Project 2

Neil Klenk (nlk322)

Instructions

Please submit both this completed Rmarkdown document and its knitted HTML, converted to PDF, on Canvas **no later than 11:59 pm on March 29th, 2015**. These two documents will be graded jointly, so they must be consistent (as in, don't change the Rmarkdown file without also updating the knitted HTML!).

All results presented **must** have corresponding code. Any answers/results given without the corresponding R code that generated the result will be considered absent. All code reported in your final project document should work properly. Please bear in mind that **you will lose points** for the following:

- an R-code chunk with no comments
- results without corresponding R code
- extraneous code which does not contribute to the question

For this project, you will work with a dataset collected from Pima Native American women. Studies have shown that Pima women have a much higher incidence of Type II Diabetes than the general population. Since the 1960s, NIH researchers have periodically asked Pima women to undergo various medical tests in order to assess possible diabetes risk factors. Consequently, data on Pima women has proven useful for predicting how likely an individual is to develop diabetes. [Source: J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Symposium on Computer Applications in Medical Care, 261â“265.] Your goal for this project is to analyze the Pima women dataset using several statistical approaches we have learned, in two parts.

We have already subdivided the full data set into training and test data sets (`pima_training` and `pima_test`). And we also provide the full data set (`pima_full`). Please use the training and test data sets for Part 1 and either data set for Part 2.

```
##### Use these datasets for part 1 (described below) #####
# Dataset to use specifically for training in Part 1
pima_training <- read.csv("http://wilkelab.org/classes/SDS348/2016_spring/proje
cts/project2/pima_training.csv")
# Dataset to use specifically for testing your model in Part 1
pima_test <- read.csv("http://wilkelab.org/classes/SDS348/2016_spring/projects/
project2/pima_test.csv")

##### Use this dataset for part 2 (described below) #####
# Complete Pima data, with a single observation per individual
pima_full <- read.csv("http://wilkelab.org/classes/SDS348/2016_spring/projects/
project2/pima.csv")

head(pima_full)
```

```
##   npreg glucose dbp skin insulin  bmi pedigree age diabetic
## 1     6     148  72   35       0 33.6    0.627  50      Yes
## 2     1      85  66   29       0 26.6    0.351  31       No
## 3     8     183  64    0       0 23.3    0.672  32      Yes
## 4     1      89  66   23      94 28.1    0.167  21       No
## 5     0     137  40   35     168 43.1    2.288  33      Yes
## 6     5     116  74    0       0 25.6    0.201  30       No
```

The column contents are as follows:

- **npreg**: number of times pregnant
- **glucose**: plasma glucose concentration at 2 hours in an oral glucose tolerance test (units: mg/dL)
- **dpb**: diastolic blood pressure (units: mm Hg)
- **skin**: triceps skin-fold thickness (units: mm)
- **insulin**: 2-hour serum insulin level (units: μ U/mL)
- **bmi**: Body Mass Index
- **age**: age in years
- **diabetic**: whether or not the individual has diabetes

Part 1 (40 points). We have divided the dataset, which consists of observations from 768 individuals, into a training and a test data set. Fit a logistic regression model (to predict diabetes incidence) on the training data set. When building your model, use backwards selection to choose predictors which are significant **at your chosen significance level (be sure to report your chosen value!)**. Your code should be appropriately commented with high-level statements about the code's function.

Using your final model, predict the outcome on the test data set, and plot and discuss your results. You should have two final plots: a plot with two ROC curves for the training and test data each, and a plot of the fitted probability of diabetes incidence as a function of the predictors, colored by diabetes, on the test data. Your discussion should, at least, cover the differences and similarities in model performance on the training vs. test data (including AUC) as well as a clear interpretation of each plot. Please limit your discussion to a maximum of 8 sentences.

Part 2 (60 points). Think of two **scientific** questions to ask about this data set (for this, you are welcome to use either the training, test, or full data set). Scientific questions should not be procedural, they should be **conceptual**. (For example, “What is the distribution of ages?” is a **procedural** question because all it asks you to do is plot a distribution, but, “Are incidence of diabetes higher in older women or in younger women?” is a **conceptual** question because you have to determine which type of plot is appropriate for the question and interperet that plot.) For each question, perform an exploratory statistical analysis (PCA, k-means, logistic regression, linear model, etc.) with a corresponding figure. Discuss your findings, in particular how your analysis’ results reveal (or don’t reveal) an answer your proposed question. Please limit each question’s discussion to a maximum of 5 sentences.

