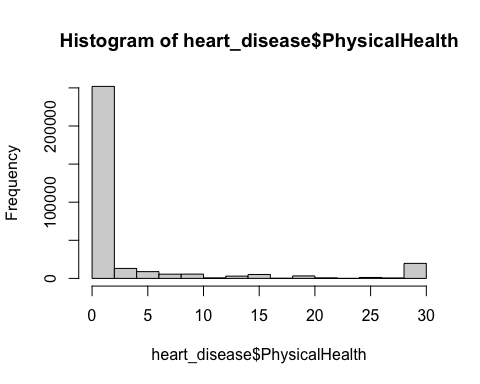
## >>> suggestions  
## PieChart(AlcoholDrinking, hole=0) # traditional pie chart  
## PieChart(AlcoholDrinking, values="%") # display %'s on the chart  
## PieChart(AlcoholDrinking) # bar chart  
## Plot(AlcoholDrinking) # bubble plot  
## Plot(AlcoholDrinking, values="count") # lollipop plot   
##   
## --- AlcoholDrinking ---   
##   
## No Yes Total   
## Frequencies: 298018 21777 319795   
## Proportions: 0.932 0.068 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 238618.772, df = 1, p-value = 0.000

PieChart(Stroke, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "Stroke")

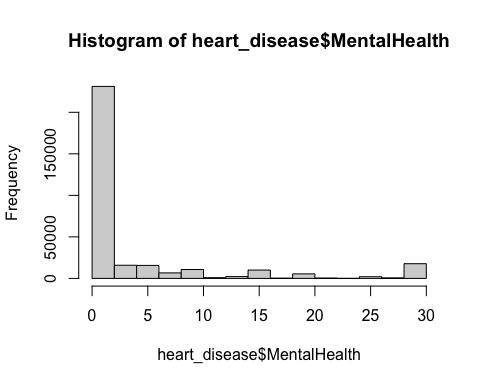


## >>> suggestions  
## PieChart(Stroke, hole=0) # traditional pie chart  
## PieChart(Stroke, values="%") # display %'s on the chart  
## PieChart(Stroke) # bar chart  
## Plot(Stroke) # bubble plot  
## Plot(Stroke, values="count") # lollipop plot   
##   
## --- Stroke ---   
##   
## No Yes Total   
## Frequencies: 307726 12069 319795   
## Proportions: 0.962 0.038 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 273340.927, df = 1, p-value = 0.000

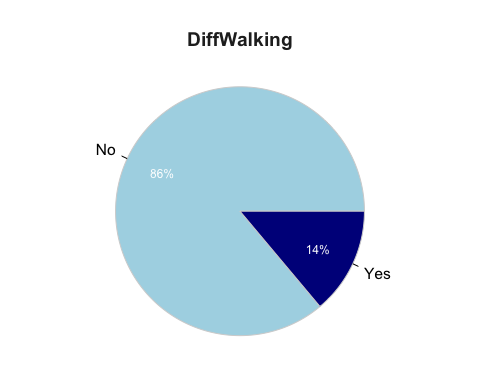
hist(heart\_disease$PhysicalHealth)



hist(heart\_disease$MentalHealth)

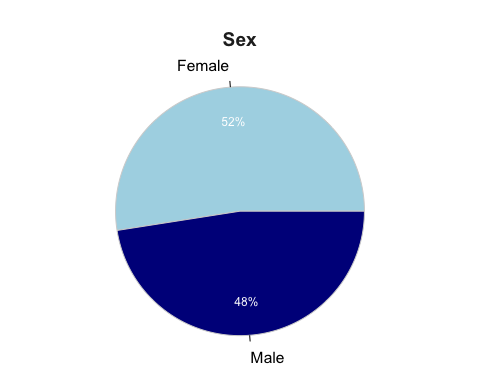


PieChart(DiffWalking, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "DiffWalking")



