Pima Analysis

Evan Yacek ety78

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.]

```
# import training data set which we will build glm off
pima_training <- read.csv("http://wilkelab.org/classes/SDS348/2016_spring/projects/project2/pima_trai
ning.csv")
# Dataset which we will implement our model on
pima_test <- read.csv("http://wilkelab.org/classes/SDS348/2016_spring/projects/project2/pima_test.cs
v")



# 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: μU/mL)
- **bmi**: Body Mass Index
- **age**: age in years
- **diabetic**: whether or not the individual has diabetes

```
# 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

Significance level(Anything over will not accept glm) = 0.05

```
#Fitting logistic regression model to training data)
glm.out.complete <- glm(diabetic ~ bmi + age + insulin + skin + dbp + glucose + npreg, data=pima_trai
ning, family="binomial")
summary(glm.out.complete)
```

```
##
## Call:
## glm(formula = diabetic ~ bmi + age + insulin + skin + dbp + glucose +
##     npreg, 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 ***
## bmi          0.121072   0.021516   5.627 1.83e-08 ***
## age          0.027301   0.011879   2.298   0.0216 *
## insulin     -0.001311   0.001109  -1.182   0.2374
## skin        -0.006479   0.008700  -0.745   0.4565
## dbp         -0.016633   0.007225  -2.302   0.0213 *
## glucose      0.039411   0.004706   8.374  < 2e-16 ***
## npreg        0.050062   0.040698   1.230   0.2187
## ---
## 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
```

```
#Take out skin due to Pr > 0.05

glm.out <- glm(diabetic ~ bmi + age + insulin +  dbp + glucose + npreg, data=pima_training, family="b
inomial")
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ bmi + age + insulin + dbp + glucose +
##     npreg, family = "binomial", data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3367  -0.7170  -0.3837   0.6789   2.3820
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.310619   0.982049  -9.481  < 2e-16 ***
## bmi          0.115870   0.020225   5.729 1.01e-08 ***
## age          0.027934   0.011869   2.353   0.0186 *
## insulin     -0.001655   0.001002  -1.653   0.0984 .
## dbp         -0.017212   0.007171  -2.400   0.0164 *
## glucose      0.039915   0.004679   8.530  < 2e-16 ***
## npreg        0.050346   0.040861   1.232   0.2179
## ---
## 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: 451.74  on 481  degrees of freedom
## AIC: 465.74
##
## Number of Fisher Scoring iterations: 5
```

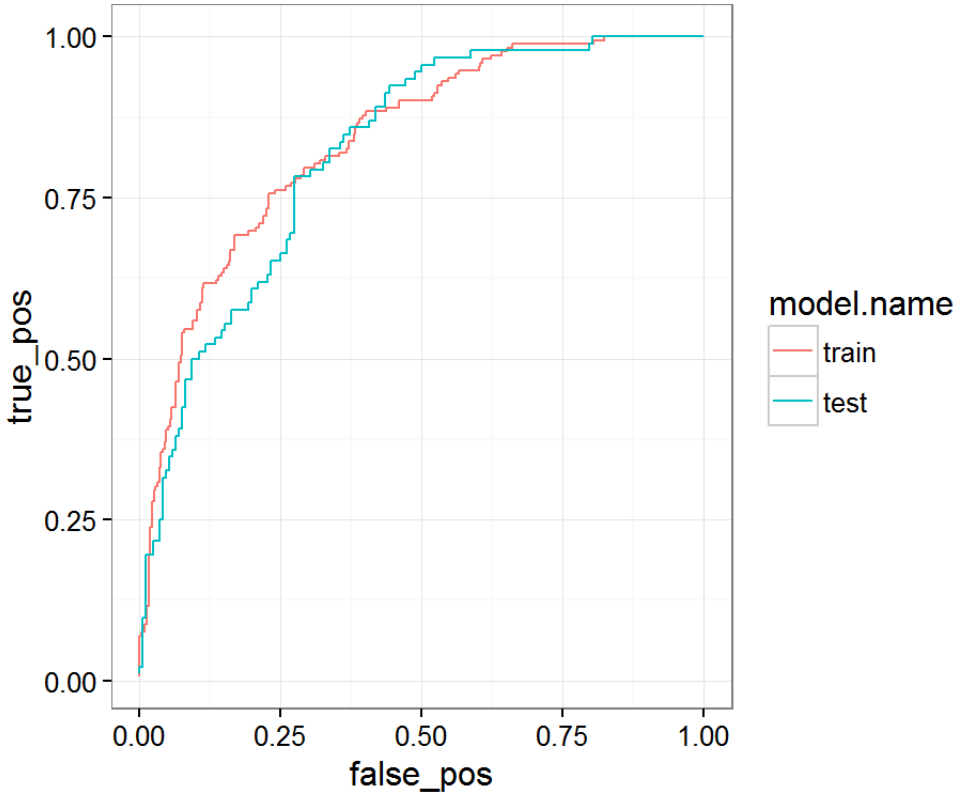
```
#Take out npreg due to Pr > 0.05
glm.out <- glm(diabetic ~ bmi + age + insulin +  dbp + glucose , data=pima_training, family="binomia
l")
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ bmi + age + insulin + dbp + glucose,
##     family = "binomial", data = pima_training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3336  -0.7203  -0.3853   0.7041   2.3744
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.369987   0.983191  -9.530  < 2e-16 ***
## bmi          0.115692   0.020150   5.741 9.39e-09 ***
## age          0.035466   0.010225   3.469 0.000523 ***
## insulin     -0.001721   0.001008  -1.706 0.087922 .
## dbp         -0.016542   0.007149  -2.314 0.020667 *
## glucose      0.039693   0.004679   8.483  < 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: 633.38  on 487  degrees of freedom
## Residual deviance: 453.26  on 482  degrees of freedom
## AIC: 465.26
##
## Number of Fisher Scoring iterations: 5
```

```
#Take out insulin due to Pr > 0.05
glm.out <- glm(diabetic ~ bmi + age +   dbp + glucose , data=pima_training, family="binomial")
summary(glm.out)
```

```
##
## Call:
## glm(formula = diabetic ~ bmi + age + dbp + glucose, 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 ***
## bmi          0.108550   0.019422   5.589 2.28e-08 ***
## age          0.038776   0.010060   3.855 0.000116 ***
## dbp         -0.016453   0.007036  -2.338 0.019362 *
## glucose      0.037198   0.004339   8.574  < 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: 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
```

