Diabetes in the Pima Tribe

Melissa Swann

April 7, 2020

Data Preparation

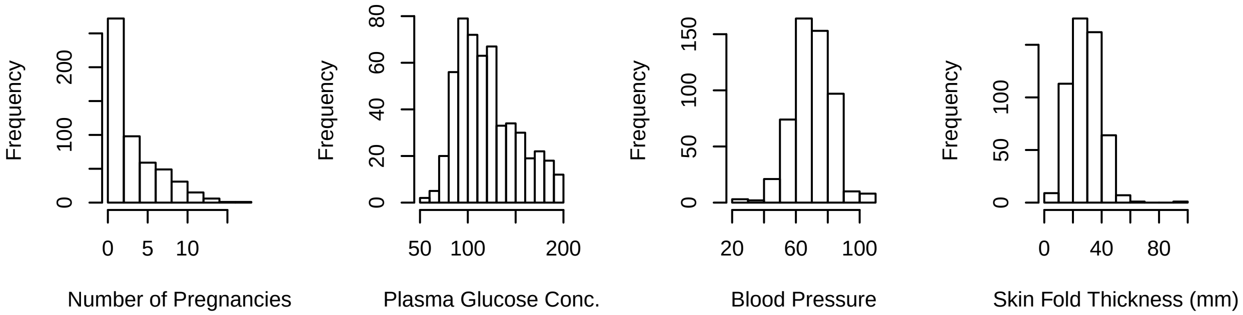
```r
Dtrain <- Pima.tr # Training set
Dtest <- Pima.te # Test set
D <- rbind(Dtrain,Dtest) # Combined dataset
Dtrain$type_num <- ifelse(Dtrain$type == "No",0,1)
Dtest$type_num <- ifelse(Dtest$type == "No",0,1)
D$type_num <- ifelse(D$type == "No",0,1)
```

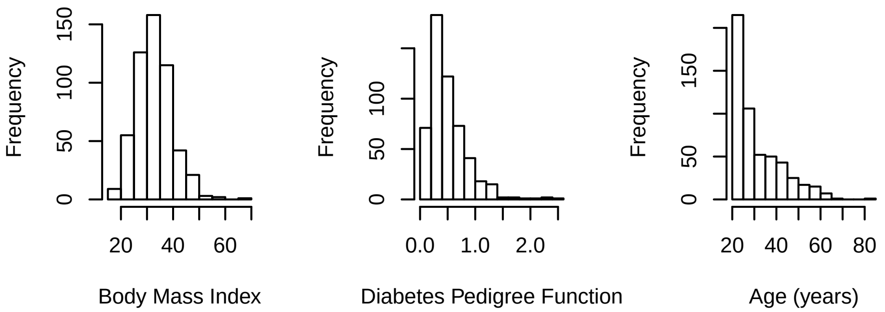
Introduction

Diabetes is a disease where a person's body does not make or use insulin well. This causes blood sugar levels to be high and the glucose in the blood doesn't reach the cells of the body like it should be. Over time high glucose levels damage nerve and blood vessels, leading to heart disease, stroke, blindness, kidney disease, and gum infections. I will be using a dataset consisting of 532 female members of the Pima Tribe (a group of Native Americans living in an area consisting of what is now central and southern Arizona). This tribe has the highest reported prevalence of diabetes of any population in the world so it could be helpful to use this dataset to find linked trends between diabetes and different medical variables. It may also be helpful for predicting diabetes and advising at-risk patients to take preventative measures like changing their diet and exercise habits before more serious measures like insulin are required.

Data Exploration

There are 7 variables available in the dataset for each subject to use to predict diabetes. Each has their own distribution and background connection to diabetes, as displayed below, which could give insight into how well we can use them to predict diabetes.





Now I'll look more closely at what each variable means to get more of a sense of the context.

```
summary(D$npreg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   3.517   5.000  17.000
```

Npreg: the number of pregnancies an individual has had. The median of individuals have 2 pregnancies, skewing the distribution towards that lower bound of 0-2. Some women in the data set have had as many as 17 pregnancies. These larger outliers pull the mean to about 4 pregnancies to be higher than the median. There isn't a direct connection between the number of pregnancies and diabetes in the literature. The more pregnancies one has can lead to a higher chance of getting gestational diabetes and diabetes itself can sometimes affect the fertility of an individual. A potential connection between a lower number of pregnancies and diabetes could be found but there are many other factors that affect the number of pregnancies that an individual has.

```
summary(D$glu)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   56.00   98.75  115.00  121.03  141.25  199.00
```

Glu: This is the concentration of glucose in the plasma of the subject as calculated by an oral glucose tolerance test. This is the blood sugar level. In this population the glucose levels range from 56 to 199 with the mean individual having a concentration of 121. In general normal glucose levels (with the test done after fasting) are around 70-100. Levels between 100-120 are considered pre-diabetic values while levels above 125 are considered diabetic. Since blood glucose levels are a metric already used for people with diabetes to track their status it seems likely that this variable should help predict diabetes.

```
summary(D$bp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   64.00   72.00   71.51   80.00  110.00
```

bp: This is the diastolic blood pressure of an individual which is the blood pressure when your heart muscle is between beats. Normal diastolic blood pressure levels are between 60-80. Once you go above 80-90 you have high blood pressure and anything above 120 is really bad. In this population the mean blood pressure was at 71 with blood pressures as low as 24 or as high as 110. In general the distribution was pretty normal without much skew.

High blood pressure seems to come hand in hand with diabetes because they both originate from similar conditions like obesity, insulin resistance, etc.

```
summary(D$skin)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   22.00   29.00   29.18   36.00   99.00
```

Skin: This is the skin fold thickness of the triceps in mm. This is a measure that is supposed to give information about the fat reserves in the body by assuming that the body fat is equally distributed over the body and the thickness is a direct measure for subcutaneous fat. There seems to be an outlier at 99 mm in the dataset. The rest of the subjects fall between 0 and 60 mm with the mean individual having a skin fold thickness of 29 mm. A larger value would correspond to more body fat and be a potential sign of being overweight or obese.

```
summary(D$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.20   27.88   32.80   32.89   36.90   67.10
```

Bmi: This is the Body mass index calculated by dividing the weight of the subject by their height squared. It is supposed to be used screening tool used to identify individuals who are underweight, overweight, or obese although it can be a flawed metric. A “healthy” BMI is between 18.5–24.9. A BMI between 25.0–29.9 is considered overweight, above 30 is considered obese ad below 18.5 is underweight.