Project responses should be entered below. model.name=NULL) { outcome <- as.numeric(factor (known_truth))-1 pos <- sum(outcome) # total known positives neg <- sum(1-outcome) # total known negatives pos_probs <- outcome*probabilities # probabilities for known positives

---

```
# This R code chunk contains the calc_ROC function.
calc_ROC <- function(probabilities, known_truth, model.name=NULL)
  {
  outcome <- as.numeric(factor(known_truth))-1
  pos <- sum(outcome) # total known positives
  neg <- sum(1-outcome) # total known negatives
  pos_probs <- outcome*probabilities # probabilities for known positives
  neg_probs <- (1-outcome)*probabilities # probabilities for known negatives
  true_pos <- sapply(probabilities,
                     function(x) sum(pos_probs>=x)/pos) # true pos. rate
  false_pos <- sapply(probabilities,
                     function(x) sum(neg_probs>=x)/neg)
  if (is.null(model.name))
    result <- data.frame(true_pos, false_pos)
  else
    result <- data.frame(true_pos, false_pos, model.name)
  result %>% arrange(false_pos, true_pos)
  }
```

Part 1

```
#logistic regression using all avaliable predictors
glm.out.training <- glm(diabetic ~ npreg + glucose + dbp + skin + insulin + bm
i + age, data = pima_training, family = binomial)
summary(glm.out.training)
```

```
##
## Call:
## glm(formula = diabetic ~ npreg + glucose + dbp + skin + insulin +
##     bmi + age, family = binomial, data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3865  -0.7106  -0.3792   0.6812   2.3933
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.329441   0.984276  -9.478  < 2e-16 ***
## npreg        0.050062   0.040698   1.230   0.2187
## glucose      0.039411   0.004706   8.374  < 2e-16 ***
## dbp         -0.016633   0.007225  -2.302   0.0213 *
## skin        -0.006479   0.008700  -0.745   0.4565
## insulin     -0.001311   0.001109  -1.182   0.2374
## bmi          0.121072   0.021516   5.627 1.83e-08 ***
## age          0.027301   0.011879   2.298   0.0216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 451.18  on 480  degrees of freedom
## AIC: 467.18
##
## Number of Fisher Scoring iterations: 5
```

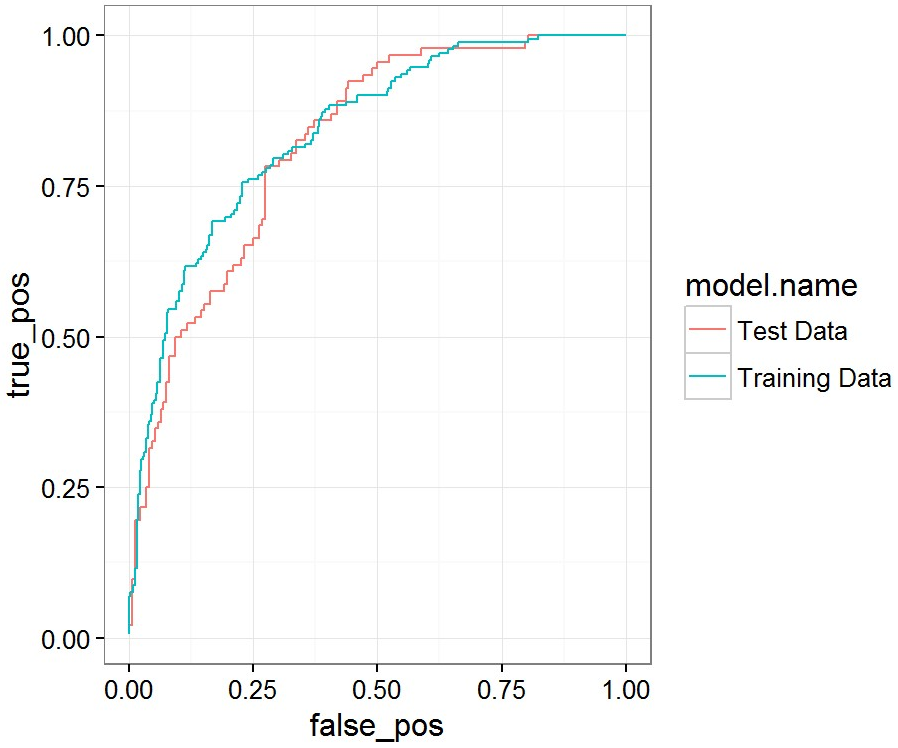
```
#logistic regression after the worst performing predictor was removed
glm.out.training <- glm(diabetic ~ glucose + dbp + skin + insulin + bmi + age,
data = pima_training, family = binomial)
summary(glm.out.training)
```

```
##
## Call:
## glm(formula = diabetic ~ glucose + dbp + skin + insulin + bmi +
##     age, family = binomial, data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3835  -0.7147  -0.3860   0.6902   2.3935
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.387367   0.985437  -9.526  < 2e-16 ***
## glucose      0.039178   0.004708   8.321  < 2e-16 ***
## dbp         -0.015994   0.007196  -2.223 0.026248 *
## skin        -0.006543   0.008752  -0.748 0.454734
## insulin     -0.001365   0.001122  -1.217 0.223542
## bmi          0.120976   0.021475   5.633 1.77e-08 ***
## age          0.034796   0.010248   3.395 0.000685 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 452.71  on 481  degrees of freedom
## AIC: 466.71
##
## Number of Fisher Scoring iterations: 5
```

```
#logistic regression where all predictors have a p-value < 0.05
glm.out.training <- glm(diabetic ~ glucose + dbp + bmi + age, data = pima_train
ing, family = binomial)
summary(glm.out.training)
```

```
##
## Call:
## glm(formula = diabetic ~ glucose + dbp + bmi + age, family = binomial,
##     data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2073  -0.7260  -0.3905   0.6954   2.3531
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.089916   0.954222  -9.526  < 2e-16 ***
## glucose      0.037198   0.004339   8.574  < 2e-16 ***
## dbp         -0.016453   0.007036  -2.338 0.019362 *
## bmi          0.108550   0.019422   5.589 2.28e-08 ***
## age          0.038776   0.010060   3.855 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 633.38  on 487  degrees of freedom
## Residual deviance: 456.10  on 483  degrees of freedom
## AIC: 466.1
##
## Number of Fisher Scoring iterations: 5
```

