Comparison of Classification and Clustering Algorithms on PimaIndiansDiabetes Dataset Using R

Talha Hanif Butt

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(mclust)
```

```
## Package 'mclust' version 5.4.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(fpc)
library(cluster)
library(clusteval)
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(ggplot2)
library(kmed)
library(mlbench)
```

Loading Pima Indians Diabetes Dataset

```
# attach the Pima Indians Diabetes Database to the environment
data("PimaIndiansDiabetes")
# rename the dataset
dataset <- PimaIndiansDiabetes
```

Partitioning Data for Validation

```
# create a list of 80% of the rows inthe original dataset we can use for training
validation_index <- createDataPartition(dataset$diabetes, p=0.80, list=FALSE)
# select 20% of the data for validation
validation <- dataset[-validation_index,]
# use the remaining 80% of data to training and testing the models
dataset <- dataset[validation_index,]
```

Getting Insights from Data

```
# dimensions of dataset
dim(dataset)
```

```
## [1] 615   9
```

```r
# list types for each attribute
sapply(dataset, class)
```

```
##  pregnant   glucose  pressure   triceps   insulin      mass  pedigree 
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" 
##       age  diabetes 
## "numeric"  "factor" 
```

```r
# take a peek at the first 6 rows of the data
head(dataset)
```

```
##   pregnant glucose pressure triceps insulin mass pedigree age diabetes
## 1        6     148       72      35       0 33.6    0.627  50      pos
## 2        1      85       66      29       0 26.6    0.351  31      neg
## 3        8     183       64       0       0 23.3    0.672  32      pos
## 4        1      89       66      23      94 28.1    0.167  21      neg
## 5        0     137       40      35     168 43.1    2.288  33      pos
## 6        5     116       74       0       0 25.6    0.201  30      neg
```

```r
# list the levels for the class
levels(dataset$diabetes)
```

```
## [1] "neg" "pos"
```

```r
# summarize the class distribution
percentage <- prop.table(table(dataset$diabetes)) * 100
cbind(freq=table(dataset$diabetes), percentage=percentage)
```

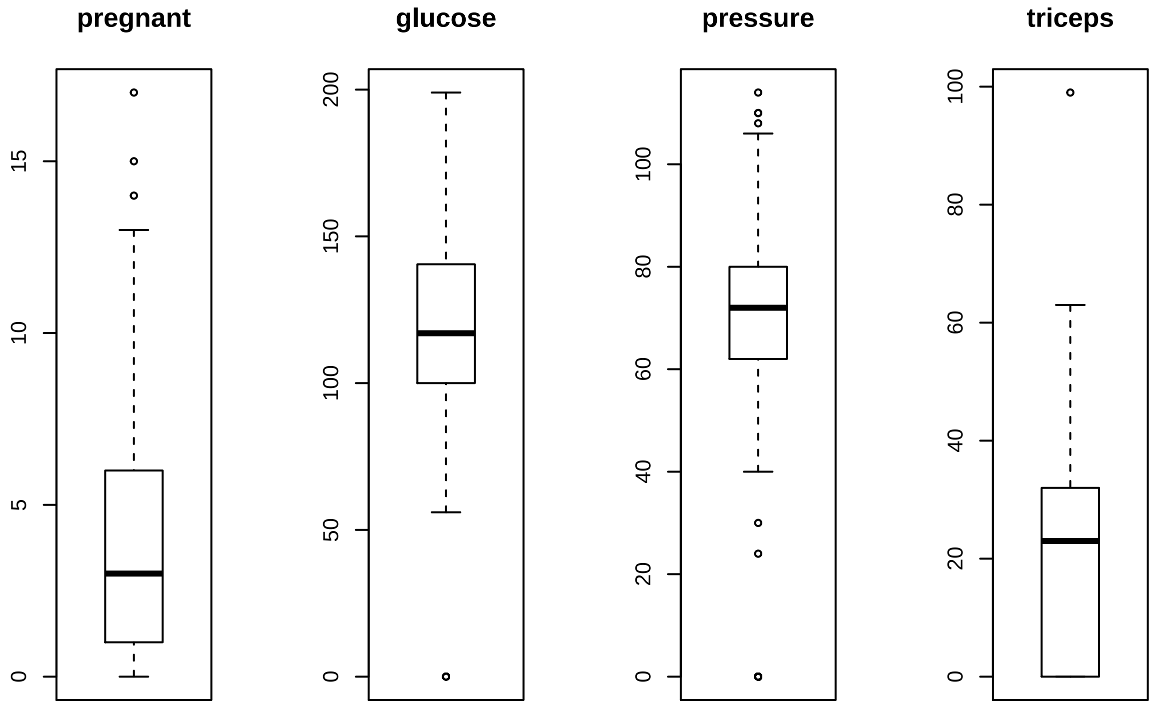
```
##     freq percentage
## neg  400   65.04065
## pos  215   34.95935
```

