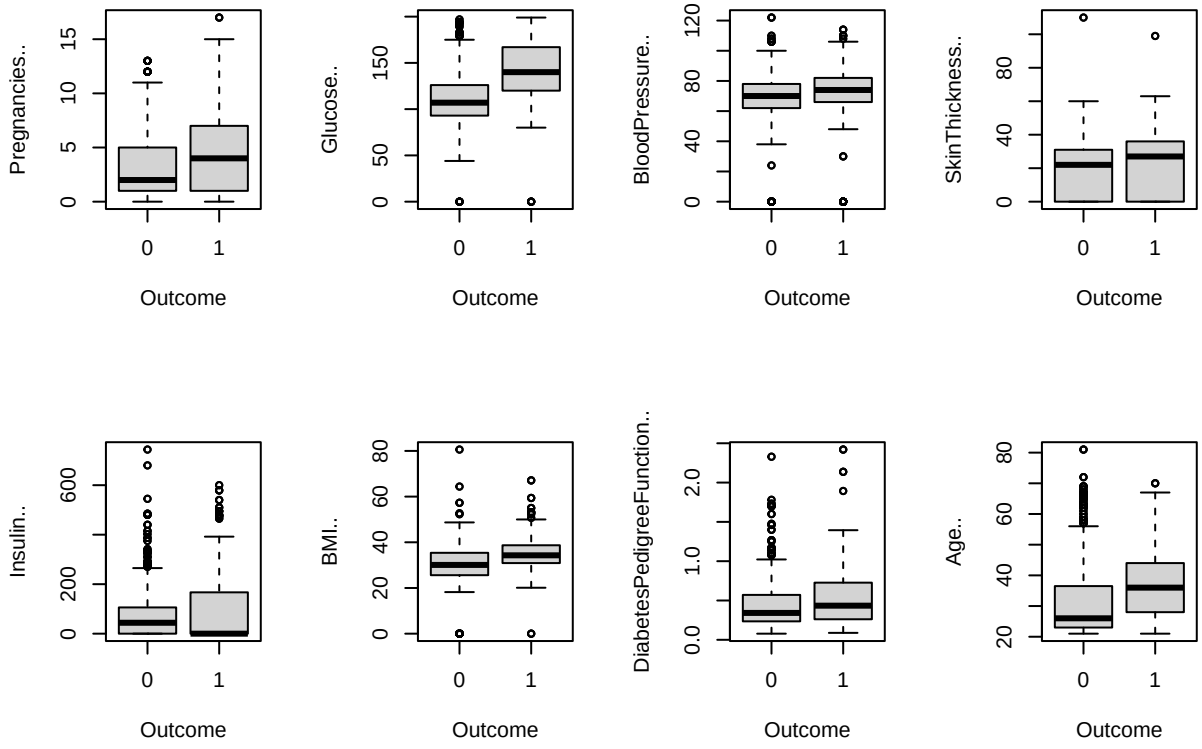
STAT4360_miniproject3

John Kenney

10/19/2021

Section 1

Question 1 (a) From the box plots we can see that while some of the distributions are similar for both responses we can see that there medians are not exactly the same. Therefore we can use each of these predictors for modeling our data.



(b) The best model I found is using all the predictors except skin thickness. After fitting the full model every predictor had a p-value less than 0.05 except skin thickness which had a p-value of 0.90244. I next created a reduced model leaving out Skin thickness and preformed an anova test to see if skin thickness was significant to our model. After performing the anova test I got a p-value of 0.9024 which does not reject our null hypothesis that skin thickness was not significant to our model.

(c) predicted Outcome = -8.027314632 + 0.126370737(Pregnancies..) + 0.033680984(Glucose..) - 0.009580617(BloodPressure..) - 0.001212277(Insulin..) + 0.077874259(BMI..) + 0.889494650(DiabetesPedigreeFunction..) + 0.012894361(Age..)

The train error rate of the best model is shown below with an error rate of 21.6%.