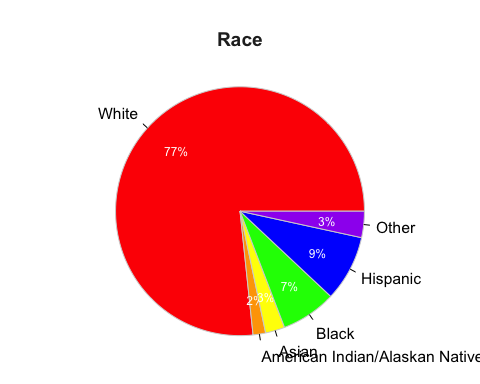
## >>> suggestions  
## PieChart(DiffWalking, hole=0) # traditional pie chart  
## PieChart(DiffWalking, values="%") # display %'s on the chart  
## PieChart(DiffWalking) # bar chart  
## Plot(DiffWalking) # bubble plot  
## Plot(DiffWalking, values="count") # lollipop plot   
##   
## --- DiffWalking ---   
##   
## No Yes Total   
## Frequencies: 275385 44410 319795   
## Proportions: 0.861 0.139 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 166823.905, df = 1, p-value = 0.000

PieChart(Sex, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "darkblue"), main = "Sex")



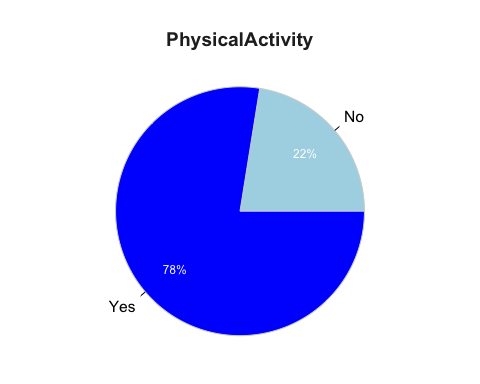
## >>> suggestions  
## PieChart(Sex, hole=0) # traditional pie chart  
## PieChart(Sex, values="%") # display %'s on the chart  
## PieChart(Sex) # bar chart  
## Plot(Sex) # bubble plot  
## Plot(Sex, values="count") # lollipop plot   
##   
## --- Sex ---   
##   
## Female Male Total   
## Frequencies: 167805 151990 319795   
## Proportions: 0.525 0.475 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 782.108, df = 1, p-value = 0.000

PieChart(Race, hole = 0, values = "%", data = heart\_disease,  
 fill = c("red", "orange", "yellow", "green", "blue", "purple"), main = "Race")



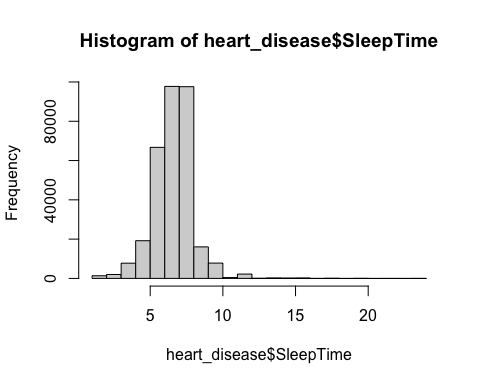
## >>> suggestions  
## PieChart(Race, hole=0) # traditional pie chart  
## PieChart(Race, values="%") # display %'s on the chart  
## PieChart(Race) # bar chart  
## Plot(Race) # bubble plot  
## Plot(Race, values="count") # lollipop plot   
##   
## --- Race ---   
##   
## Race Count Prop   
## -------------------------------------   
## White 245212 0.767   
## AmericnIndn/AlsknNtv 5202 0.016   
## Asian 8068 0.025   
## Black 22939 0.072   
## Hispanic 27446 0.086   
## Other 10928 0.034   
## -------------------------------------   
## Total 319795 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 836320.265, df = 5, p-value = 0.000

PieChart(PhysicalActivity, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "blue"), main = "PhysicalActivity")

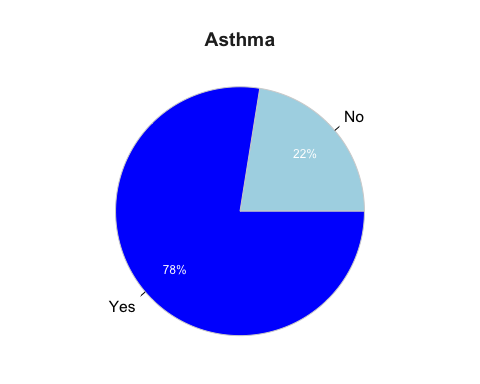


## >>> suggestions  
## PieChart(PhysicalActivity, hole=0) # traditional pie chart  
## PieChart(PhysicalActivity, values="%") # display %'s on the chart  
## PieChart(PhysicalActivity) # bar chart  
## Plot(PhysicalActivity) # bubble plot  
## Plot(PhysicalActivity, values="count") # lollipop plot   
##   
## --- PhysicalActivity ---   
##   
## No Yes Total   
## Frequencies: 71838 247957 319795   
## Proportions: 0.225 0.775 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 96993.080, df = 1, p-value = 0.000

hist(heart\_disease$SleepTime)