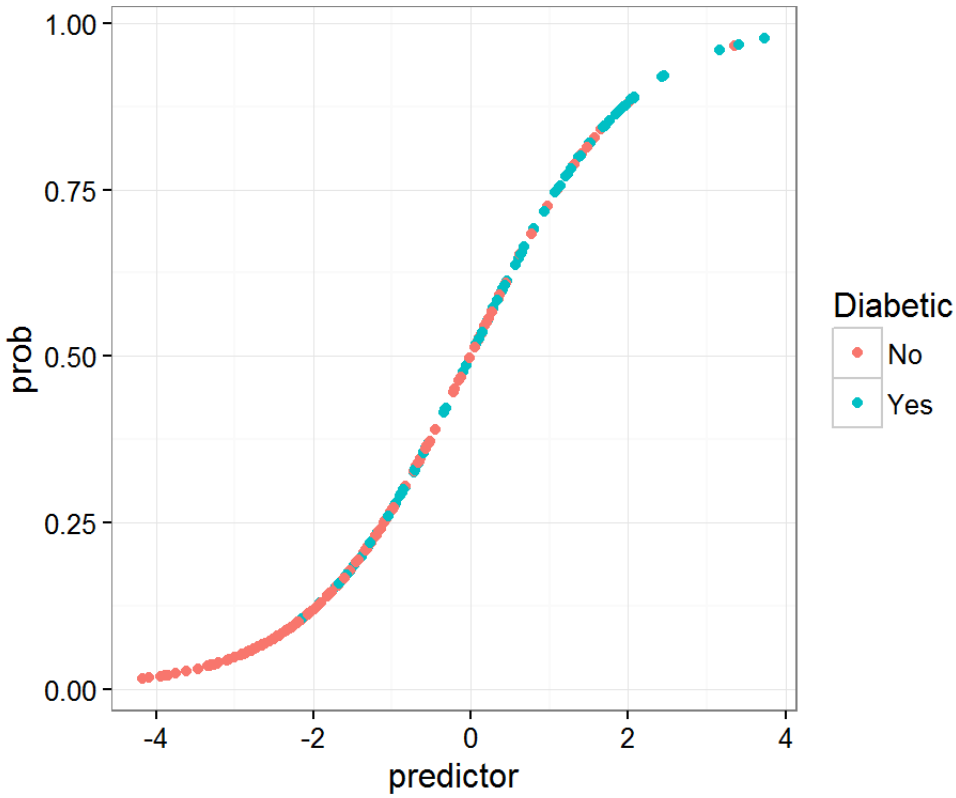
```
#Predicts fitted values on test data set
test_pred <- predict(glm.out, pima_test, type='response')

#Predicts linear predictors on test data set
test_pred2 <- predict(glm.out, pima_test)

#Create traing ROC curve
ROC.train <- calc_ROC(probabilities=glm.out$fitted.values,
                known_truth=pima_training$diabetic,
                model.name="train")
#Create test ROC curve
ROC.test <- calc_ROC(probabilities=test_pred,
                known_truth=pima_test$diabetic,
                model.name="test")
#Combine curves to one dataframe
ROCs <- rbind(ROC.train, ROC.test)

#Plot ROC Curves
ggplot(ROCs, aes(x = false_pos, y = true_pos, color = model.name))+ geom_line()
```




```
#Calculate area under the curves for 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      train 0.8376877
## 2       test 0.8197042
```

```
#Create new dataframe to plot  the fitted probability of diabetes incidence as a function of the pred
ictors

lr_data <- data.frame(predictor=test_pred2, prob= test_pred, Diabetic=pima_test$diabetic)
ggplot(lr_data, aes(x=predictor, y=prob, color=Diabetic)) + geom_point()
```




```
pred_data <- data.frame(probability=test_pred, Diabetic=pima_test$diabetic)

# cutoff of 0.5
cutoff <- 0.5

# Number of true non diabetics samples identified as non diabetics (true positives)
pred_data %>% filter(probability <= cutoff & Diabetic=="No") %>%
  tally() -> nond_true

# Number of true diabetics identified as diabetics 2 (true negatives)
pred_data %>% filter(probability > cutoff & Diabetic=="Yes") %>%
  tally() -> d_true

# Total number of true diabetics(known postives)
pred_data %>% filter(Diabetic=="No") %>%
  tally() -> nond_total
# Total number of true non diabetics (known negatives)
pred_data %>% filter(Diabetic=="Yes") %>%
  tally() -> d_total

# True positive rate
tp <- nond_true$n/(nond_total$n)
# True negative rate
tn <- d_true$n/(d_total$n)
tp
```

```
## [1] 0.8430233
```

```
tn
```

```
## [1] 0.5543478
```