```r
# summarize attribute distributions
summary(dataset)
```

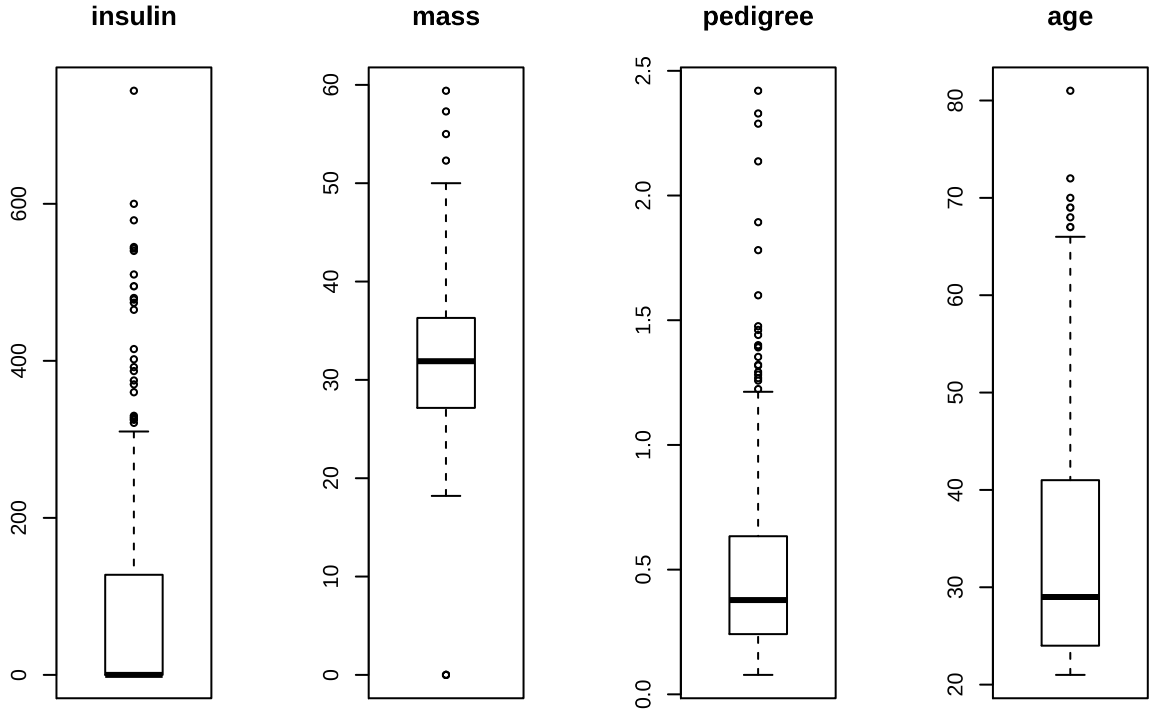
```
##     pregnant         glucose         pressure         triceps     
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
##  1st Qu.: 1.000   1st Qu.:100.0   1st Qu.: 62.00   1st Qu.: 0.00  
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
##  Mean   : 3.979   Mean   :121.0   Mean   : 68.66   Mean   :20.27  
##  3rd Qu.: 6.000   3rd Qu.:140.5   3rd Qu.: 80.00   3rd Qu.:32.00  
##  Max.   :17.000   Max.   :199.0   Max.   :114.00   Max.   :99.00  
##     insulin            mass          pedigree           age       
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0780   Min.   :21.00  
##  1st Qu.:  0.00   1st Qu.:27.15   1st Qu.:0.2415   1st Qu.:24.00  
##  Median :  0.00   Median :31.90   Median :0.3780   Median :29.00  
##  Mean   : 77.91   Mean   :31.72   Mean   :0.4727   Mean   :33.71  
##  3rd Qu.:127.50   3rd Qu.:36.30   3rd Qu.:0.6340   3rd Qu.:41.00  
##  Max.   :744.00   Max.   :59.40   Max.   :2.4200   Max.   :81.00  
##  diabetes 
##  neg:400  
##  pos:215  
##           
##           
##           
## 
```