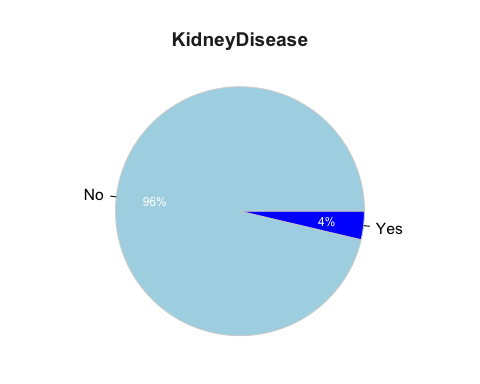


PieChart(PhysicalActivity, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "blue"), main = "Asthma")



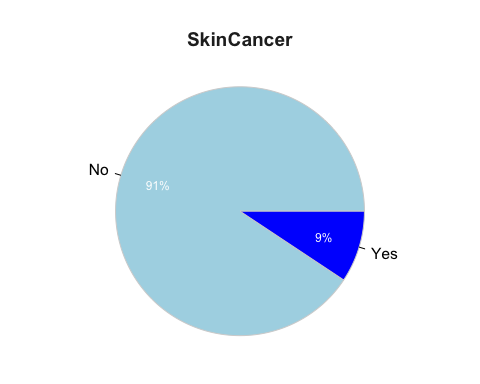
## >>> suggestions  
## PieChart(PhysicalActivity, hole=0) # traditional pie chart  
## PieChart(PhysicalActivity, values="%") # display %'s on the chart  
## PieChart(PhysicalActivity) # bar chart  
## Plot(PhysicalActivity) # bubble plot  
## Plot(PhysicalActivity, values="count") # lollipop plot   
##   
## --- PhysicalActivity ---   
##   
## No Yes Total   
## Frequencies: 71838 247957 319795   
## Proportions: 0.225 0.775 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 96993.080, df = 1, p-value = 0.000

PieChart(KidneyDisease, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "blue"), main = "KidneyDisease")



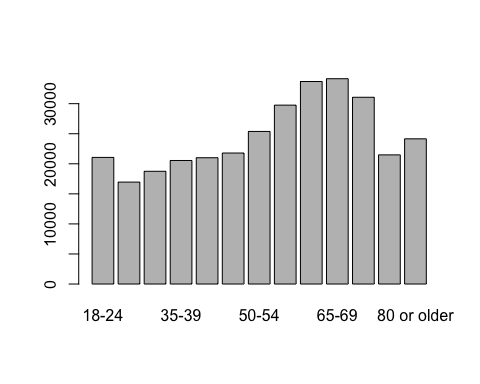
## >>> suggestions  
## PieChart(KidneyDisease, hole=0) # traditional pie chart  
## PieChart(KidneyDisease, values="%") # display %'s on the chart  
## PieChart(KidneyDisease) # bar chart  
## Plot(KidneyDisease) # bubble plot  
## Plot(KidneyDisease, values="count") # lollipop plot   
##   
## --- KidneyDisease ---   
##   
## No Yes Total   
## Frequencies: 308016 11779 319795   
## Proportions: 0.963 0.037 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 274414.422, df = 1, p-value = 0.000

PieChart(SkinCancer, hole = 0, values = "%", data = heart\_disease,  
 fill = c("lightblue", "blue"), main = "SkinCancer")