After constructing a linear model and creating ROC curves we can see how the model created acts on training vs test data sets. At first the model performs better on training model, however it fluctuates multiple times bewtween performing better on the test model and performing better on the training model at a false positive rate between 0.25 and 0.5. Ultimately, it seems the model performed better on the training data set, and is justified by the AUC values of 0.84 for training data and 0.82 for the test data. . With a probability cut-off of 0.5, which I decided based on the fitted probability graph,I calculated the true positive rate to be 84.3%, and the false positive rate to be 55.4%. Thus our regression model based on the predictors bmi, age, dbp, and glucose correctly classifies 84.3% of the non diabetics. The model incorrectly classifies 55.4% of diabetics as non diabetics, both of which coincide with the ROC curves.

What are the distinguishing factors bewteen non diabetic and diabetic Pima women?

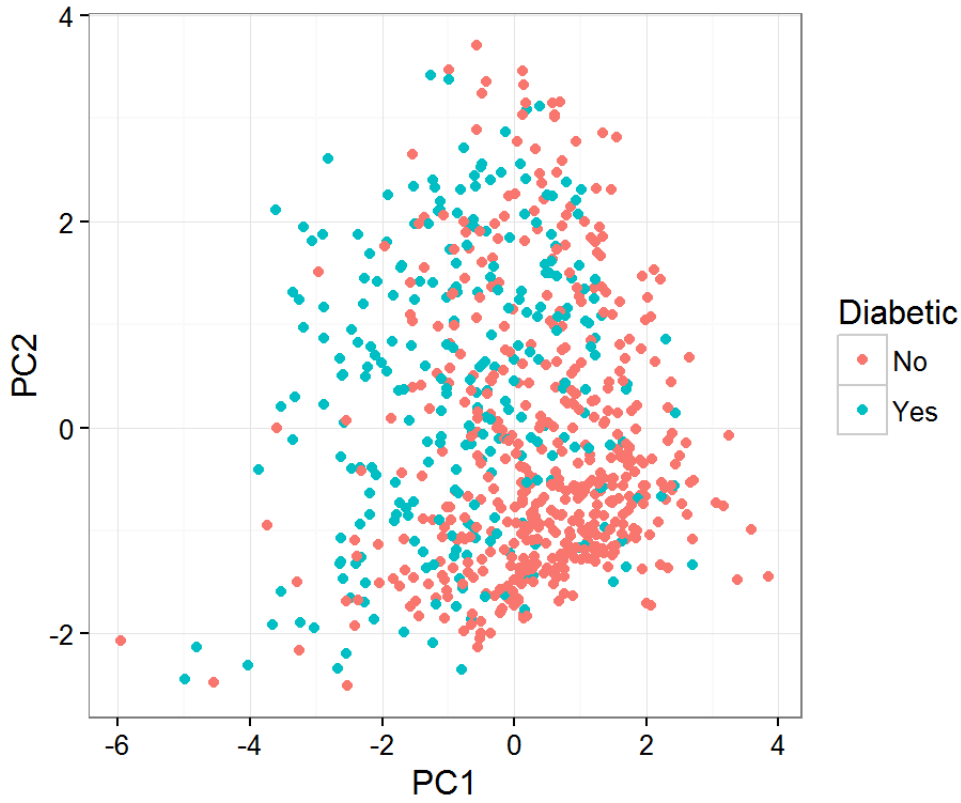
```
pima_full %>% select(-diabetic) %>% # remove diabetic column
  scale() %>%                 # scale to 0 mean and unit variance
  prcomp() ->                 # do PCA
  pca                         # store result as `pca`

# now display the results from the PCA analysis
pca
```

```
## Standard deviations:
## [1] 1.4286236 1.3187099 1.0020977 0.9375989 0.8715639 0.8655845 0.6513624
## [8] 0.6353064
##
## Rotation:
##                  PC1        PC2         PC3         PC4          PC5
## npreg    -0.1228742  0.5951911 -0.02071036  0.10513489 -0.322556825
## glucose  -0.4096361  0.1770538  0.43505735 -0.43195394  0.439742705
## dbp      -0.3100062  0.2168148 -0.58899397  0.04152591  0.026465664
## skin     -0.4526842 -0.3133019 -0.29024528  0.05552072 -0.398212480
## insulin  -0.4531089 -0.2369177  0.26440040 -0.38123825 -0.449823190
## bmi      -0.4448432 -0.1002605 -0.29167139  0.14356977  0.574337402
## pedigree -0.2830633 -0.1135639  0.46420261  0.79259623 -0.003713205
## age      -0.1809845  0.6246230  0.09314575  0.05823948 -0.105317120
##                  PC6         PC7         PC8
## npreg    -0.37621262 -0.58753663  0.16849078
## glucose   0.15688073 -0.03948877  0.45326205
## dbp       0.69370610 -0.15736975 -0.03943801
## skin     -0.23522956  0.32171020  0.54054646
## insulin   0.06654788 -0.17205845 -0.53566646
## bmi      -0.48261941 -0.07719859 -0.34297714
## pedigree  0.23889958 -0.07773974  0.01246200
## age      -0.05355631  0.69524920 -0.26017562
```