For the coefficient of Glucose we notice a 3.4% increase of the odds getting diabetes for one unit increase of Glucose. For the coefficient of Age we notice a 1.3% increase of the odds of getting diabetes for one unit increase of Age.

```
##                                Estimate   Std. Error        2.5 %        97.5 %
## (Intercept)                -8.027314632 0.4306244275 -8.889630784 -7.2009252668
## Pregnancies..               0.126370737 0.0199944440  0.087447559  0.1658700222
## Glucose..                   0.033680984 0.0022019516  0.029435255  0.0380709843
## BloodPressure..            -0.009580617 0.0032012505 -0.015885768 -0.0033221648
## Insulin..                  -0.001212277 0.0005228432 -0.002241105 -0.0001893038
## BMI..                       0.077874259 0.0084946154  0.061474284  0.0947952879
## DiabetesPedigreeFunction..  0.889494650 0.1855205018  0.527470753  1.2549028449
## Age..                       0.012894361 0.0056879063  0.001711033  0.0240290378
```

Question 2

(a) The error rate is .216 for the full model fitted on all the data. The sensitivity for this model is 0.5673. The specificity of this model is 0.8967.

(b) From my LOOCV implementation on the full model I get a test error rate of .2195.

(c) Using LOOCV from the caret package I get a test error rate of .2195. Which is the same as the test error rate I got using my implementation of the LOOCV.

(d) From the model I found that was best in question 1 the LOOCV error rate is 0.2185 using the caret package.

(e) I ran LDA through the caret package with LOOCV being the validation measure and I got a test error rate of 0.223.

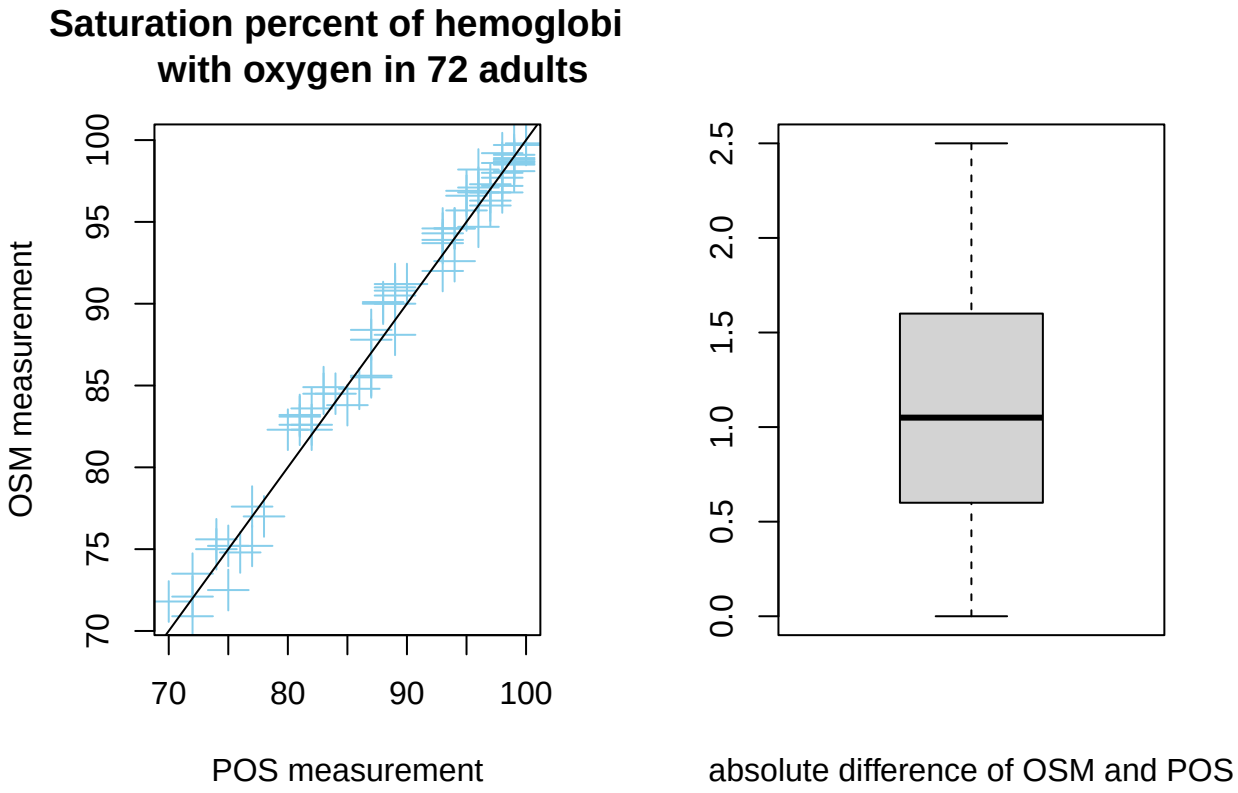
(f) Using the caret package and LOOCV I got an error rate of 0.2445 for QDA.