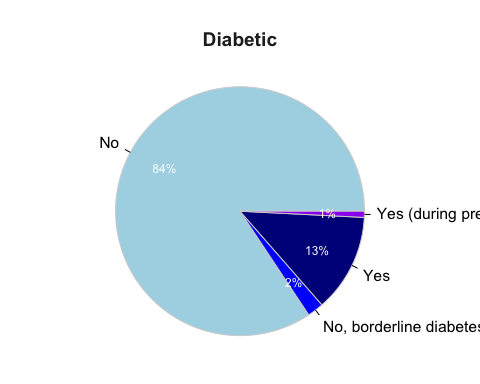


## >>> suggestions  
## PieChart(SkinCancer, hole=0) # traditional pie chart  
## PieChart(SkinCancer, values="%") # display %'s on the chart  
## PieChart(SkinCancer) # bar chart  
## Plot(SkinCancer) # bubble plot  
## Plot(SkinCancer, values="count") # lollipop plot   
##   
## --- SkinCancer ---   
##   
## No Yes Total   
## Frequencies: 289976 29819 319795   
## Proportions: 0.907 0.093 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 211640.784, df = 1, p-value = 0.000

plot(originalvars$heart\_disease.AgeCategory)

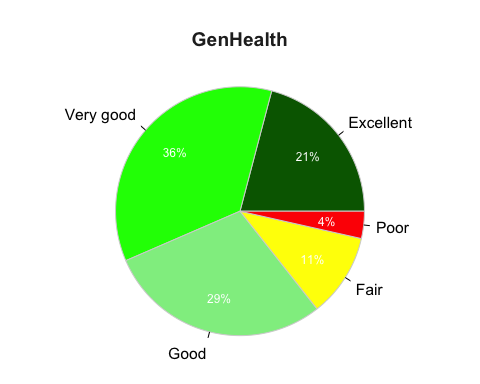


PieChart(heart\_disease.Diabetic, hole = 0, values = "%", data = originalvars,  
 fill = c("lightblue", "blue", "darkblue", "purple"), main = "Diabetic")



## >>> suggestions  
## PieChart(heart\_disease.Diabetic, hole=0) # traditional pie chart  
## PieChart(heart\_disease.Diabetic, values="%") # display %'s on the chart  
## PieChart(heart\_disease.Diabetic) # bar chart  
## Plot(heart\_disease.Diabetic) # bubble plot  
## Plot(heart\_disease.Diabetic, values="count") # lollipop plot   
##   
## --- heart\_disease.Diabetic ---   
##   
## hert\_dss.Dbtc Count Prop   
## -------------------------------------   
## No 269653 0.843   
## No,borderlindiabetes 6781 0.021   
## Yes 40802 0.128   
## Yes(duringpregnancy) 2559 0.008   
## -------------------------------------   
## Total 319795 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 611177.330, df = 3, p-value = 0.000

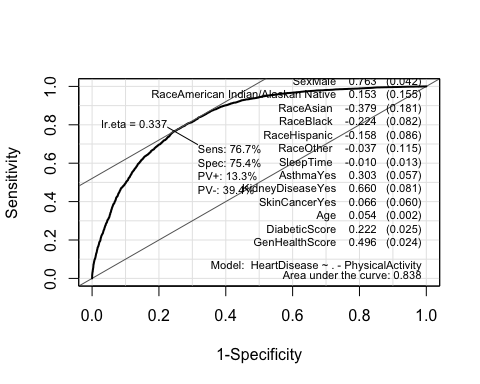
PieChart(heart\_disease.GenHealth, hole = 0, values = "%", data = originalvars,  
 fill = c("darkgreen", "green", "lightgreen", "yellow", "red"), main = "GenHealth")



## >>> suggestions  
## PieChart(heart\_disease.GenHealth, hole=0) # traditional pie chart  
## PieChart(heart\_disease.GenHealth, values="%") # display %'s on the chart  
## PieChart(heart\_disease.GenHealth) # bar chart  
## Plot(heart\_disease.GenHealth) # bubble plot  
## Plot(heart\_disease.GenHealth, values="count") # lollipop plot   
##   
## --- heart\_disease.GenHealth ---   
##   
## Excellent Very good Good Fair Poor Total   
## Frequencies: 66842 113858 93129 34677 11289 319795   
## Proportions: 0.209 0.356 0.291 0.108 0.035 1.000   
##   
## Chi-squared test of null hypothesis of equal probabilities   
## Chisq = 109142.970, df = 4, p-value = 0.000

### Plot the ROC curve, find AUC, and the best cutoff point for classification.

library(Epi)  
# ROC curve and AUC  
hd.roc = ROC(form=HeartDisease ~ . -PhysicalActivity, data=test.df, plot="ROC", MX = TRUE)



The best cutoff point for classification is 0.337.