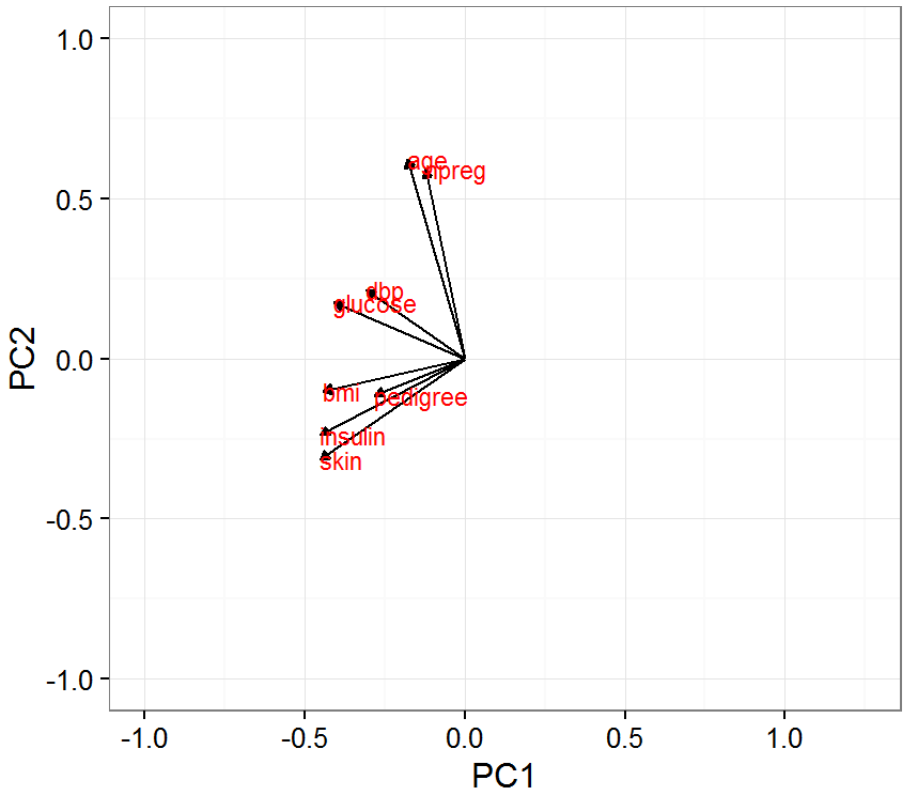
```
pca_data <- data.frame(pca$x, Diabetic=pima_full$diabetic)
head(pca_data)
```

```
##          PC1        PC2        PC3        PC4         PC5        PC6
## 1 -1.0173148  1.2628405  0.1387275  0.4979875  0.07411197 -0.3213504
## 2  1.2598047 -0.7536751 -0.6686611  0.2947882 -0.61828905  0.1491513
## 3  0.4721765  1.5850733  1.8725728 -0.2687733  0.55689213  0.6813908
## 4  1.2253848 -1.2964824 -0.6895155 -0.5521642 -0.55922809  0.1203129
## 5 -2.5382931 -2.1966005  3.2316033  3.9417273  0.71300265 -0.2059609
## 6  1.5962029  0.8087923 -0.1580763 -0.4869840  0.10917321  0.5589964
##           PC7        PC8 Diabetic
## 1  0.92360414  0.9272363      Yes
## 2  0.82050000  0.3120539       No
## 3 -1.07240714  1.2865993      Yes
## 4 -0.01248025 -0.1256445       No
## 5  0.50136173 -0.2724417      Yes
## 6 -0.58306926  0.0357253       No
```

```
ggplot(pca_data, aes(x=PC1, y=PC2, color=Diabetic)) + geom_point()
```