```
# split input and output
x <- dataset[,1:8]
y <- dataset[,9]
```

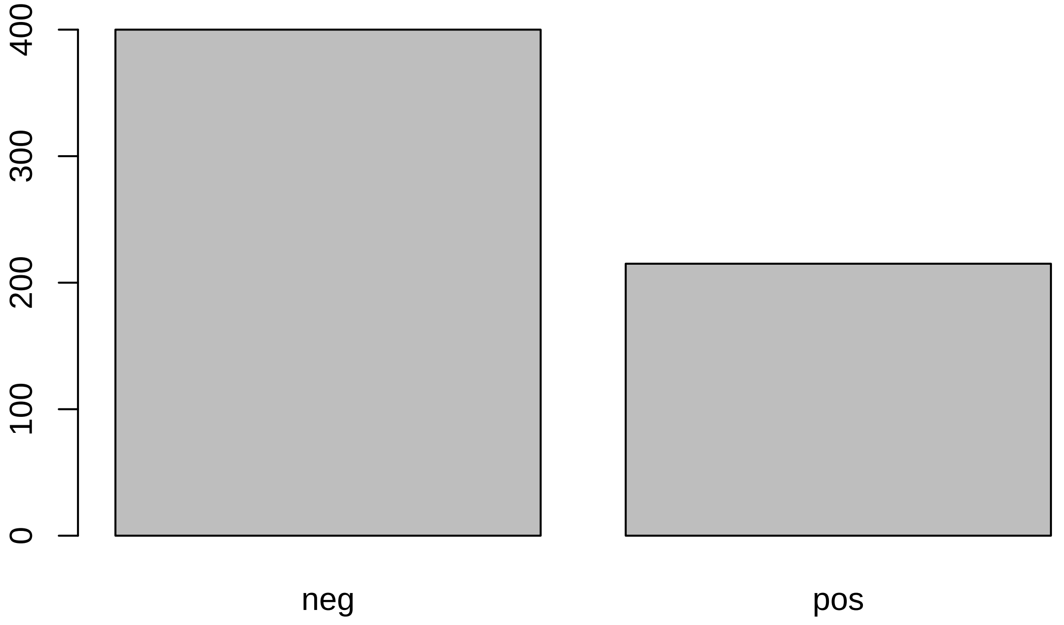
```
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 1:4) {
  boxplot(x[,i], main=names(PimaIndiansDiabetes)[i])
}
```



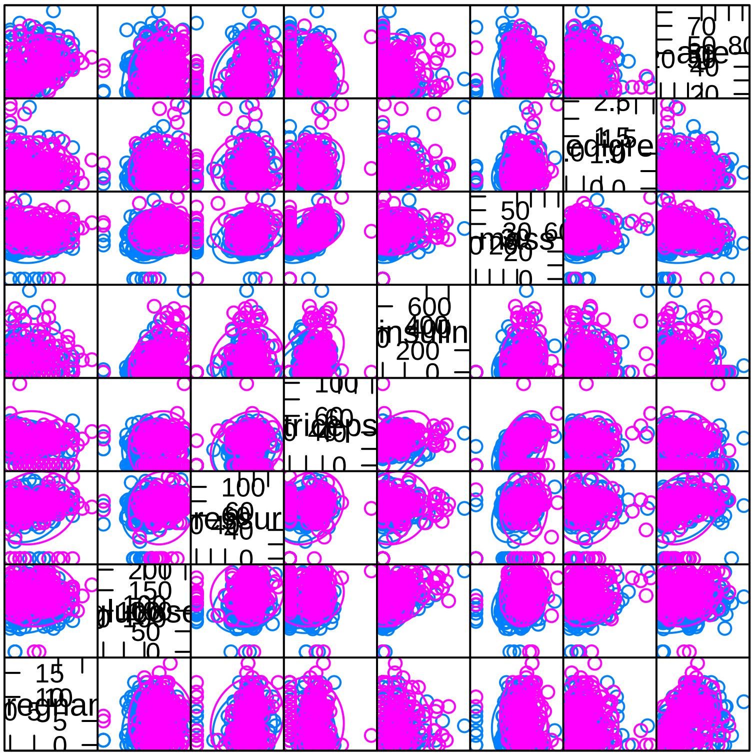
```
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 5:8) {
  boxplot(x[,i], main=names(PimaIndiansDiabetes)[i])
}
```



```
# barplot for class breakdown
plot(y)
```

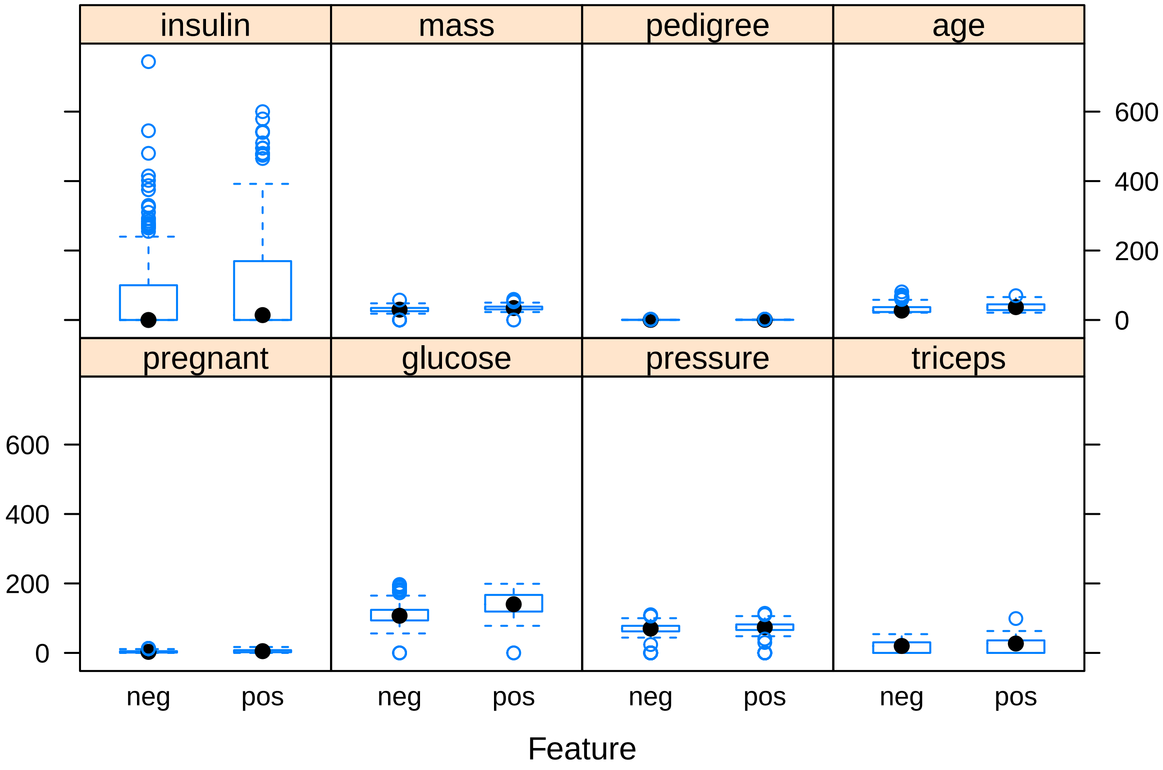