# pi\_0 = 0.337  
glm.pred=rep("No", nrow(test.df)) # generate a vector with each element as "No", name this vector glm.pred  
glm.pred[glm.probs>0.337]="Yes" #if the predicted probability of heart disease > 0.5, assign predicted direction as "up"  
table(glm.pred,test.df$HeartDisease) # produce the confusion matrix

##   
## glm.pred No Yes  
## No 8226 1234  
## Yes 2766 4198

table(glm.pred,test.df$HeartDisease) %>% prop.table()

##   
## glm.pred No Yes  
## No 0.50085241 0.07513395  
## Yes 0.16841208 0.25560156

mean(glm.pred==test.df$HeartDisease)

## [1] 0.756454

Sensitivity is the true positive rate; P(predicted=1 | y=1) = 4198/(4198+1234) = 0.7728. Specificity is the true negative rate; P(predicted=0 | y=0) = 8226/(8226+2766) = 0.7484.

#Perform LOOCV and k-fold cross-validation.

library(boot)

##   
## Attaching package: 'boot'

## The following object is masked from 'package:car':  
##   
## logit

set.seed(1)  
# k-fold cross validation  
glm.cv <- glm(HeartDisease ~ .-PhysicalActivity, data=train.df, family = binomial)  
cv.error.10 <- cv.glm(train.df, glm.cv, K = 10)$delta[1]  
cv.error.10

## [1] 0.1510169

The training error is 0.15 and our test error is around 0.25. The difference is around 10%. This suggests our model is overfit to the training data.

#Try the probit link and the identity links to model data.

# Probit  
glm.probit <- glm(HeartDisease ~ .-PhysicalActivity, data = train.df, family = binomial(link = "probit"))  
summary(glm.probit)

##   
## Call:  
## glm(formula = HeartDisease ~ . - PhysicalActivity, family = binomial(link = "probit"),   
## data = train.df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.1277 -0.7307 -0.3497 0.7809 3.5007   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -3.4319522 0.0497633 -68.966  
## BMI 0.0060573 0.0009688 6.252  
## SmokingYes 0.2201179 0.0119938 18.353  
## AlcoholDrinkingYes -0.1356333 0.0262001 -5.177  
## StrokeYes 0.6744269 0.0228605 29.502  
## PhysicalHealth 0.0022140 0.0007505 2.950  
## MentalHealth 0.0035969 0.0007674 4.687  
## DiffWalkingYes 0.1536681 0.0163138 9.420  
## SexMale 0.4306214 0.0120918 35.613  
## RaceAmerican Indian/Alaskan Native 0.0378650 0.0441706 0.857  
## RaceAsian -0.2153245 0.0493014 -4.368  
## RaceBlack -0.1304751 0.0239399 -5.450  
## RaceHispanic -0.0485090 0.0244890 -1.981  
## RaceOther 0.0191942 0.0333636 0.575  
## SleepTime -0.0126798 0.0037992 -3.337  
## AsthmaYes 0.1654748 0.0167214 9.896  
## KidneyDiseaseYes 0.3645147 0.0242235 15.048  
## SkinCancerYes 0.0918728 0.0175938 5.222  
## Age 0.0298507 0.0004270 69.905  
## DiabeticScore 0.1478860 0.0072999 20.259  
## GenHealthScore 0.2953130 0.0069288 42.621  
## Pr(>|z|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## BMI 0.000000000405 \*\*\*  
## SmokingYes < 0.0000000000000002 \*\*\*  
## AlcoholDrinkingYes 0.000000225693 \*\*\*  
## StrokeYes < 0.0000000000000002 \*\*\*  
## PhysicalHealth 0.003176 \*\*   
## MentalHealth 0.000002768341 \*\*\*  
## DiffWalkingYes < 0.0000000000000002 \*\*\*  
## SexMale < 0.0000000000000002 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.391310   
## RaceAsian 0.000012567123 \*\*\*  
## RaceBlack 0.000000050340 \*\*\*  
## RaceHispanic 0.047608 \*   
## RaceOther 0.565087   
## SleepTime 0.000845 \*\*\*  
## AsthmaYes < 0.0000000000000002 \*\*\*  
## KidneyDiseaseYes < 0.0000000000000002 \*\*\*  
## SkinCancerYes 0.000000177117 \*\*\*  
## Age < 0.0000000000000002 \*\*\*  
## DiabeticScore < 0.0000000000000002 \*\*\*  
## GenHealthScore < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 83691 on 65694 degrees of freedom  
## Residual deviance: 60494 on 65674 degrees of freedom  
## AIC: 60536  
##   
## Number of Fisher Scoring iterations: 5