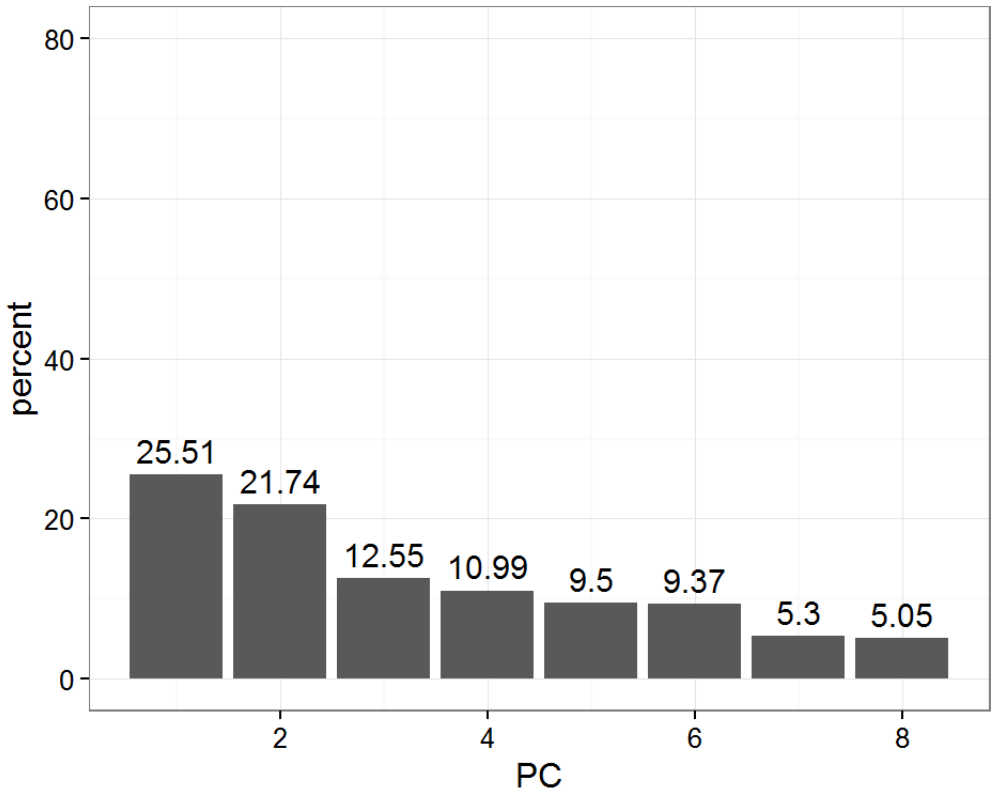

```
# capture the rotation matrix in a data frame
rotation_data <- data.frame(pca$rotation, variable=row.names(pca$rotation))
# define arrow style
arrow_style <- arrow(length = unit(0.05, "inches"),
                     type = "closed")
# now plot, using geom_segment() for arrows and geom_text
ggplot(rotation_data) +
  geom_segment(aes(xend=PC1, yend=PC2), x=0, y=0, arrow=arrow_style) +
  geom_text(aes(x=PC1, y=PC2, label=variable), hjust=0, size=3, color='red') +
  xlim(-1.,1.25) +
  ylim(-1.,1.) +
  coord_fixed() # fix aspect ratio to 1:1
```




```
percent <- 100*pca$sdev^2/sum(pca$sdev^2)
pca$rotation
```

```
##                 PC1        PC2         PC3         PC4          PC5
## npreg    -0.1228742  0.5951911 -0.02071036  0.10513489 -0.322556825
## glucose  -0.4096361  0.1770538  0.43505735 -0.43195394  0.439742705
## dbp      -0.3100062  0.2168148 -0.58899397  0.04152591  0.026465664
## skin     -0.4526842 -0.3133019 -0.29024528  0.05552072 -0.398212480
## insulin  -0.4531089 -0.2369177  0.26440040 -0.38123825 -0.449823190
## bmi      -0.4448432 -0.1002605 -0.29167139  0.14356977  0.574337402
## pedigree -0.2830633 -0.1135639  0.46420261  0.79259623 -0.003713205
## age      -0.1809845  0.6246230  0.09314575  0.05823948 -0.105317120
##                 PC6         PC7         PC8
## npreg    -0.37621262 -0.58753663  0.16849078
## glucose   0.15688073 -0.03948877  0.45326205
## dbp       0.69370610 -0.15736975 -0.03943801
## skin     -0.23522956  0.32171020  0.54054646
## insulin   0.06654788 -0.17205845 -0.53566646
## bmi      -0.48261941 -0.07719859 -0.34297714
## pedigree  0.23889958 -0.07773974  0.01246200
## age      -0.05355631  0.69524920 -0.26017562
```

```
perc_data <- data.frame(percent=percent, PC=1:length(percent))
ggplot(perc_data, aes(x=PC, y=percent)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=round(percent, 2)), size=4, vjust=-.5) +
  ylim(0, 80)
```