```
# scatterplot matrix
featurePlot(x=x, y=y, plot="ellipse")
```

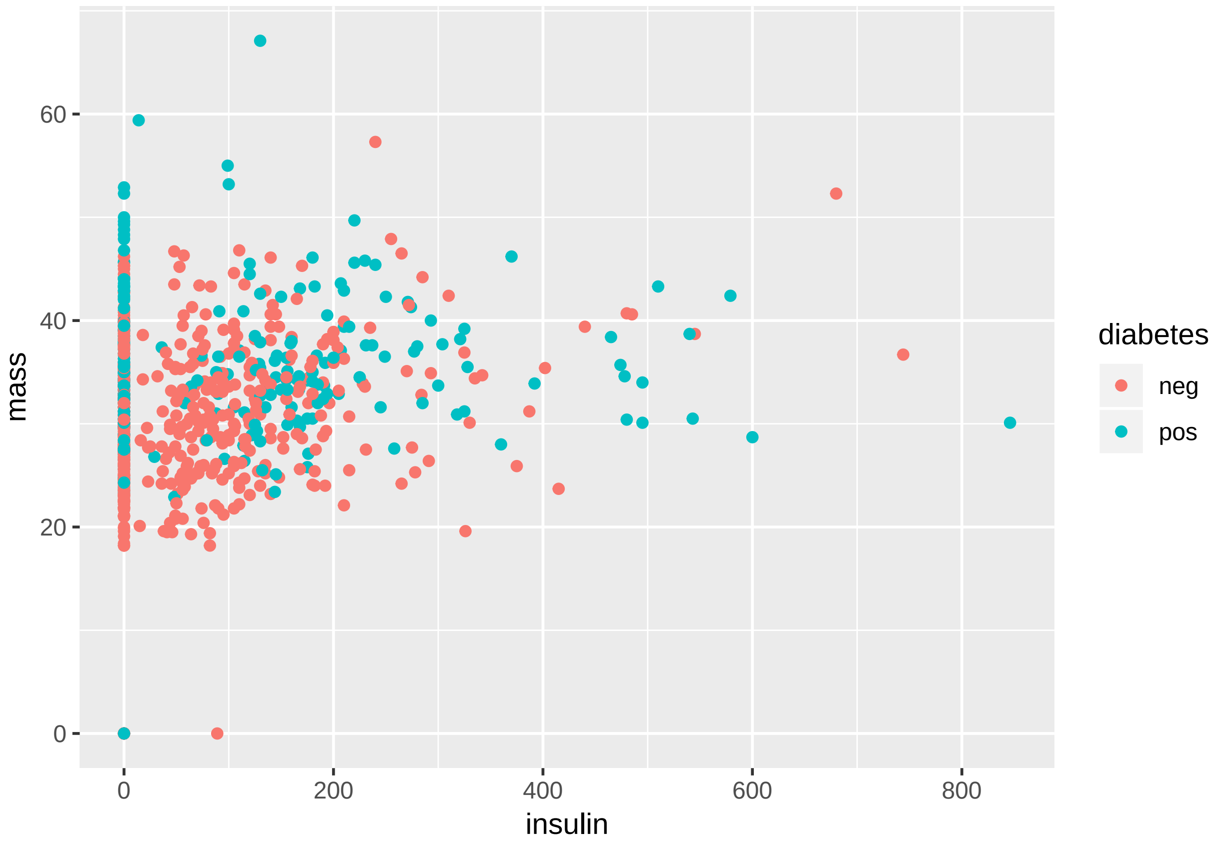


Scatter Plot Matrix

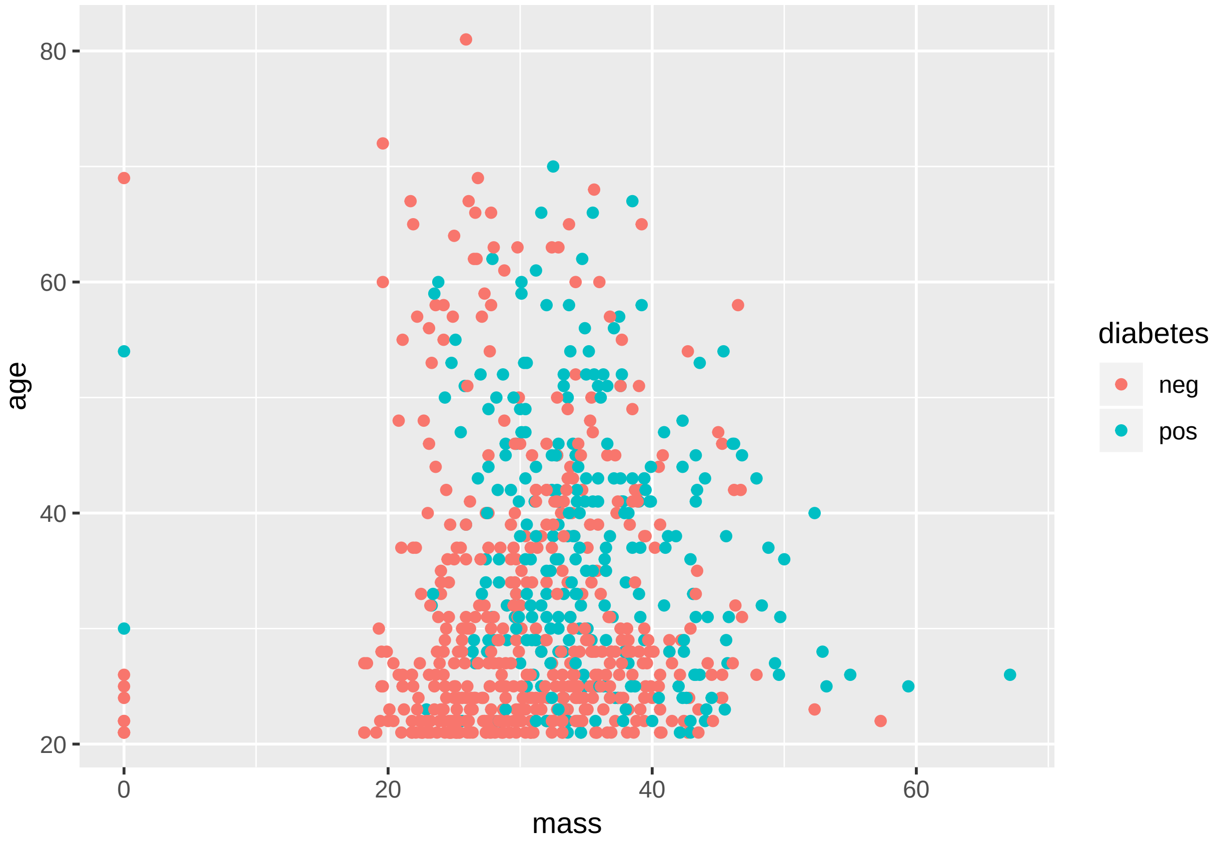