In this population the mean BMI is 33 with a pretty even normal distribution around this mean. THe lowest MI present is 18.2 while the maximum BMI is 67.9.

```
summary(D$ped)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0850  0.2587  0.4160  0.5030  0.6585  2.4200
```

Ped: This is the diabetes pedigree function which seems to be a variable specific to this study of the Pima population. As stated in the description “It utilizes information from a person’s family history to predict how diabetes will affect that individual.” It is a likelihood so a higher number would point to a higher risk one might have with the onset of diabetes mellitus. The mean ped value is 0.5. There are a few outlier values with ped values as high as 2.4.

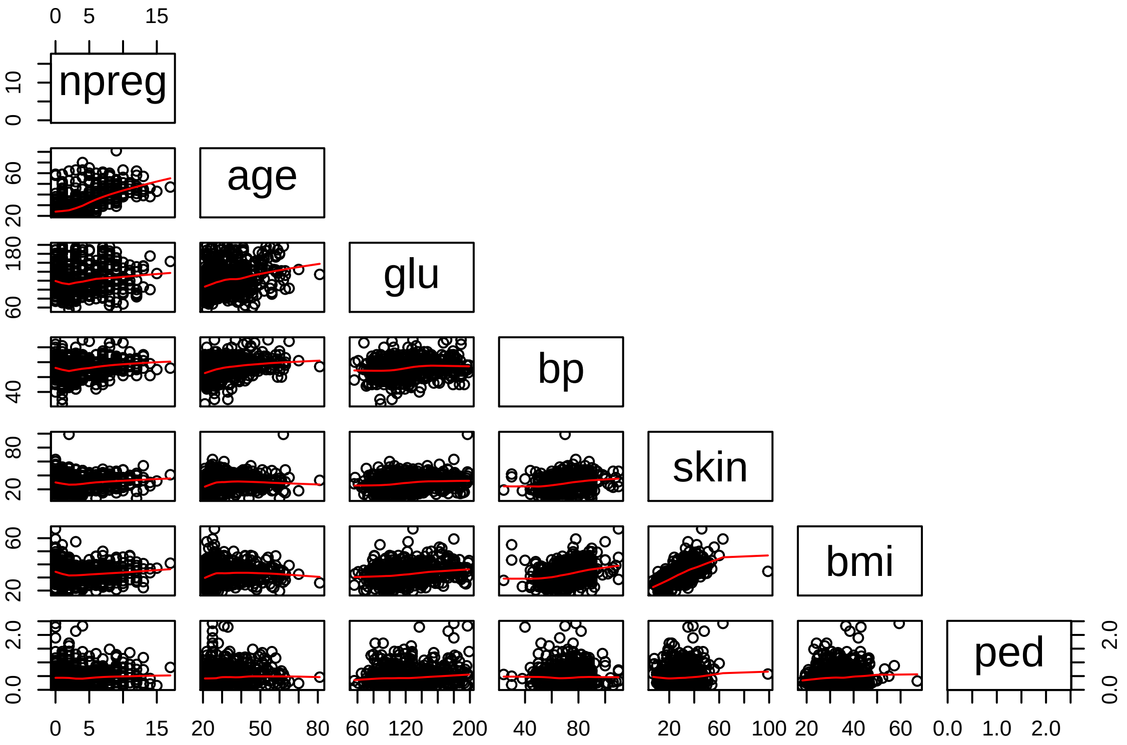
```
summary(D$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.00   23.00   28.00   31.61   38.00   81.00
```

Age: This is the age in years of the individuals. It spans from females as young as 21 years old and as old as 81. Most of the population seems to be on the younger side with a mean of about 32 years of age that is pulled by the larger age outliers and a median that is lower at 28 years of age.

Understanding each variable available for predicting diabetes in the female Pima population is important to see how the variables relate with each other.

```r
pairs(~npreg+age+glu+bp+skin+bmi+ped, data=D,lower.panel=panel.smooth, upper.panel = NULL)
```

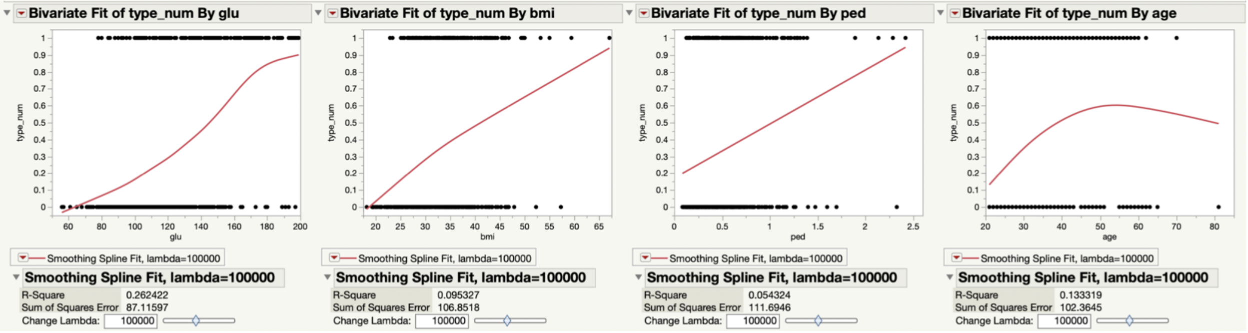


Looking at the scatterplot matrix, most of the variables don't seem to be correlated with one another. The relationships that stand out the most are between skin and bmi and between npreg and age. Skin is positively related to bmi, which makes sense since both explain how over- or underweight someone is so they are essentially sharing the same trend in information in two different ways. It probably isn't necessary to use both in the model and bmi seems more accurate, so something to think about. Age is positively related to the number of pregnancies an individual has had. This also makes sense because an older person has had more time to accumulate pregnancies than someone who is 21 years old, for example, who has not had as much time to accumulate multiple pregnancies.