All variables contribute negatively to PC1, as seen by the negative values in the rotational matrix, hence PC1 decreases with inceasing values for npreg,glucose, dbp, skin, insulin, bmi, pedigree, an age.This suggests all criteria may vary together. However, glucose, skin, insulin, and bmi contribute the most negatively at around -0.4.

PC2 seems to measure the difference between non diabetics and diabetics. Most diabetics score Positvely for PC2 but negative on PC1. While most non-diabetics score positively in PC1. This can be seen in the scatter plot.So if we see which values score positevely in PC2 we will have an idea of what variables contribute more actively to the onset of diabetes which are: npreg, glucose, dbp, and age.

Finally, the bar plot shows that most the percent variance caused by PC1 and PC2, 25.51, 21.74 respectively.

Do non diabetic women have similar glucose and diastolic blood pressure levels as diabetic women?

```
pima_full%>% select(-diabetic,-npreg,-skin,-insulin,-bmi,-pedigree,-age)%>% # remove all columns but
glucose and dpb
  kmeans(centers=5, nstart =10) ->        # do k-means clustering with 5 centers
  km                          # store result as `km`

# now display the results from the analysis
km
```

```
## K-means clustering with 5 clusters of sizes 30, 171, 198, 252, 101
##
## Cluster means:
##     glucose      dbp
## 1 121.63333  1.80000
## 2 140.88304 77.71930
## 3  88.10606 67.80303
## 4 113.50000 71.40079
## 5 177.35644 75.80198
##
## Clustering vector:
##   [1] 2 3 5 3 2 4 3 1 5 4 5 2 5 5 1 4 4 1 4 2 4 5 4 2 4 2 3 2 4 4 2 3 3 4 4
##  [36] 2 4 3 4 5 2 4 5 5 5 2 3 4 4 3 3 5 2 3 5 4 2 4 2 3 2 4 3 4 4 3 2 3 2 2
##  [71] 2 3 3 3 1 4 4 3 3 2 4 4 3 2 4 3 4 3 2 2 2 3 3 3 4 5 2 2 3 3 4 4 2 3 3
## [106] 5 2 3 3 5 2 4 3 3 3 5 4 4 2 4 3 4 4 4 4 5 4 5 