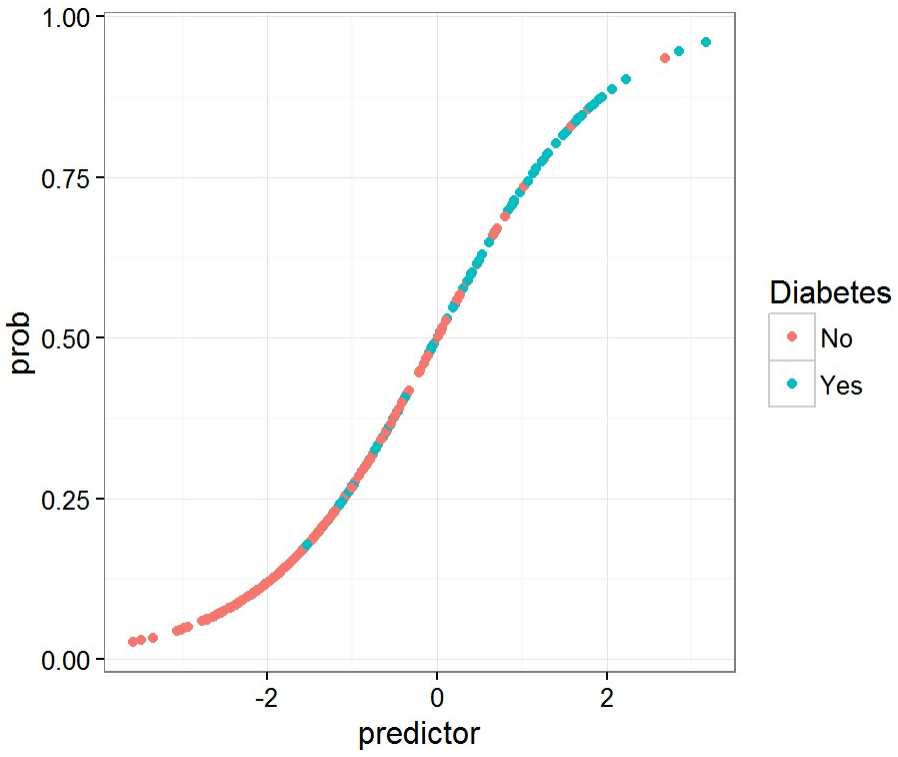
```
#Created a logistic regression model for the data in pima_test using the predic
tor variables from the model that was created using the training data
test_pred <- predict(glm.out.training, pima_test, type = 'response')

#Created a ROC model for the pima_test data to see how well the model perfomed
ROC_test <- calc_ROC(probabilities = test_pred, known_truth = pima_test$diabeti
c, model.name = "Test Data")

#Created a ROC model for the pima_training data to see how the model performed
ROC_training <- calc_ROC(probabilities = glm.out.training$fitted.values, known_
truth = pima_training$diabetic, model.name = "Training Data")

#Plotting the two ROC curves on the same graph for easy comparative analysis
ggplot(data = NULL, aes(x = false_pos, y = true_pos)) + geom_line(data = ROC_te
st, aes(color = model.name)) + geom_line(data = ROC_training, aes(color = mode
l.name))
```




```
#combining the two different ROCs
ROCs <- rbind(ROC_test, ROC_training)

#determing the area under the curve (AUC) for each of the ROCs
ROCs %>% group_by(model.name) %>%
  mutate(delta=false_pos-lag(false_pos)) %>%
  summarize(AUC=sum(delta*true_pos, na.rm=T)) %>%
  arrange(desc(AUC))
```

```
## Source: local data frame [2 x 2]
##
##      model.name       AUC
##           (fctr)     (dbl)
## 1 Training Data 0.8376877
## 2     Test Data 0.8197042
```

```
#Created a new dataset for use while making the final plot
glm.out.test <- glm(diabetic ~ glucose + dbp + bmi + age, data = pima_test, fam
ily = binomial)

#Created a new data frame consisting of the fitted probability of getting diabe
tes, as a function of the predictors.
lr_data <- data.frame(predictor=glm.out.test$linear.predictors, prob=glm.out.te
st$fitted.values, Diabetes=pima_test$diabetic)

#Plotted the itted probability of getting diabetes, as a function of the predic
tors.
ggplot(lr_data, aes(x=predictor, y=prob, color=Diabetes)) + geom_point()
```