There also seems to be a very slight positive relationship between age and glucose. As the age of the individual increases so does the blood sugar level.

Finally, there seems to be an outlier point if we look at the plot for Skin vs BMI plot. There is an individual with a large skin value of 99 and a BMI that is only around 30. It is very weird for a person to have such a tricep thickness and also have such a low BMI in comparison to the other members of the population. There isn't enough reason to take this point out of the population but it is important to keep this point in mind when looking at the patterns associated with the skin variable.

Lastly, using JMP we can get a preliminary sense of the relationships between the variables and the response through smoothing spline fits.



Bivariate Fit of type_num By glu

Smoothing Spline Fit, lambda=100000

R-Square 0.262422
Sum of Squares Error 87.11597
Change Lambda: 100000

Bivariate Fit of type_num By bmi

Smoothing Spline Fit, lambda=100000

R-Square 0.095327
Sum of Squares Error 106.8518
Change Lambda: 100000

Bivariate Fit of type_num By ped

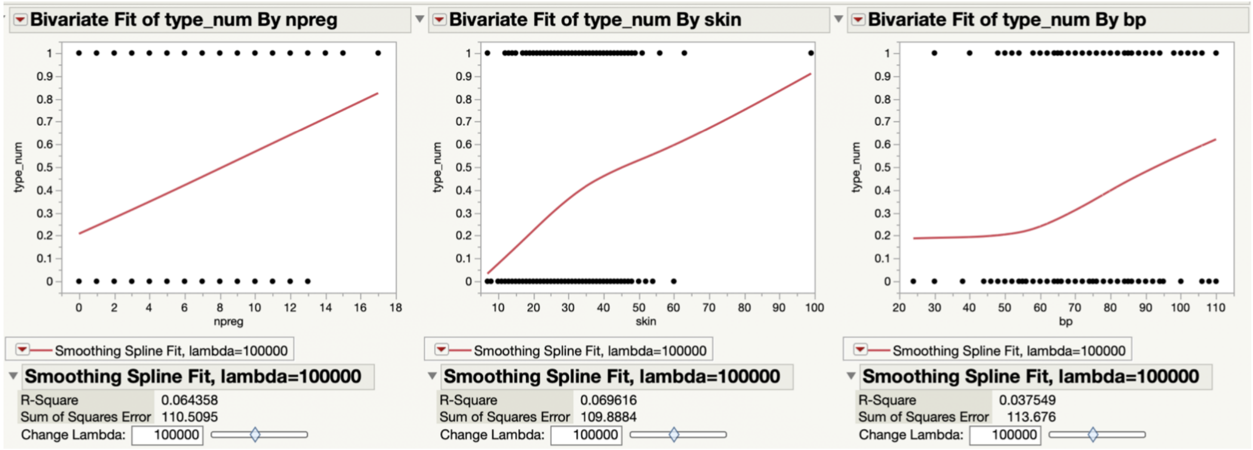
Smoothing Spline Fit, lambda=100000

R-Square 0.054324
Sum of Squares Error 111.6946
Change Lambda: 100000

Bivariate Fit of type_num By age

Smoothing Spline Fit, lambda=100000

R-Square 0.133319
Sum of Squares Error 102.3645
Change Lambda: 100000



Bivariate Fit of type_num By npreg

Smoothing Spline Fit, lambda=100000

R-Square 0.064358
Sum of Squares Error 110.5095
Change Lambda: 100000

Bivariate Fit of type_num By skin

Smoothing Spline Fit, lambda=100000

R-Square 0.069616
Sum of Squares Error 109.8884
Change Lambda: 100000

Bivariate Fit of type_num By bp

Smoothing Spline Fit, lambda=100000

R-Square 0.037549
Sum of Squares Error 113.676
Change Lambda: 100000

These plots suggest that nothing particularly wild is happening between the variables and the responses. Intuitively and based on the data it seems like the glu variable has the strongest linear relationship with having diabetes or not. Also ped seem to have a linear relationship with having diabetes. Other variables like bmi or age seem to have more complex, possibly spline or polynomially-described relationships with diabetes. Once again we see the skin outlier of 99 mm. Most don't look like they have any relationship that is sticking out to the naked eye at the moment. It would be best to get into looking at them through the models.

Model Fitting

First, I should standardized the quantitative variables before carrying out the analysis to make sure that units and differences between variable ranges are not confounding.

```
# Scale the data
Dtrain = cbind(scale(Dtrain[,1:7]),Dtrain[,8:9])
Dtest = cbind(scale(Dtest[,1:7]),Dtest[,8:9])
```

Now I will fit and compare the performances of logistic regression, lasso, elastic net, and stepwise GAM models on predicting Diabetes in the Pima dataset.

Logistic Regression

First let's try with a basic logistic regression, using stepwiseAIC to choose the best model with basic linear relationships.

```
# Fit the model
fullmodel <- glm(type_num ~npreg+skin+bmi+ped+age+glu+bp, data = Dtrain, family = binomial)
logfit <- stepAIC(fullmodel, trace = FALSE)
# Look at the final selected model
summary(logfit)
```

```
##
## Call:
## glm(formula = type_num ~ npreg + bmi + ped + age + glu, family = binomial,
##     data = Dtrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0009  -0.6816  -0.3664   0.6467   2.2898
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9562     0.1986  -4.815 1.48e-06 ***
## npreg         0.3472     0.2172   1.599  0.10989
## bmi           0.4884     0.2001   2.440  0.01468 *
## ped           0.5565     0.2031   2.740  0.00614 **
## age           0.4312     0.2301   1.874  0.06097 .
## glu           1.0073     0.2111   4.771 1.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 178.47  on 194  degrees of freedom
## AIC: 190.47
##
## Number of Fisher Scoring iterations: 5
```

```
# Make predictions
probabilities <- predict(logfit, Dtest, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)
# Model Error
```

```
cfM <- xtabs(~Dtest$type_num+predicted.classes)
error <- (cfM[1,2] + cfM[2,1])/(sum(cfM))
```