3 3 4 3 3 2 4 2 4 4 4 2
## [141] 3 4 2 3 2 4 2 2 5 2 3 4 3 5 2 4 4 3 2 4 2 4 4 4 4 2 1 3 3 5 3 2 2 2 3
## [176] 4 3 2 5 5 2 4 2 4 4 5 1 3 2 4 4 4 2 4 2 4 3 4 4 5 5 3 5 3 2 5 2 4 2 4
## [211] 4 3 4 5 2 1 2 3 3 4 5 5 4 2 2 3 4 3 5 5 5 5 4 3 3 2 4 2 5 4 5 4 4 4 2
## [246] 3 3 3 4 4 4 5 2 5 1 3 2 4 3 1 4 3 1 4 4 4 3 4 3 4 4 4 4 2 2 2 5 4 2 2
## [281] 4 3 4 3 4 2 4 5 2 2 2 4 4 1 2 3 4 2 4 5 2 4 4 3 4 2 4 4 4 3 5 4 5 4 4
## [316] 4 2 4 2 4 5 4 4 4 3 1 4 3 5 1 4 2 5 2 3 4 3 2 2 1 3 3 2 3 3 3 5 4 1 3
## [351] 5 5 2 4 2 2 3 4 3 3 2 5 3 4 4 2 3 3 2 3 4 4 4 3 4 4 4 4 2 3 3 5 2 4 2
## [386] 4 3 2 3 5 3 2 2 3 5 4 4 3 5 5 4 4 2 2 2 5 3 2 3 4 4 3 4 4 2 5 5 2 3 1
## [421] 3 3 2 3 1 2 2 3 4 5 3 4 4 4 5 3 3 4 4 3 2 3 1 3 5 2 3 2 2 4 3 3 3 4 4
## [456] 3 3 1 2 2 2 4 2 4 2 4 4 4 2 2 4 3 3 1 2 2 5 3 5 3 3 3 4 5 4 3 5 2 4 3
## [491] 3 3 3 5 4 3 4 3 2 3 3 3 5 2 2 3 2 3 4 2 4 3 3 4 4 4 4 4 3 1 3 1 4 3 2
## [526] 2 3 2 3 3 3 5 5 2 5 5 4 3 4 3 3 4 3 4 4 3 4 5 3 3 3 3 3 3 2 4 3 2 4 3
## [561] 2 4 4 4 2 5 2 4 4 4 4 3 2 4 5 1 4 4 2 3 4 5 3 1 5 4 4 1 4 2 1 4 5 3 2
## [596] 4 4 5 5 4 2 4 4 3 4 1 4 3 5 3 4 3 4 2 2 3 4 4 4 2 3 4 4 3 3 3 4 4 2 1
## [631] 4 2 5 5 2 4 3 4 4 4 4 2 3 4 2 3 5 5 5 2 4 4 2 4 3 2 5 3 3 2 3 5 2 3 4
## [666] 3 3 5 3 4 2 2 4 2 2 4 2 4 2 3 2 5 1 2 4 4 4 5 1 4 3 4 5 3 2 4 4 2 4 5
## [701] 5 3 4 3 3 4 2 4 4 4 4 2 5 3 2 4 5 4 4 3 2 3 3 4 4 3 4 2 2 4 2 3 5 5 2
## [736] 4 4 5 2 2 2 4 4 5 3 5 3 4 4 4 4 3
##
## Within cluster sum of squares by cluster:
## [1] 22435.77 39260.19 42800.09 50713.52 29513.21
##  (between_SS / total_SS =  80.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

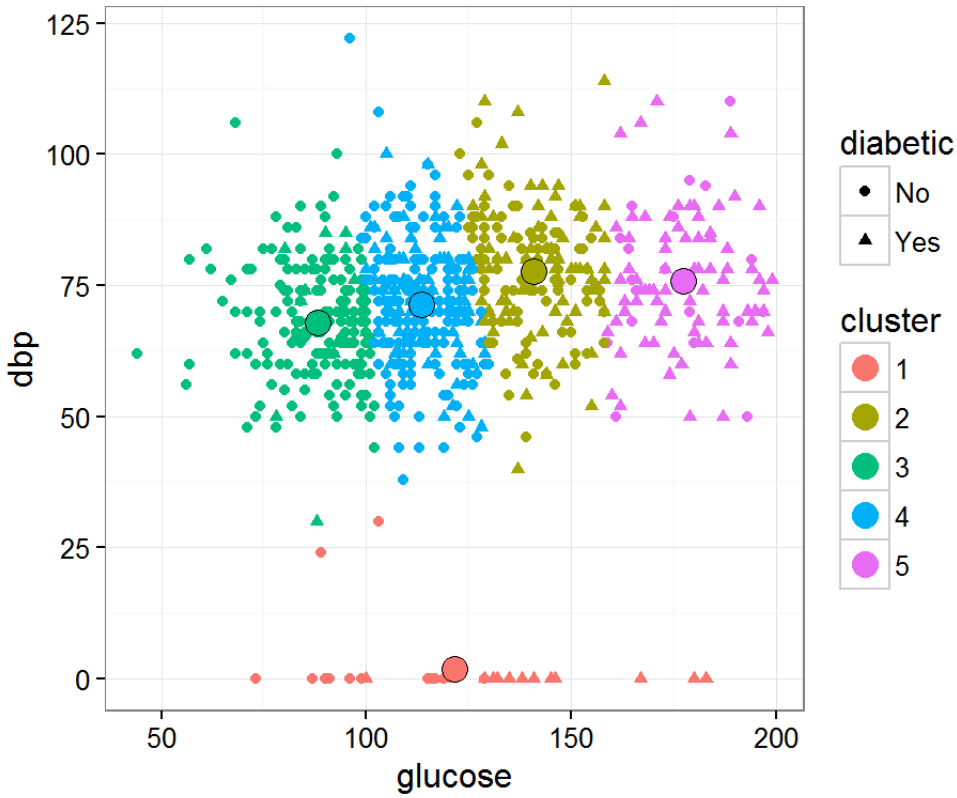
```
km$centers
```

```
##     glucose      dbp
## 1 121.63333  1.80000
## 2 140.88304 77.71930
## 3  88.10606 67.80303
## 4 113.50000 71.40079
## 5 177.35644 75.80198
```