Discussion for part 1 goes here.

With a singificance P-value defined as 0.5, glucose, dbp, bmi, and age were all found to be significan predictors of diabetes.

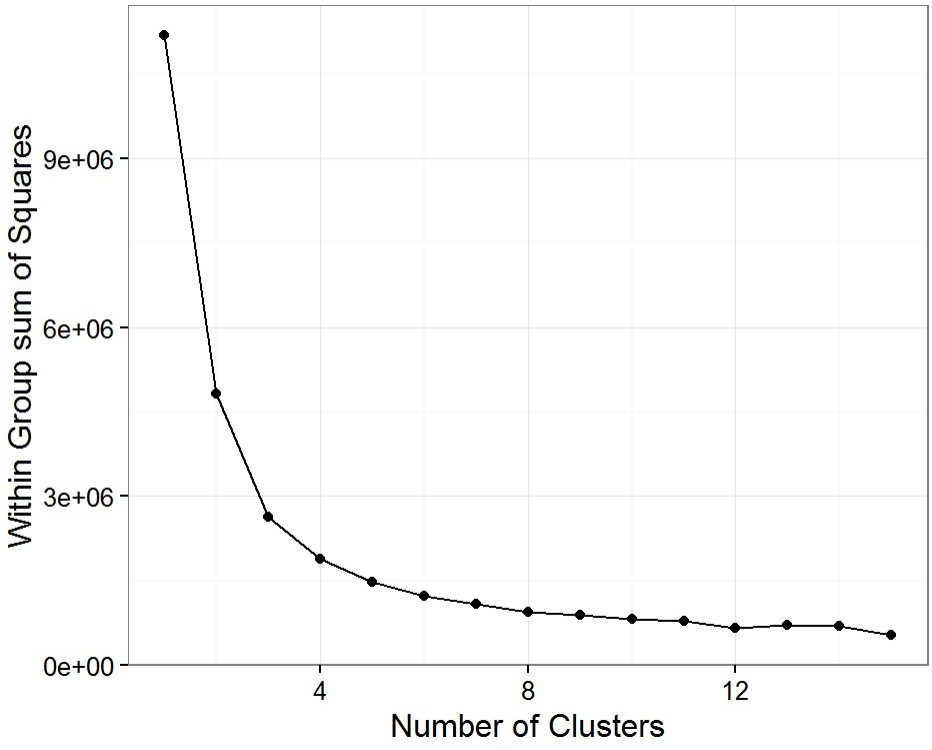
It can be seen that both models performed much better than a random guess when using the glm that was prepared using the training data. The training data performs better than the test data, and this can be seen through the area under the curve (AUC) values, 0.838 and 0.819 respectively. This was expected as a predictor always works best on the data set on which it was trained. Both of those values are much higher than the value of 0.5 that would be obtained if the model was completly ineffective.