Looking at the “best” model summary, this linear equation returns the predicted log odds that a subject has diabetes. From the p-values of each of the predictors, we can see that (at α =.05) glu, ped and bmi and Number are significant with p-values ~1.83e-06, ~.006, and 0.014 respectively, while npreg is not “significant”" with p-value ~.1 . Age is borderline having a p-value ~.06. Because we standardized the variables we can more confidently interpret and compare the effects of the coefficients. We can interpret the coefficient on glu by saying that as the blood sugar level is one unit higher, the log odds of having diabetes increases by about 1. Glu has the largest effect on log odds compared to all the other predictors. Logically this makes sense wit what we already know medically. The coefficient for ped has the next largest effect with one unit increase in the diabetes pedigree function score leading to a 0.56 increase in the log odds of having diabetes.

```
actual <- Dtest[,9]
predicted <- predicted.classes

final.logistic <- data.frame(actual, predicted)
names(final.logistic) <- c("actual", "predicted")

cFM(final.logistic)
```

```
##                          Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                      66                    23
## Predicted Not Diabetic                  43                   200
```

```
## [1] 0.1988
```

```
# See Rmd for the code I wrote for the function cFM, which makes a nice confusion matrix
```

The misclassification error rate for this model is decent at about 20% for a basic cutoff value at 0.5. We should see how this rate affects other models like lasso or stepwise GAM.

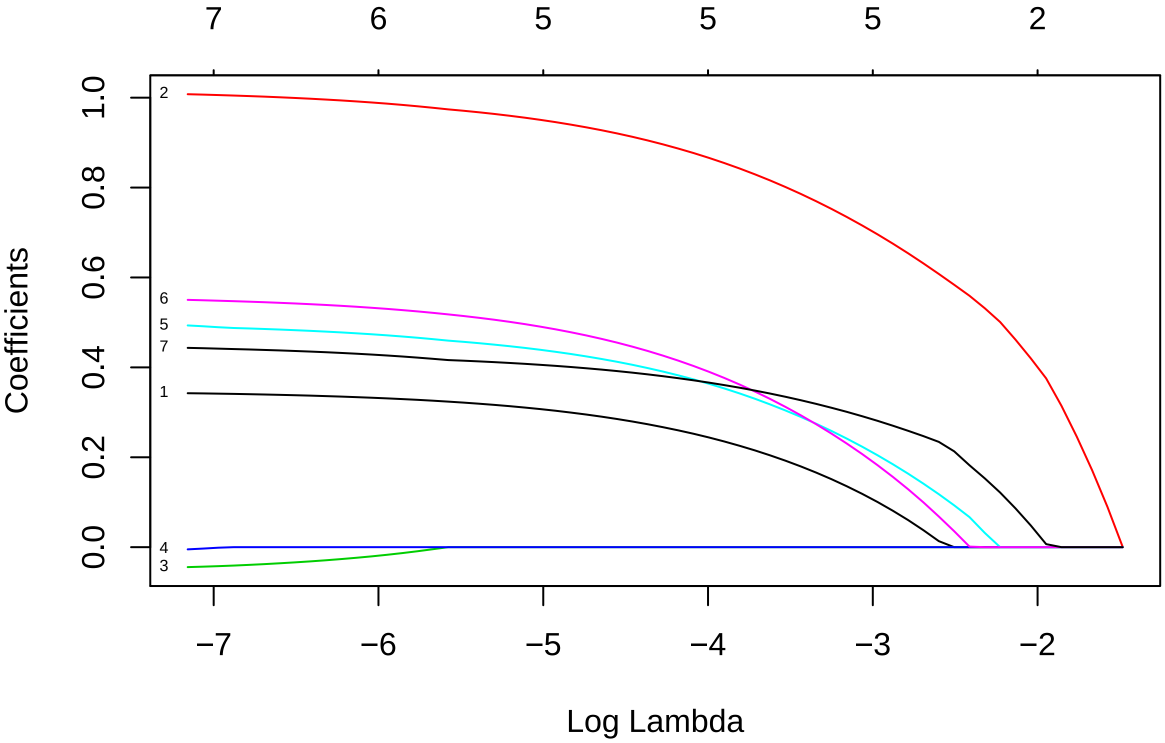
Lasso

Next we consider a lasso model.

```
set.seed(88)
x = cbind(npreg = Dtrain$npreg,glu = Dtrain$glu, bp= Dtrain$bp, skin=Dtrain$skin,
          bmi= Dtrain$bmi, ped=Dtrain$ped, age=Dtrain$age)
head(x)
```

```
##           npreg        glu         bp       skin        bmi        ped
## [1,]  0.4248028 -1.1990315 -0.2839819 -0.1036283 -0.3441969 -0.3149648
## [2,]  1.0189326  2.2430131 -0.1097599  0.3228257 -1.1761421 -0.9692087
## [3,]  0.4248028 -1.4832370  0.9355723  1.0051521  0.5693115 -0.9919933
## [4,] -1.0605217  1.2956613  0.4129062  1.1757337  2.5431421 -0.6567340
## [5,] -1.0605217 -0.5358853 -0.9808701 -0.3595007 -0.9640776 -1.0668571
## [6,]  0.4248028 -0.8516692  0.4129062 -0.1889191  0.5366862 -0.2693955
##             age
## [1,] -0.7389228
## [2,]  2.0855663
## [3,]  0.2633153
## [4,] -0.5566977
## [5,] -0.8300353
## [6,]  1.8122286
```