```
km$cluster
```

```
##   [1] 2 3 5 3 2 4 3 1 5 4 5 2 5 5 1 4 4 1 4 2 4 5 4 2 4 2 3 2 4 4 2 3 3 4 4
##  [36] 2 4 3 4 5 2 4 5 5 5 2 3 4 4 3 3 5 2 3 5 4 2 4 2 3 2 4 3 4 4 3 2 3 2 2
##  [71] 2 3 3 3 1 4 4 3 3 2 4 4 3 2 4 3 4 3 2 2 2 3 3 3 4 5 2 2 3 3 4 4 2 3 3
## [106] 5 2 3 3 5 2 4 3 3 3 5 4 4 2 4 3 4 4 4 4 5 4 5 3 3 4 3 3 2 4 2 4 4 4 2
## [141] 3 4 2 3 2 4 2 2 5 2 3 4 3 5 2 4 4 3 2 4 2 4 4 4 4 2 1 3 3 5 3 2 2 2 3
## [176] 4 3 2 5 5 2 4 2 4 4 5 1 3 2 4 4 4 2 4 2 4 3 4 4 5 5 3 5 3 2 5 2 4 2 4
## [211] 4 3 4 5 2 1 2 3 3 4 5 5 4 2 2 3 4 3 5 5 5 5 4 3 3 2 4 2 5 4 5 4 4 4 2
## [246] 3 3 3 4 4 4 5 2 5 1 3 2 4 3 1 4 3 1 4 4 4 3 4 3 4 4 4 4 2 2 2 5 4 2 2
## [281] 4 3 4 3 4 2 4 5 2 2 2 4 4 1 2 3 4 2 4 5 2 4 4 3 4 2 4 4 4 3 5 4 5 4 4
## [316] 4 2 4 2 4 5 4 4 4 3 1 4 3 5 1 4 2 5 2 3 4 3 2 2 1 3 3 2 3 3 3 5 4 1 3
## [351] 5 5 2 4 2 2 3 4 3 3 2 5 3 4 4 2 3 3 2 3 4 4 4 3 4 4 4 4 2 3 3 5 2 4 2
## [386] 4 3 2 3 5 3 2 2 3 5 4 4 3 5 5 4 4 2 2 2 5 3 2 3 4 4 3 4 4 2 5 5 2 3 1
## [421] 3 3 2 3 1 2 2 3 4 5 3 4 4 4 5 3 3 4 4 3 2 3 1 3 5 2 3 2 2 4 3 3 3 4 4
## [456] 3 3 1 2 2 2 4 2 4 2 4 4 4 2 2 4 3 3 1 2 2 5 3 5 3 3 3 4 5 4 3 5 2 4 3
## [491] 3 3 3 5 4 3 4 3 2 3 3 3 5 2 2 3 2 3 4 2 4 3 3 4 4 4 4 4 4 3 1 3 1 4 3 2
## [526] 2 3 2 3 3 3 5 5 2 5 5 4 3 4 3 3 4 3 4 4 3 4 5 3 3 3 3 3 3 2 4 3 2 4 3
## [561] 2 4 4 4 2 5 2 4 4 4 4 3 2 4 5 1 4 4 2 3 4 5 3 1 5 4 4 1 4 2 1 4 5 3 2
## [596] 4 4 5 5 4 2 4 4 3 4 1 4 3 5 3 4 3 4 2 2 3 4 4 4 2 3 4 4 3 3 3 4 4 2 1
## [631] 4 2 5 5 2 4 3 4 4 4 4 2 3 4 2 3 5 5 5 2 4 4 2 4 3 2 5 3 3 2 3 5 2 3 4
## [666] 3 3 5 3 4 2 2 4 2 2 4 2 4 2 3 2 5 1 2 4 4 4 5 1 4 3 4 5 3 2 4 4 2 4 5
## [701] 5 3 4 3 3 4 2 4 4 4 4 2 5 3 2 4 5 4 4 3 2 3 3 4 4 3 4 2 2 4 2 3 5 5 2
## [736] 4 4 5 2 2 2 4 4 5 3 5 3 4 4 4 4 3
```

```
# add diabetic  information back into data
# use `factor(km$cluster)` to tell R that the cluster numbers represent distinct categories, not cont
inuous values

pima_clustered <- data.frame(pima_full, cluster=factor(km$cluster))

#Create Centroids
centroids <- data.frame(km$centers)
centroids <- data.frame(centroids, cluster=factor(1:5))
ggplot(pima_clustered, aes(x=glucose, y=dbp, color=cluster)) +
  geom_point(aes(shape=diabetic)) + # individual points from the `pima_clustered` data frame
  geom_point(data=centroids, size=4) + # centroids
  geom_point(data=centroids, shape=1, color="black", size=4) # black outline for centroid
```



In order, to see if we can group diabetics and non-diabetics based on glucose and dpb, I created a dataframe that only included only those two data entries.With five clusters, we still ended up clustering some diabetic cases among the non-diabetic cases. It seems that this k-means clustering is not a reliable method to separate the diabetic and non-diabetics for just those two variables.