The logistic regression model is of the fitted probability as a function of the predictors shows a clear sepparation beween those with and without diabetes. This clear seppartaion indicates that the multiple explanatory variables chosen strongly determine if the individual will have daibetes or not.

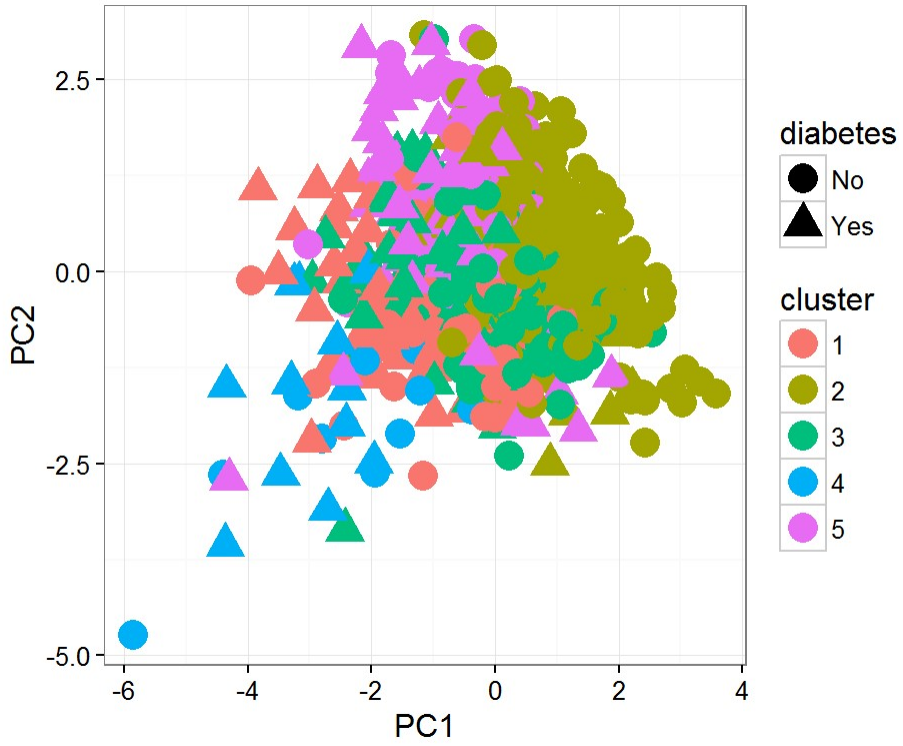
Part 2

The Pima is a very particular group of peolple. Using the provided infomration, make a reasonlable deduction to the degree of genetic homology present.

```
#Determining the best number of clusters to use by plotting the within group su
m of squares against the number of clusters used.
pima_full_numeric_test <- pima_full %>% select(-diabetic, -npreg, -skin)

#Calculating the within group sum of squares (wss) for each of potential cluste
r numbers (2 - 15)
wss <- (nrow(pima_full_numeric_test)-1)*sum(apply(pima_full_numeric_test,2,va
r))
for (i in 2:15) wss[i] <- sum(kmeans(pima_full_numeric_test, nstart=10, centers
=i)$withinss)

#Creating a new data frame composed of the wss score for each number of potenti
al clusters
wss_data <- data.frame(centers=1:15, wss)

#Plotting the wss against the number of clusters
ggplot(wss_data, aes(x=centers, y=wss)) + geom_point() + geom_line() + xlab("Nu
mber of Clusters") + ylab("Within Group sum of Squares")
```



```
#Stating that there are 5 clusters in this dataset
pima_full_numeric_test %>% kmeans(centers = 5) -> km

#Creating a dataframe consisting of the pima_full data set with the custer colu
mn added on
pima_clustered <- data.frame(pima_full, cluster = factor(km$cluster))

#Performing a PCA analysis to make the data more interpretable
pca <- pima_full_numeric_test %>% scale() %>% prcomp()

#Creating a new dataframe with teh results from the PCA analysis, the cluster
s, and if the subject was diabetic or not
cluster_data <- data.frame(pca$x, cluster = factor(km$cluster), diabetes = pima
_full$diabetic)

#Plotting the PCA analysis, indicating diabetic status, and cluster number
ggplot(cluster_data, aes(x=PC1, y=PC2, color = cluster, shape = diabetes)) + ge
om_point(size = 5)
```