(g) I found that the optimal k using LOOCV was k = 7. The LOOCV error rate is 0.1715 for knn.

(h) Looking at each of LOOCV error rates for each classifier I find that KNN has the lowest error rate with 0.1715. Therefore, I can conclude from the data given on diabetes and on the decision to use LOOCV as the criteria to compare each of my models that KNN is best to predict the outcome of whether a person has diabetes.

Question 3 (a)

We can see from the scatter plot and the boxplot that while for the most part these two methods agree because they follow the 45 degree line pretty closely and that the distribution of their absolute differences are between 0 and 2.5 we can see that there are differences. These differences can be clearly seen in the scatter plot from the fact that points do not all fall on the 45 degree line, and for the box plot of the absolute differences we can see that we see that most of the absolute differences fall between .6 and 1.6. Therefore we do not have perfect agreement between the two methods of OSM and POS, but we still have a reasonably good model.



(b)

It is true that with a smaller theta We would get a better agreement because the closer we get to zero the less the two measures differ from each other.

(c) The point estimate I calculated is $\hat{\theta} = 2$ using p = .9 specified in part b.

(d)
Using my method for performing bootstrap I got:
Estimate of $\hat{\theta}$: 2.0015 Bias of $\hat{\theta}$: 0.0015
SE of $\hat{\theta}$: 0.1303
The 95% upper confidence bound for θ is 2.2.

From the results above we can say that we are 95% confident that θ is less than 2.2. This is great considering the 95% confidence upper bound for our original sample was 2.2225.

(e) Using the boot package I got: Bootstrap Statistics : original bias std. error t1* 2 0.00476 0.1257742 The 95% upper confidence bound for θ is 2.20.

From the results we can see that for the most part my implementation and boots implementation is similar. I had a lower bias and higher standard deviation, but this can be attributed to randomly selecting items from the original sample for each of the new samples. For both methods although different in implementation give the same upper bound therefore we can conclude that we are 95% confident that θ is less than 2.2

(f) I would say that agreement between these two methods is reasonably good. In practice, I cannot say if they could be used interchangeably but if an estimated absolute difference of 2 + small bias, standard error of around .126, and a 95% confidence upper bound of theta(TDI) is less than 2.2 is a reasonable amount of error between the two methods then they are interchangeable. If you would like a smaller theta of the absolute difference measure between these two methods more data is needed.