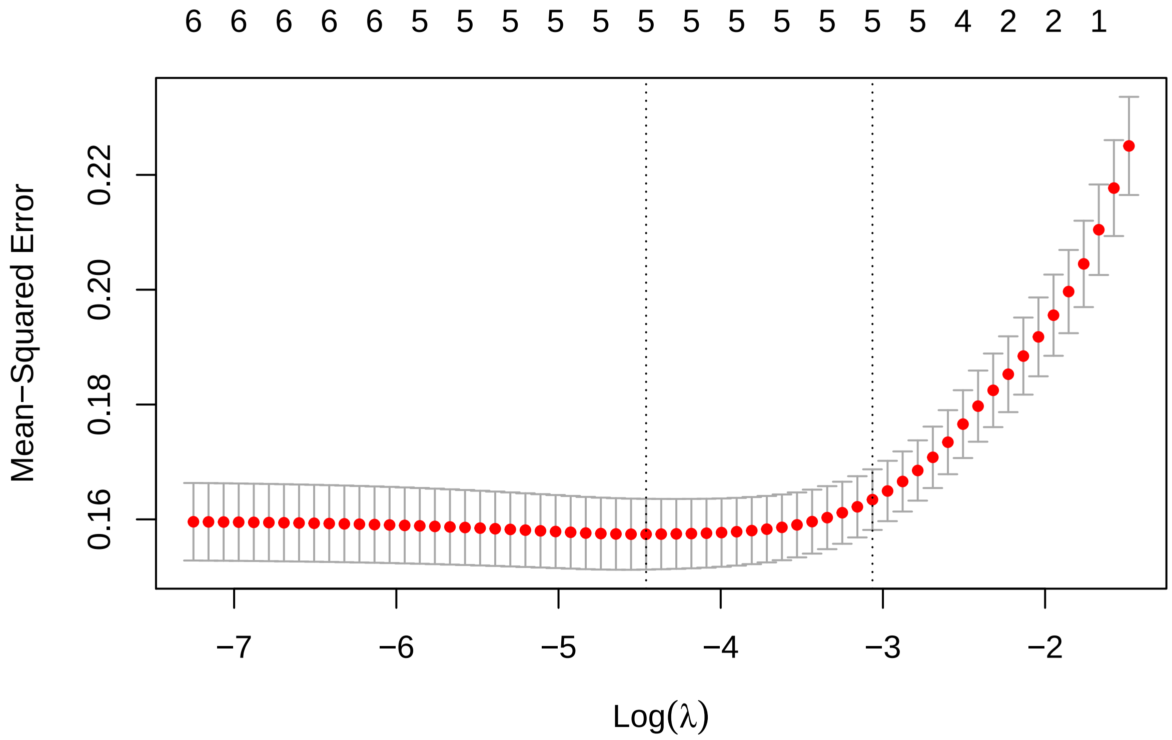
```
y = Dtrain$type_num
fit.lasso = glmnet(x,y, family = "binomial")
plot(fit.lasso,xvar="lambda", label = TRUE)
```



The plot above show us that lasso shrinks all of the coefficients towards zero as log lambda increases.

We can also look at the plot and coefficients of the mean-squared error for a range of log lambda values.

```
cv.lasso = cv.glmnet(x,y)
plot(cv.lasso)
```



```r
coef(cv.lasso)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 0.34000000
## npreg       0.02337951
## glu         0.14495844
## bp          .
## skin        .
## bmi         0.03803642
## ped         0.04207929
## age         0.06117063
```

The plot shows us that a log lambda value somewhere around -4 will give us the lowest mean-squared error of about 0.16.

The coefficients suggest that glucose levels play the largest role, though age, bmi, pedigree, and number of pregnancies are important to the model as well.

Next, we calculate the best value of lambda for our lasso model.

```r
best_lam <- cv.lasso$lambda.min
best_lam # best lambda value
```

```
## [1] 0.01156326
```

Using the best lambda value of $\lambda = 0.02$, we will compare our predictions for whether the Pima women in the test set have diabetes with their actual conditions.

```r
lasso_best <- glmnet(x, y, alpha=1, lambda = best_lam)
xvars.Dtest <- Dtest[,1:7]
yvars.Dtest <- Dtest[,9]
pred <- predict.glmnet(lasso_best, s = best_lam, newx = as.matrix(xvars.Dtest))
```

```
actual <- yvars.Dtest
predicted <- round(pred)

final.lasso <- data.frame(actual, predicted)
names(final.lasso) <- c("actual", "predicted")

cFM(final.lasso)
```

```
##                        Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                    67                    22
## Predicted Not Diabetic                42                   201
```