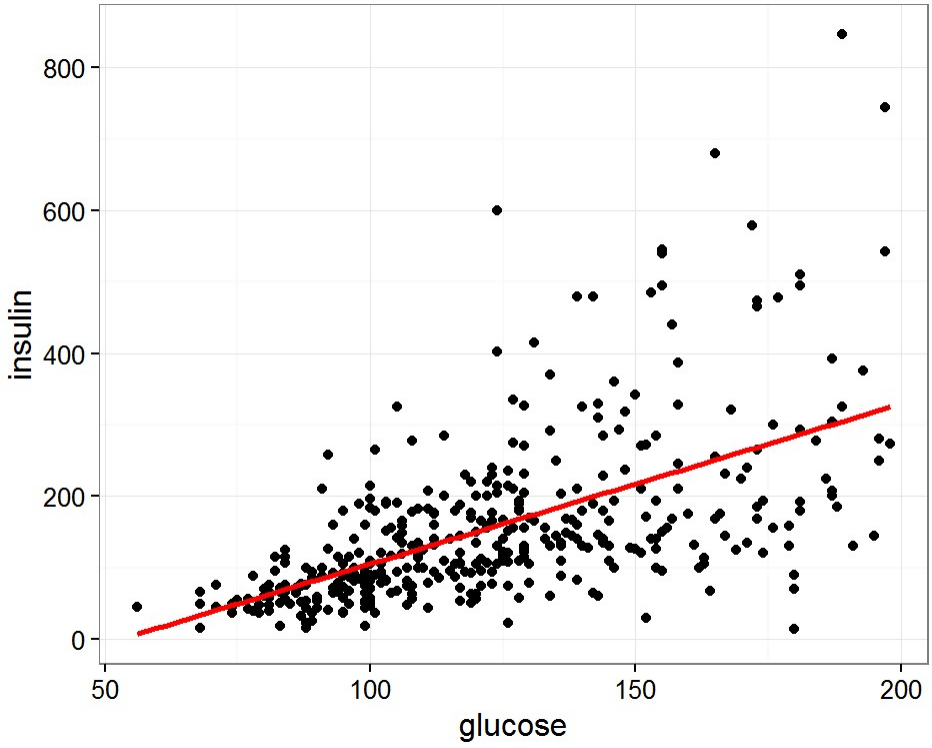


The data was clustered at the point at which the within group sum of squares began to decrease at a nonsignificant amount, resulting in up to 5 different groups being potentially detected based off of the information present. K-means clustering was chosen because it would make it easiest to identify any outlying groups, as would be expected if there was an outside set of genes in the pool (indicating a differeing genetic makeup). There was no clearly defined cluster outside of the rest, which could potentiall be explained through generations of genetic assimiation into the group. This indicates that

there is a high degree of genetic homology between the people in this data set. There is some spreading with cluster 5, so any future research into this question should begin with those individuals, partitcularly the one outlier at the far bottom left of the plot.

Is there a relationship between plasma glucose concentration at 2 hours in an oral glucose tolerance test and 2-hour serum insulin level in Pima woment?

```
#Removing all entries where the vlaue for insulin was 0
pima_insulin <- pima_full %>% filter(pima_full$insulin != 0)

#Creating the scatterplot with linear model's line running though it
pima_insulin %>% ggplot(aes(x = glucose, y=insulin)) + geom_point() + geom_smoo
th(method = "lm", se=FALSE, col = "red")
```




```
#Calculating the linear model
fit <- lm(insulin ~ glucose, data = pima_insulin)

#Printing out the r-squared value
summary(fit)$r.squared
```

```
## [1] 0.3378202
```

There is a definite positive relationship between the plasma glucose concentration and serum insulin level after the two hour test was performed on these Pima women. with an r-squared value of 0.338, it has been determined that 33.8% of the variance in insulin levels is due to the variance in plasma glucose levels. A scatter plot with an embeded linear model was chosen to answer this question. The scatterplot allows the reader to easily understand the overall trend of the relationship between these two variables while the linear model's line gives the reader a direct line of best fit that is easier to intrepret.