Section 2 q1 code

```
#loads library and sets working directory
library(ISLR)
library(ggplot2)
library(MASS)
library(lattice)
library(caret)
library(MethComp)
library(boot)
setwd("C:/Users/John/Documents/R/Programs/Stat4360/MiniProjects/Miniproject3/")
#setwd("C:/Users/jkenn/OneDrive/Documents/R/Stat4360/miniprojects/miniproject3/")
#loads data
diabetes <- read.csv("diabetes.csv")
str(diabetes)
```

```
#plots box plots for each predictor against its outcome
par(mfrow = c(2, 4))
boxplot(diabetes[, "Pregnancies.."] ~ diabetes[,"Outcome"], ylab = "Pregnancies..",
        xlab = "Outcome", ylim = c( 0, 17))
boxplot(diabetes[, "Glucose.."] ~ diabetes[,"Outcome"], ylab = "Glucose..",
        xlab = "Outcome", ylim = c(0, 199))
boxplot(diabetes[, "BloodPressure.."] ~ diabetes[,"Outcome"], ylab = "BloodPressure..",
        xlab = "Outcome", ylim = c(0, 122))
boxplot(diabetes[, "SkinThickness.."] ~ diabetes[,"Outcome"], ylab = "SkinThickness..",
        xlab = "Outcome", ylim = c( 0, 110))

boxplot(diabetes[, "Insulin.."] ~ diabetes[,"Outcome"], ylab = "Insulin..",
        xlab = "Outcome", ylim = c(0, 744 ))
boxplot(diabetes[, "BMI.."] ~ diabetes[,"Outcome"], ylab = "BMI..",
        xlab = "Outcome", ylim = c(0.0, 80.6))
boxplot(diabetes[, "DiabetesPedigreeFunction.."] ~ diabetes[,"Outcome"],
        ylab = "DiabetesPedigreeFunction..", xlab = "Outcome", ylim = c(0.078, 2.420))
boxplot(diabetes[, "Age.."] ~ diabetes[,"Outcome"], ylab = "Age..",
        xlab = "Outcome", ylim = c( 21, 81))
```

```
#fits full model of logistic regression
full_logit_dia <- glm(Outcome ~., family = binomial, data = diabetes)
summary(full_logit_dia)
```

```
# fits logistic model without skinthickness
reduced_logit_dia <- glm(Outcome ~Pregnancies.. + Glucose.. + BloodPressure.. +
                           Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age..,
                         family = binomial, data = diabetes)
summary(reduced_logit_dia)
```

```
#compares full and reduced model to make sure that skinthickness is not significant to our model
anova(reduced_logit_dia, full_logit_dia, test = "Chisq")
```

```
# predicts the error of our model on training data and outputs various summary output
lr.prob <- predict(reduced_logit_dia, diabetes, type = "response")
lr.pred <- ifelse(lr.prob >=0.5,1,0)
train_err_rate <- mean(lr.pred != diabetes[, "Outcome"])
```

```
table1 <- cbind(summary(reduced_logit_dia)$coef[,1:2],
                confint(reduced_logit_dia,level = 0.95))
table1
paste("Train error rate: ",train_err_rate)
exp(0.033680984)#glucose
exp(0.012894361)#age
```

q2 code

```
#refits full logistic model
full_logit_dia <- glm(Outcome ~., family = binomial, data = diabetes)
summary(full_logit_dia)
```

```
# Estimated probabilities for test data
lr.prob <- predict(full_logit_dia, diabetes, type = "response")
# Predicted classes (using 0.5 cutoff)
lr.pred <- ifelse(lr.prob >= 0.5, 1, 0)
# Confusion matrix and (sensitivity, specificity)
# `+' =1, `-' = 0

table(lr.pred, diabetes[, "Outcome"])

paste("sensitivity: ",(388/(388+296)))
paste("specificity: ",(1180/(1180+136)))
# Test error rate
paste("error rate",mean(lr.pred != diabetes[, "Outcome"]))
```

```
#my own LOOCV
#sets n = k for LOOCV
n = 2000
err_rate <- c()#empty array to hold error
for (i in 1:n) {
  train_2b <- diabetes[-c(i),]#leave out one
  temp.prob <- predict(glm(Outcome ~., family = binomial, data = train_2b),
                       diabetes[i,-c(9)], type = "response")
  #predicts on the one you left out
  temp.pred <- ifelse(temp.prob >=0.5,1,0)
  err_rate <- c(err_rate,(temp.pred - diabetes[i, "Outcome"])^2)
  #calcs the error^2 of each prediction
}
```

```
paste("error rate from our LOOCV: ",sum(err_rate)/n)#outputs the MSE of my LOOCV
```

```
#Uses carets package to model the data and preform LOOCV
set.seed(1)
train_control <- trainControl(method="LOOCV")
# train the model
model <- train(as.factor(Outcome)~., data=diabetes, trControl=train_control, method="glm",
               family=binomial())
print(model)
```

```
#prints LOOCV error
paste("error rate from caret package:",(1-model$results[2]))
```

```
set.seed(1)
#calcs the LOOCV error for the model proposed in q1
train_control <- trainControl(method="LOOCV")
# train the model
model_d <- train(as.factor(Outcome) ~Pregnancies.. + Glucose.. + BloodPressure.. + Insulin..
                 + BMI.. + DiabetesPedigreeFunction.. + Age.., data=diabetes,
                 trControl=train_control, method="glm",family=binomial())
print(model_d)
```

```
paste("LOOCV error rate from on best model using glm from Q1:",(1-model_d$results[2]))
```

```
set.seed(1)
#calcs the LOOCV error rate of LDA of the full model.
train_control <- trainControl(method="LOOCV")
# train the model
model_e <- train(as.factor(Outcome) ~., data=diabetes,
                 trControl=train_control, method="lda")
print(model_e)
```

```
paste("LOOCV error rate using LDA and full parameters:",(1-model_e$results[2]))
```

```
set.seed(1)
#calcs the LOOCV error rate of QDA of the full model using caret.
train_control <- trainControl(method="LOOCV")
# train the model
model_f <- train(as.factor(Outcome) ~., data=diabetes,
                 trControl=train_control, method="qda")
print(model_f)
```

```
paste("LOOCV error rate using QDA and full parameters:",(1-model_f$results[2]))
```

```
set.seed(1)
#calcs the LOOCV error rate of knn of the full model using caret.
train_control <- trainControl(method="LOOCV")
# train the model
model_g <- train(as.factor(Outcome) ~., data=diabetes,
                 trControl=train_control, method="knn")
print(model_g)
```

```
paste("LOOCV error rate using knn with full parameters with the best k being",
      (model_g$results)[1][2,1],"is",(1-model_g$results[2][2,1]))
```