# Recode HeartDisease in training data into 0s and 1s  
new\_train <- train.df  
new\_test <- test.df  
new\_train$HeartDisease <- ifelse(new\_train$HeartDisease == "Yes", 1, 0)  
new\_test$HeartDisease <- ifelse(new\_test$HeartDisease == "Yes", 1, 0)  
  
# To get starting value we use the linear fit (least square method)   
mod.ident1=lm(HeartDisease~. -PhysicalActivity,data = new\_train)  
(g=summary(mod.ident1))

##   
## Call:  
## lm(formula = HeartDisease ~ . - PhysicalActivity, data = new\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.23407 -0.28805 -0.07864 0.31917 1.25755   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -0.36848765 0.01191588 -30.924  
## BMI 0.00023972 0.00025442 0.942  
## SmokingYes 0.05524216 0.00322313 17.139  
## AlcoholDrinkingYes -0.04019780 0.00652230 -6.163  
## StrokeYes 0.21839791 0.00619089 35.277  
## PhysicalHealth 0.00103439 0.00020892 4.951  
## MentalHealth 0.00059619 0.00020307 2.936  
## DiffWalkingYes 0.07209013 0.00469694 15.348  
## SexMale 0.11692315 0.00313893 37.249  
## RaceAmerican Indian/Alaskan Native 0.00247543 0.01177348 0.210  
## RaceAsian -0.03876712 0.01080354 -3.588  
## RaceBlack -0.03683535 0.00615024 -5.989  
## RaceHispanic -0.01593969 0.00594821 -2.680  
## RaceOther 0.00310952 0.00861923 0.361  
## SleepTime -0.00052752 0.00101790 -0.518  
## AsthmaYes 0.03982851 0.00445880 8.933  
## KidneyDiseaseYes 0.12191934 0.00670183 18.192  
## SkinCancerYes 0.04027963 0.00499504 8.064  
## Age 0.00703427 0.00009782 71.911  
## DiabeticScore 0.05016606 0.00208135 24.103  
## GenHealthScore 0.08165127 0.00181718 44.933  
## Pr(>|t|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## BMI 0.346092   
## SmokingYes < 0.0000000000000002 \*\*\*  
## AlcoholDrinkingYes 0.000000000717340622 \*\*\*  
## StrokeYes < 0.0000000000000002 \*\*\*  
## PhysicalHealth 0.000000739340876353 \*\*\*  
## MentalHealth 0.003327 \*\*   
## DiffWalkingYes < 0.0000000000000002 \*\*\*  
## SexMale < 0.0000000000000002 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.833470   
## RaceAsian 0.000333 \*\*\*  
## RaceBlack 0.000000002119038341 \*\*\*  
## RaceHispanic 0.007370 \*\*   
## RaceOther 0.718276   
## SleepTime 0.604292   
## AsthmaYes < 0.0000000000000002 \*\*\*  
## KidneyDiseaseYes < 0.0000000000000002 \*\*\*  
## SkinCancerYes 0.000000000000000751 \*\*\*  
## Age < 0.0000000000000002 \*\*\*  
## DiabeticScore < 0.0000000000000002 \*\*\*  
## GenHealthScore < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3931 on 65674 degrees of freedom  
## Multiple R-squared: 0.3054, Adjusted R-squared: 0.3052   
## F-statistic: 1444 on 20 and 65674 DF, p-value: < 0.00000000000000022

# We use a starting values from least square estimation  
strt=c(coef(mod.ident1)[1], coef(mod.ident1)[2], coef(mod.ident1)[3], coef(mod.ident1)[4], coef(mod.ident1)[5], coef(mod.ident1)[6], coef(mod.ident1)[7], coef(mod.ident1)[8], coef(mod.ident1)[9], coef(mod.ident1)[10], coef(mod.ident1)[11], coef(mod.ident1)[12], coef(mod.ident1)[13], coef(mod.ident1)[14],coef(mod.ident1)[15], coef(mod.ident1)[16], coef(mod.ident1)[17], coef(mod.ident1)[18], coef(mod.ident1)[19], coef(mod.ident1)[20], coef(mod.ident1)[21])  
  
mod.ident=glm(HeartDisease ~ . -PhysicalActivity, data = new\_train, start=strt, family = gaussian(link = "identity"))  
(g1=summary(mod.ident))