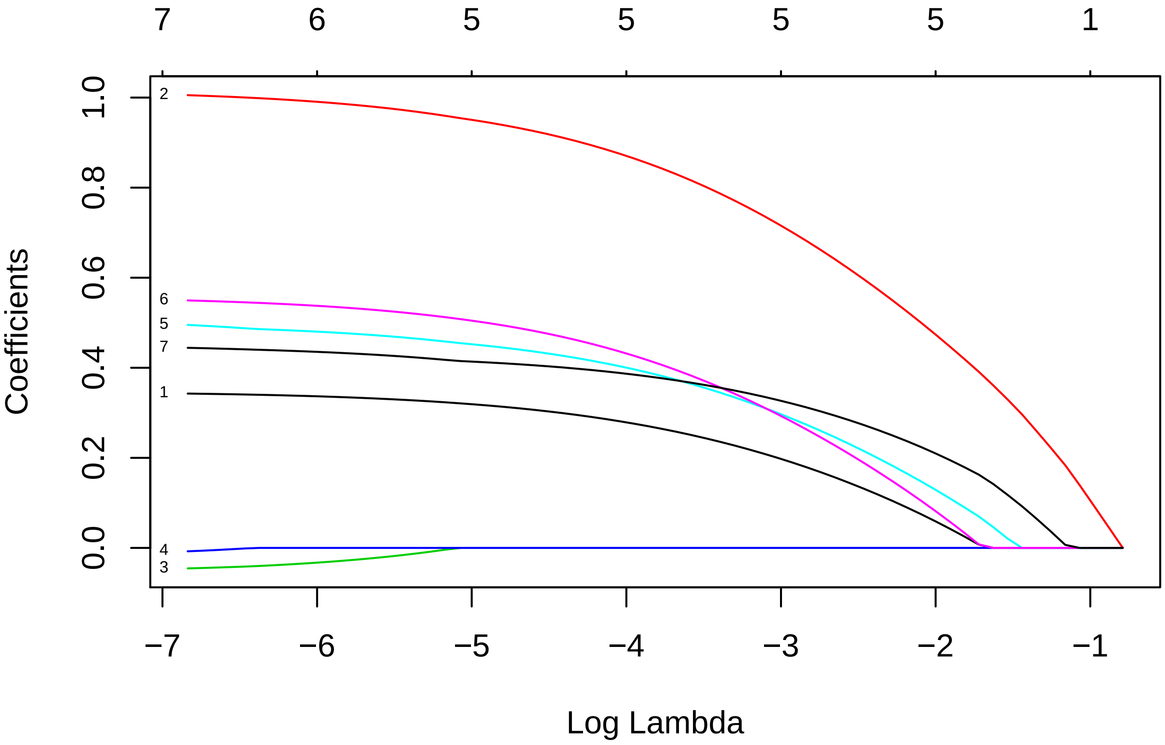
```
## [1] 0.1928
```

The confusion matrix above shows us how our lasso model performed in predicting the prevalence of diabetes among the Pima women in the test set. We get a miscalculation rate of 18.98%, which means that the model is decent at making predictions, but it is certainly not foolproof. Using this model, we should expect to make accurate diagnoses for an average of 4 in every 5 Pima women. Out of the 332 women in the sample, we would correctly tell 203 of them that they are not diabetic and 66 of them have diabetes, which sounds pretty good. However, our model would tell 43 diabetic women that they don't have diabetes, which stands out as a major concern. The fewer Type II errors our model makes, the better.

Now we will see how an elastic net model performs in comparison.

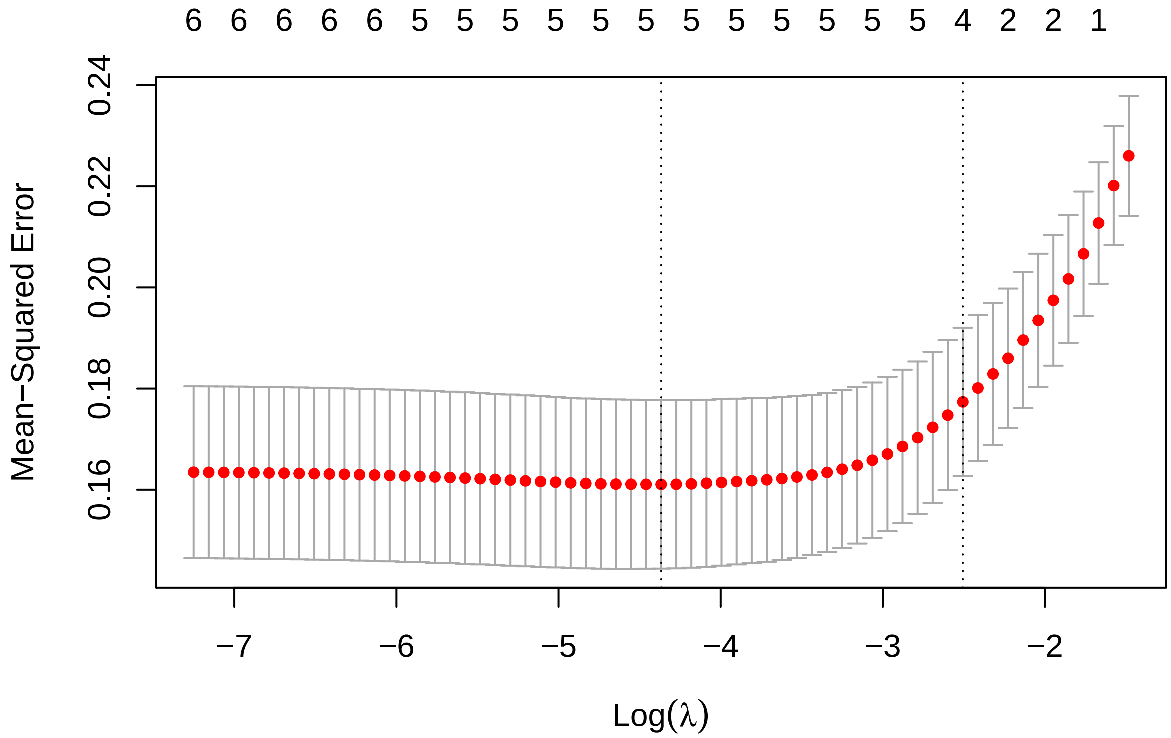
Elastic Net

```
fit.elastic= glmnet(x,y, family = "binomial", alpha = 0.5)
plot(fit.elastic,xvar="lambda", label = TRUE)
```



Similar to lasso, the elastic net model shrinks the coefficients toward zero as log lambda increases.

```
cv.elastic = cv.glmnet(x,y)
plot(cv.elastic)
```



A log lambda value of -4 or lower gets us a mean-squared error of about 0.16.

```
coef(cv.elastic)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 0.340000000
## npreg       .
## glu         0.125352895
## bp          .
## skin        .
## bmi         0.016891020
## ped         0.008531093
## age         0.047316640
```

The coefficients are comparable to those in the lasso model. Glucose levels play the biggest role, and age, bmi, pedigree, and the number of pregnancies are important as well. Higher values of all of those values are associated with a higher probability of diabetes in women from the Pima Tribe.

```
best_lam <- cv.elastic$lambda.min
best_lam
```

```
## [1] 0.01269067
```

The best lambda value is 0.0139. We will now use this to find the best elastic net model and make predictions on the test set.

```
elastic_best <- glmnet(x, y, alpha=1, lambda = best_lam)

preds.elastic <- predict.glmnet(elastic_best, s = best_lam, newx = as.matrix(xvars.Dtest))
```

```r
actual <- yvars.Dtest # same as in the other models
predicted <- round(preds.elastic) #cut-off at 0.5
# rounded predictions from the GAM model

final.elastic <- data.frame(actual, predicted)
names(final.elastic) <- c("actual", "predicted")

cFM(final.elastic)
```

```
##                         Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                     67                    21
## Predicted Not Diabetic                 42                   202
```

```
## [1] 0.1898
```

Our confusion matrix is comparable to the previous models. Our error rate is around 19%. We still make a concerning amount of Type II errors, $\frac{43}{332} = 0.1295$.