q3 code

```
##question 3
#load in the data
oxygen_saturation <- read.csv("oxygen_saturation.txt",sep = "\t")
```

```r
#View(oxygen_saturation)
#get the range of predictors
range(oxygen_saturation$pos)
range(oxygen_saturation$osm)
```

```r
#plots the scatter plot of the two measures on the left and adds in a
#line representing prefect symmetry between the two measures
#on the right shows a box plot of D the absolute difference between the two measures
par(mfrow = c(1, 2))
plot(oxygen_saturation$pos, oxygen_saturation$osm,
     pch=3,
     cex=2,
     col="skyblue",
     xlab="POS measurement", ylab="OSM measurement",
     main="Saturation percent of hemoglobin
     with oxygen in 72 adults"
     )
abline(0,1,col = "black")
D <- abs(oxygen_saturation$pos - oxygen_saturation$osm)

boxplot(D, xlab = "absolute difference of OSM and POS")
```

```r
#quantiles the box plot
summary(D)
```

```r
#point estimate
sample_estim <- quantile(D,probs=.9)
sample_estim
#upper bound of 95% CI
quantile(D,probs=.975)[[1]]
```

```r
#function to get theta
theta.fn=function (data ,index){
  return(quantile(data[index],probs=.9))#takes the .9 quantile as theta
}
# my bootstrap
estimates <- 1000
set.seed(1)
theta <- c()
for(i in 1:estimates){
  set.seed(i)
  # calcs 1000 estimates of theta by using sample with replacement
  theta <- c(theta,theta.fn(D,sample(1:length(D),length(D), replace=T)))
}
theta_estim <- mean(theta)
paste("theta hat:",theta_estim)

theta_bias <- (theta_estim-sample_estim)
paste("bias of thetahat:",theta_bias)

theta_sd <- sd(theta)
paste("Standard error of theta hat:",theta_sd)
```

```
paste("I am 95% confident that the absolute difference between OSM and POS will be less
      than",quantile(theta,probs=.975)[[1]])
```

```
#plots the boxplots of bootstrap and and D
par(mfrow = c(1, 2))
boxplot(theta, xlab = "thetas from 1000 bootstraps")
boxplot(D, xlab = "absolute difference of OSM and POS")
```

```
set.seed(1)
# performs bootstrap using the boot package
boot1 <- boot(D ,theta.fn,R=1000)
boot1
```

```
plot(boot1)#plots boot

boot.ci(boot1, type = "perc")# to get 95% upper bound CI
```