```r
# box and whisker plots for each attribute
featurePlot(x=x, y=y, plot="box")
```



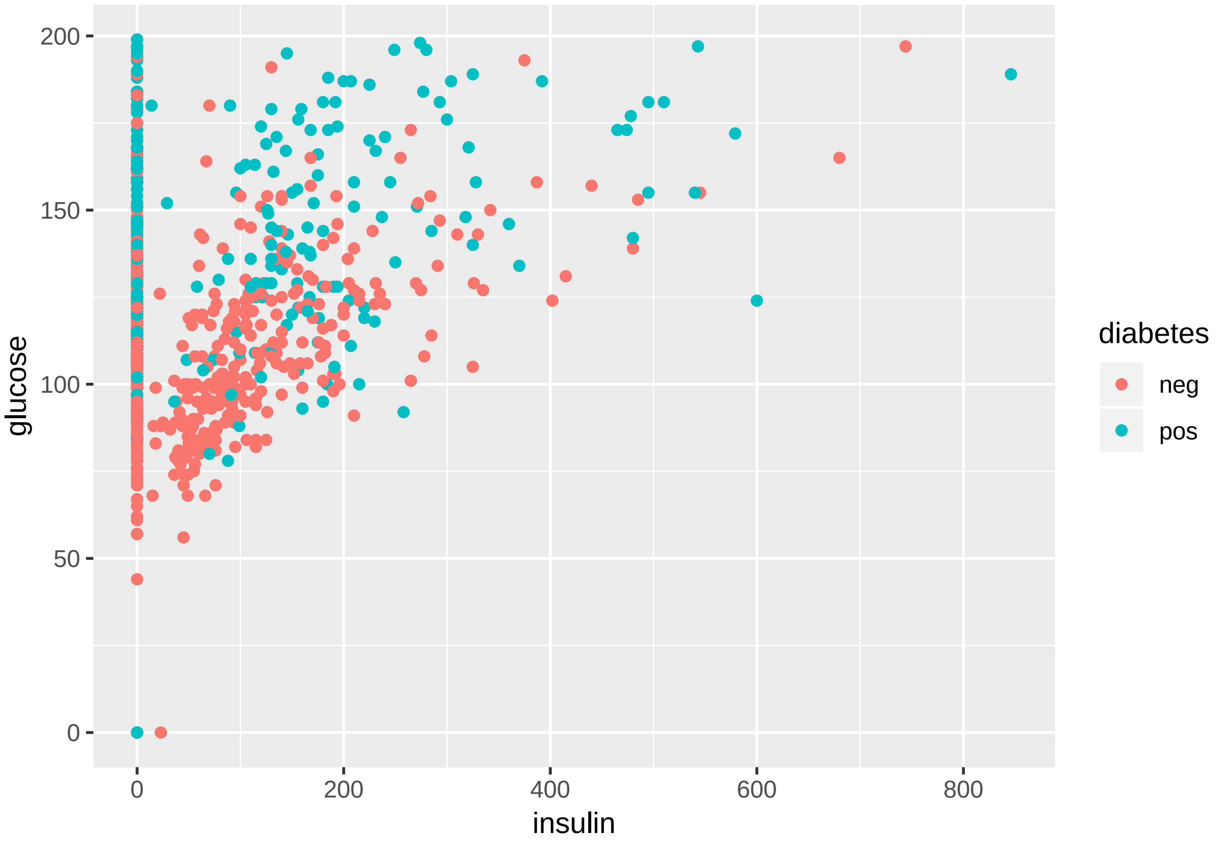
```
ggplot(PimaIndiansDiabetes, aes(insulin,mass, color = diabetes)) + geom_point()
```



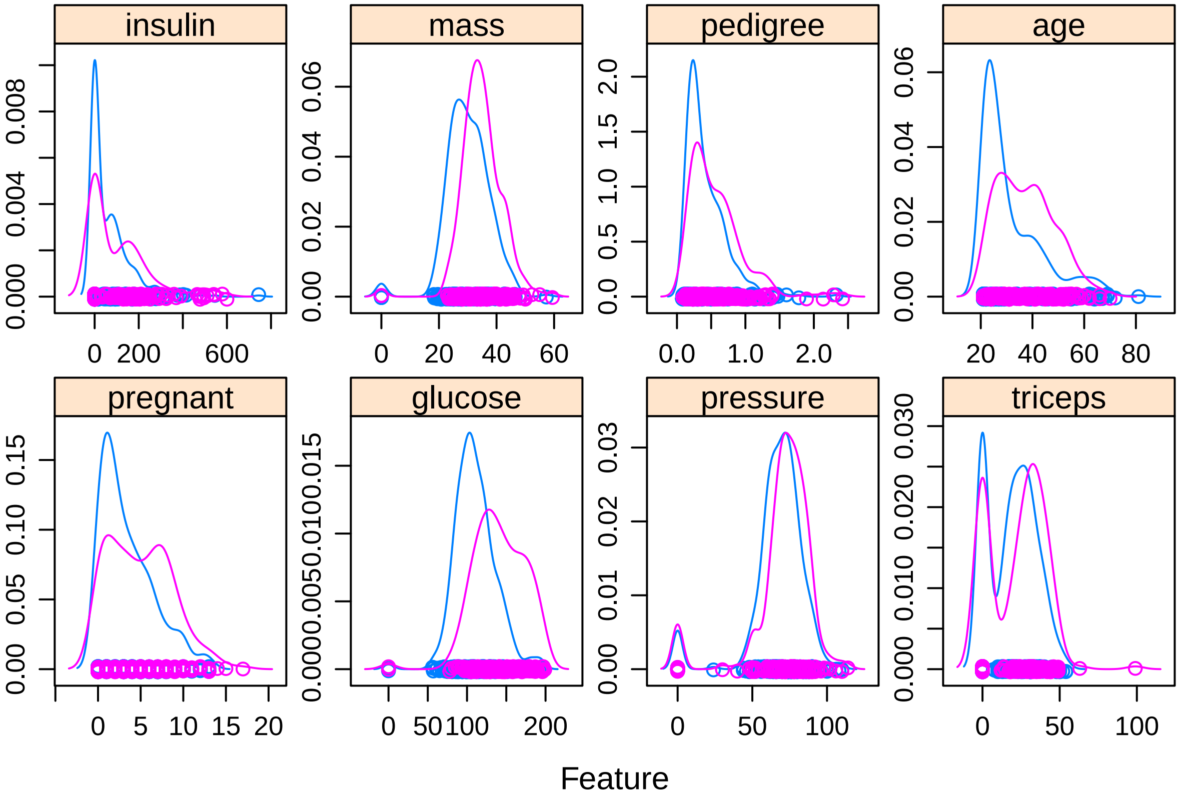
```
ggplot(PimaIndiansDiabetes, aes(mass,age, color = diabetes)) + geom_point()
```