##   
## Call:  
## glm(formula = HeartDisease ~ . - PhysicalActivity, family = gaussian(link = "identity"),   
## data = new\_train, start = strt)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.23407 -0.28805 -0.07864 0.31917 1.25755   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -0.36848765 0.01191588 -30.924  
## BMI 0.00023972 0.00025442 0.942  
## SmokingYes 0.05524216 0.00322313 17.139  
## AlcoholDrinkingYes -0.04019780 0.00652230 -6.163  
## StrokeYes 0.21839791 0.00619089 35.277  
## PhysicalHealth 0.00103439 0.00020892 4.951  
## MentalHealth 0.00059619 0.00020307 2.936  
## DiffWalkingYes 0.07209013 0.00469694 15.348  
## SexMale 0.11692315 0.00313893 37.249  
## RaceAmerican Indian/Alaskan Native 0.00247543 0.01177348 0.210  
## RaceAsian -0.03876712 0.01080354 -3.588  
## RaceBlack -0.03683535 0.00615024 -5.989  
## RaceHispanic -0.01593969 0.00594821 -2.680  
## RaceOther 0.00310952 0.00861923 0.361  
## SleepTime -0.00052752 0.00101790 -0.518  
## AsthmaYes 0.03982851 0.00445880 8.933  
## KidneyDiseaseYes 0.12191934 0.00670183 18.192  
## SkinCancerYes 0.04027963 0.00499504 8.064  
## Age 0.00703427 0.00009782 71.911  
## DiabeticScore 0.05016606 0.00208135 24.103  
## GenHealthScore 0.08165127 0.00181718 44.933  
## Pr(>|t|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## BMI 0.346092   
## SmokingYes < 0.0000000000000002 \*\*\*  
## AlcoholDrinkingYes 0.000000000717340622 \*\*\*  
## StrokeYes < 0.0000000000000002 \*\*\*  
## PhysicalHealth 0.000000739340876353 \*\*\*  
## MentalHealth 0.003327 \*\*   
## DiffWalkingYes < 0.0000000000000002 \*\*\*  
## SexMale < 0.0000000000000002 \*\*\*  
## RaceAmerican Indian/Alaskan Native 0.833470   
## RaceAsian 0.000333 \*\*\*  
## RaceBlack 0.000000002119038341 \*\*\*  
## RaceHispanic 0.007370 \*\*   
## RaceOther 0.718276   
## SleepTime 0.604292   
## AsthmaYes < 0.0000000000000002 \*\*\*  
## KidneyDiseaseYes < 0.0000000000000002 \*\*\*  
## SkinCancerYes 0.000000000000000751 \*\*\*  
## Age < 0.0000000000000002 \*\*\*  
## DiabeticScore < 0.0000000000000002 \*\*\*  
## GenHealthScore < 0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1545494)  
##   
## Null deviance: 14613 on 65694 degrees of freedom  
## Residual deviance: 10150 on 65674 degrees of freedom  
## AIC: 63789  
##   
## Number of Fisher Scoring iterations: 1

#Which model works better for this data?

# Predictions using identity link  
ident.probs <- predict(mod.ident, newdata = new\_test)  
  
ident.pred=rep(0, nrow(new\_test))  
ident.pred[ident.probs>0.337] <- 1   
table(ident.pred,new\_test$HeartDisease) # produce the confusion matrix

##   
## ident.pred 0 1  
## 0 7611 950  
## 1 3381 4482

mean(ident.pred==new\_test$HeartDisease)

## [1] 0.7363005

# Predictions using probit link  
probit.probs <- predict(glm.probit, newdata = test.df)  
  
probit.pred=rep("No", nrow(test.df))  
probit.pred[probit.probs>0.337] <- "Yes"   
table(probit.pred,test.df$HeartDisease) # produce the confusion matrix

##   
## probit.pred No Yes  
## No 10296 3362  
## Yes 696 2070

mean(probit.pred==test.df$HeartDisease)

## [1] 0.7529226

The logit model accuracy is 0.756454, the identity link model accuracy is 0.7363005, and the probit model accuracy is 0.7529226.

#Write a report