Lastly, we will look at a stepwise GAM model.

GAM

```r
# Creating a GAM Object
Gam.object <- gam(type_num~npreg+glu+ped+age+bmi+bp+skin, data=Dtrain)
step.object <- step.Gam(Gam.object, scope=list("npreg"=~1+npreg+s(npreg,2)+s(npreg,3)+s(npreg,4),
                                               "glu"=~1+glu+s(glu,2)+s(glu,3)+s(glu,4),
                                               "ped"=~1+ped+s(ped,2)+s(ped,3)+s(ped,4),
                                               "age"=~1+age+s(age,2)+s(age,3)+s(age,4),
                                               "bmi"=~1+bmi+s(bmi,2)+s(bmi,3)+s(bmi,4),
                                               "bp"=~1+bp+s(bp,2)+s(bp,3)+s(bp,4),
                                               "skin"=~1+skin+s(skin,2)+s(skin,3)+s(skin,4)))
```

```
## Start:  type_num ~ npreg + glu + ped + age + bmi + bp + skin; AIC= 202.5575
## Step:1 type_num ~ npreg + glu + ped + age + bmi + bp ; AIC= 200.561
## Step:2 type_num ~ npreg + glu + ped + age + bmi ; AIC= 198.586
## Step:3 type_num ~ npreg + glu + ped + s(age, 2) + bmi ; AIC= 197.1146
## Step:4 type_num ~ npreg + glu + ped + s(age, 2) + s(bmi, 2) ; AIC= 195.7667
## Step:5 type_num ~ glu + ped + s(age, 2) + s(bmi, 2) ; AIC= 194.6793
## Step:6 type_num ~ glu + ped + s(age, 3) + s(bmi, 2) ; AIC= 193.5195
## Step:7 type_num ~ glu + ped + s(age, 4) + s(bmi, 2) ; AIC= 193.0197
## Step:8 type_num ~ glu + ped + s(age, 4) + s(bmi, 3) ; AIC= 192.7367
```

Moving through model options stepwise, we arrive at our best option for a GAM model, which uses glucose, pedigree, a spline of age, and a spline of bmi to predict whether patients have diabetes or not.

Best GAM Model

```r
summary(step.object)
```

```
##
## Call: gam(formula = type_num ~ glu + ped + s(age, 4) + s(bmi, 3), data = Dtrain,
##     trace = FALSE)
## Deviance Residuals:
```

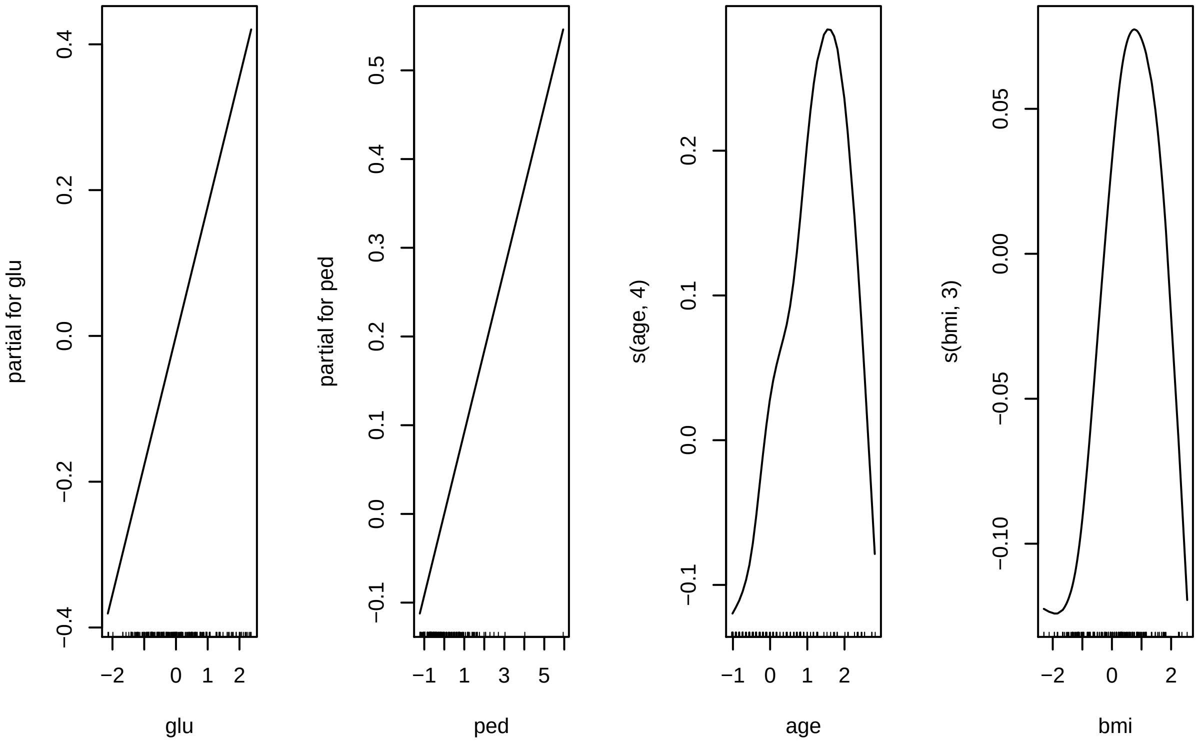
```
##      Min       1Q   Median       3Q      Max 
## -0.85278 -0.27846 -0.04304  0.25344  0.92152 
## 
## (Dispersion Parameter for gaussian family taken to be 0.1447)
## 
##     Null Deviance: 44.88 on 199 degrees of freedom
## Residual Deviance: 27.4983 on 190.0002 degrees of freedom
## AIC: 192.7367 
## 
## Number of Local Scoring Iterations: 2 
## 
## Anova for Parametric Effects
##             Df  Sum Sq Mean Sq F value    Pr(>F)    
## glu          1 10.5867 10.5867 73.1489 3.951e-15 ***
## ped          1  1.6739  1.6739 11.5659 0.0008188 ***
## s(age, 4)    1  2.0599  2.0599 14.2332 0.0002157 ***
## s(bmi, 3)    1  0.5423  0.5423  3.7467 0.0543962 .  
## Residuals  190 27.4983  0.1447                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Anova for Nonparametric Effects
##             Npar Df Npar F   Pr(F)  
## (Intercept)                         
## glu                                 
## ped                                 
## s(age, 4)         3 3.5117 0.01633 *
## s(bmi, 3)         2 2.9143 0.05667 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coefficients(step.object)
```

```
## (Intercept)         glu         ped   s(age, 4)   s(bmi, 3) 
##  0.34000000  0.17744858  0.09182162  0.10424231  0.05456620
```