```
ggplot(PimaIndiansDiabetes, aes(insulin,glucose, color = diabetes)) + geom_point()
```



```
# density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```



Applying Classification Algorithms

```
# Run algorithms using 10-fold cross validation
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
```

```
# kNN
set.seed(7)
fit.knn <- train(diabetes~., data=dataset, method="knn", metric=metric, trControl=control)
# SVM
set.seed(7)
fit.svm <- train(diabetes~., data=dataset, method="svmRadial", metric=metric, trControl=control)
# Random Forest
set.seed(7)
fit.rf <- train(diabetes~., data=dataset, method="rf", metric=metric, trControl=control)
```

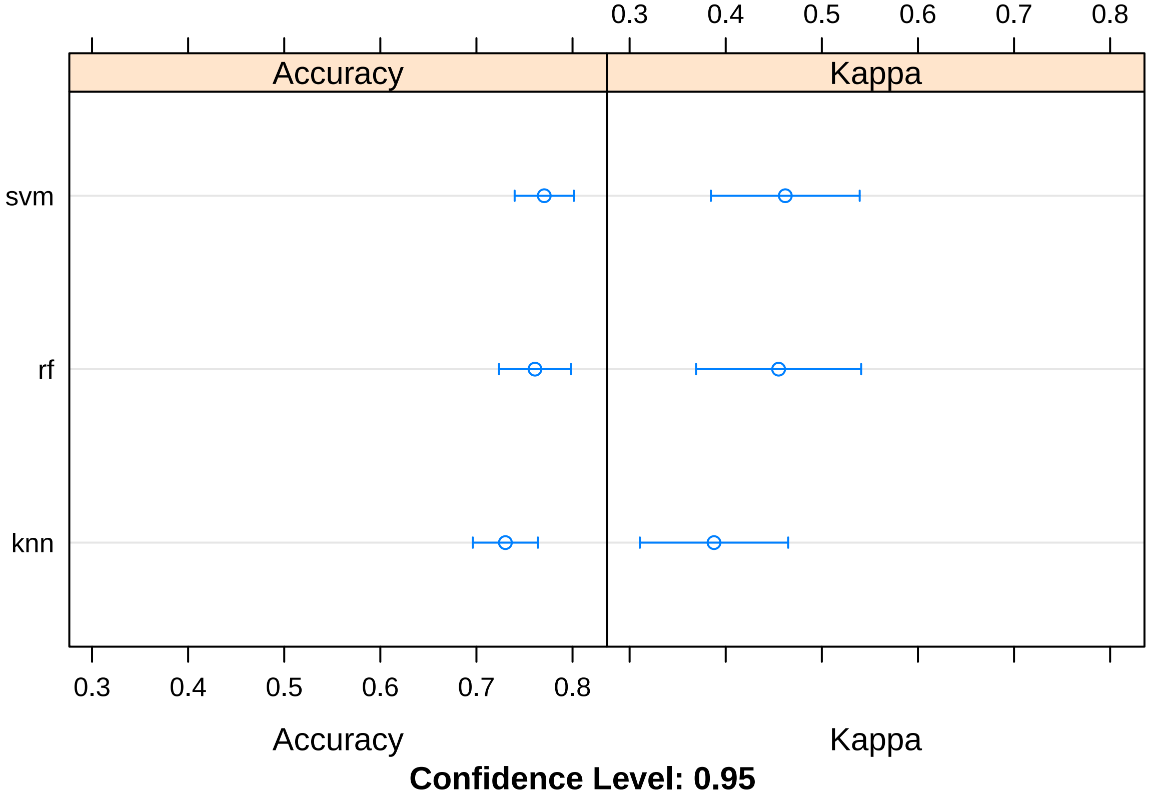
Comparison of the Classification Algorithms

```
# summarize accuracy of models
results <- resamples(list(knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
```

```
## Models: knn, svm, rf
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## knn 0.6557377 0.6895161 0.7419355 0.7301163 0.7633527 0.7868852    0
## svm 0.7258065 0.7377049 0.7559492 0.7705711 0.7960074 0.8387097    0
## rf  0.6935484 0.7213115 0.7562136 0.7609201 0.8024194 0.8524590    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## knn 0.2287778 0.2955868 0.4097712 0.3878377 0.4778409 0.5113986    0
## svm 0.3612565 0.3767561 0.4210771 0.4619966 0.5342957 0.6327014    0
## rf  0.3154613 0.3610598 0.4267109 0.4550382 0.5487726 0.6768687    0
```

```
# compare accuracy of models
dotplot(results)
```