Just as in the other models, the coefficients suggest that glucose levels appear to have the largest effect on the model predictions, though pedigree, age, and bmi are important as well.

```r
par(mfrow=c(1,4))
plot.Gam(step.object)
```



The plots of glucose vs the partial for glucose and pedigree vs the partial for pedigree show linear relationships. There is some clear curvature in the plots for age vs the spline of age and bmi vs the spline of bmi.

GAM Model Performance

```
xvars.Dtest <- Dtest[,1:7]
yvars.Dtest <- Dtest[,9]

preds.gam = predict(step.object, newdata=xvars.Dtest)

actual <- yvars.Dtest # same as in the other models
predicted <- round(preds.gam) #cut-off at 0.5
predicted[203] = 0
# rounded predictions from the GAM model

final.gam <- data.frame(actual, predicted)
names(final.gam) <- c("actual", "predicted")

cFM(final.gam)
```

```
##                           Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                       66                    26
## Predicted Not Diabetic                   43                   197
```

```
## [1] 0.2078
```

Based on the confusion matrix, the GAM model with a cut-off at 0.5 performs similarly to the previous models. Again, we do a pretty decent job at accurately predicting diabetes diagnoses with a misclassification rate of about 20%, but the model also predict that 43 women are not diabetic when they actually do have diabetes. I will look at how varying the cutoff criterion might help cut down on the number of Type II errors.

Model Selection and Cut-Off Criterion

Before we continue working with our GAM model, we must point out that diabetes is not an immediately life threatening illness when patients properly monitor their health. The preliminary treatment for it often involves keeping track of one's insulin levels and making some basic lifestyle changes related to diet and exercise. With this in mind we could imagine that we would care less about the Type I error rate. It is not the end of the world if we misdiagnose someone with diabetes even if they don't have it (false positives). We care more about missing people who are diabetic and we don't predict them as having it (Type II error rate). Those patients with undiagnosed diabetes are at risk of experiencing dangerous declines in overall health, so we should prefer diagnosing too many people with diabetes over underdiagnosing the diabetic population. In this way, instead of having a cutoff of 0.5 as we normally do, we could go a bit lower to get lower numbers of false negatives but still keep the error relatively low.

First let's remind ourselves of what the breakdown and error rate is for our basic cutoff at 0.5.

```
xvars.Dtest <- Dtest[,1:7]
yvars.Dtest <- Dtest[,9]

actual <-yvars.Dtest
predicted <- ifelse(preds.gam > 0.5, 1, 0)

final.gam <- data.frame(actual, predicted)
names(final.gam) <- c("actual", "predicted")

cFM(final.gam)
```

```
##                        Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                    66                    26
## Predicted Not Diabetic                43                   197
```

```
## [1] 0.2078
```

We have more false negatives (43) than false positives (26) and the error rate is about 0.2.

Let's go slightly lower with our cutoff to 0.4.

```
##                        Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                    80                    42
## Predicted Not Diabetic                29                   181
```

```
## [1] 0.2139
```

We have slighty fewer false negatives (29) than false positives (42) and the error rate is only slightly higher at about 0.214. This is closer to what we want, but we will look at what happens if we push the cutoff a little further.

Here is what happens when we go much lower to a cutoff of 0.2:

```
##                        Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                    99                   104
## Predicted Not Diabetic                10                   119
```

```
## [1] 0.3434
```

We have much less false negatives (10) than false positives (104) but now the error rate is slightly higher at about 0.34. This is clearly too big of a jump and is working against the predictive power of the model. A cuttoff somwhere in between 0.4 and 0.2 looks like it will be best for our goals.

Here is our final change with a cutoff value of 0.32.

```
##                         Actually Diabetic Actually Not Diabetic
## Predicted Diabetic                     90                    60
## Predicted Not Diabetic                 19                   163
```

```
## [1] 0.238
```

We have half as many false negatives (19) as false positives (60) and now the error rate is only slightly higher at about 0.24. This seems like the best cutoff for what we want to be predicting with our model for diabetes. We manage to decrease the number of false negatives without sacrificing much in the way of our overall error rate.

Summary

The best model we have decided on for prediction is the Stepwise GAM model with glu, ped, age and bmi as the relevant variables for prediction and a cut off value of 0.32. With an error rate of about 0.24 this model does a pretty good job at predicting diabetes in Pima Tribe women. With the cut off of 0.32 there is much lower Type II error as compared to Type I error. So while keeping the misclassification rate low, this model will err towards the side of caution, preferring to tell people they are diabetic even if they are not as opposed to telling patients they are not diabetic when they truly are.

Regardless of our predicting power, this model has shown that the medical connection of blood sugar (glu), BMI, and family history (ped) as determinants of diabetes seems to have held true in our model. Blood sugar levels seem to have the largest effect on having diabetes or not with all the models that were tried, which also goes along with the current literature.

#Sources

https://www.cdc.gov/pregnancy/diabetes-gestational.html https://www.webmd.com/diabetes/guide/normal-blood-sugar-levels-chart-adults https://www.healthline.com/health/high-blood-pressure-hypertension/blood-pressure-reading-explained