Insights from the best model

```
# summarize Best Model
print(fit.svm)
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 615 samples
##   8 predictors
##   2 classes: 'neg', 'pos'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 554, 553, 554, 553, 554, 553, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.7641460  0.4374239
##   0.50  0.7705711  0.4619966
##   1.00  0.7608408  0.4482180
##
## Tuning parameter 'sigma' was held constant at a value of 0.1351037
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.1351037 and C = 0.5.
```

```
# estimate skill of SVM on the validation dataset
predictions <- predict(fit.svm, validation)
confusionMatrix(predictions, validation$diabetes)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg  88  22
##        pos  12  31
##
##                Accuracy : 0.7778
##                  95% CI : (0.7036, 0.8409)
##     No Information Rate : 0.6536
##     P-Value [Acc > NIR] : 0.000586
##
##                   Kappa : 0.4865
##  Mcnemar's Test P-Value : 0.122713
##
##             Sensitivity : 0.8800
##             Specificity : 0.5849
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.7209
##              Prevalence : 0.6536
##          Detection Rate : 0.5752
##    Detection Prevalence : 0.7190
##       Balanced Accuracy : 0.7325
##
##        'Positive' Class : neg
##
```

Applying Clustering Algorithms

```
# K-means
set.seed(20)
fit.kmeans <- kmeans(PimaIndiansDiabetes[, 1:8], 2, nstart = 20)
# Hierarchical Agglomerative
set.seed(20)
```

```
d <- dist(PimaIndiansDiabetes[,1:8], method = "euclidean") # distance matrix
fit.ha <- hclust(d, method="ward.D")
# K-Medoids Clustering
num <- as.matrix(PimaIndiansDiabetes[,1:8])
mrwdist <- distNumeric(num, num, method = "mrw")
fit.kmedoids <- fastkmed(mrwdist, ncluster = 2, iterate = 50)
```

Getting insights from Hierarchical Agglomerative Clustering

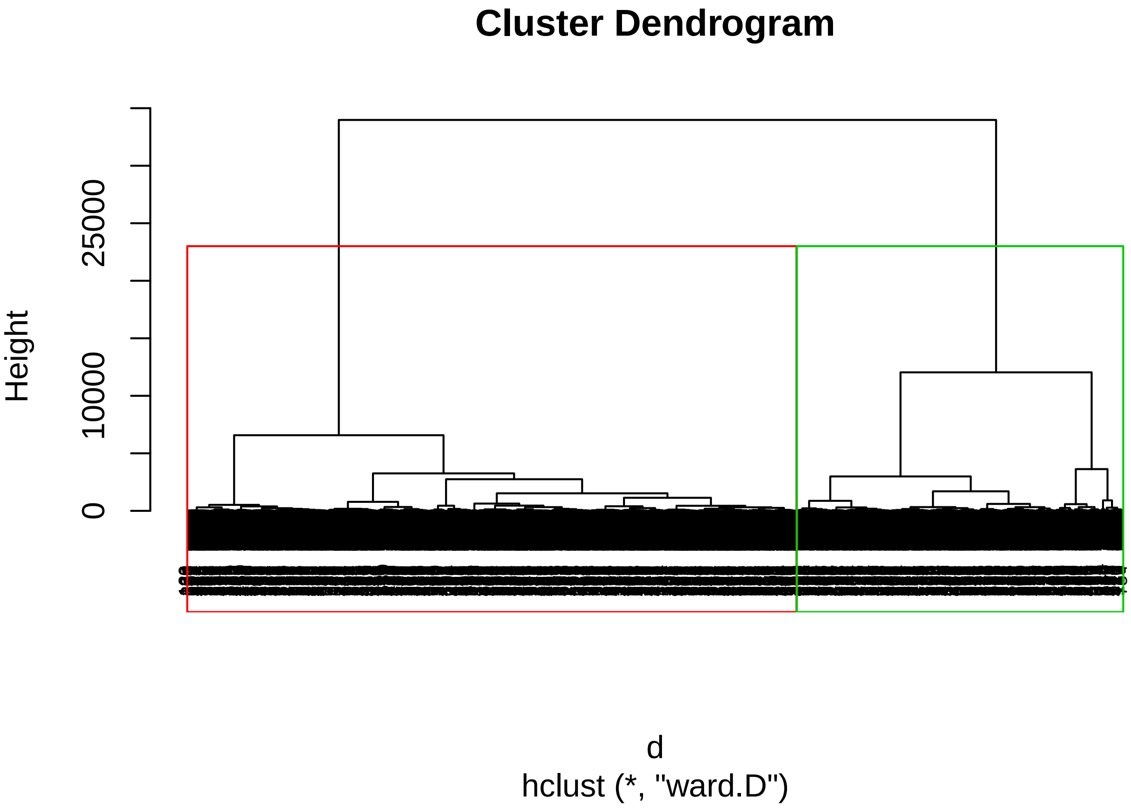
```
# Cut tree into 4 groups
sub_grp <- cutree(fit.ha, k = 2)

# Number of members in each cluster
table(sub_grp)
```

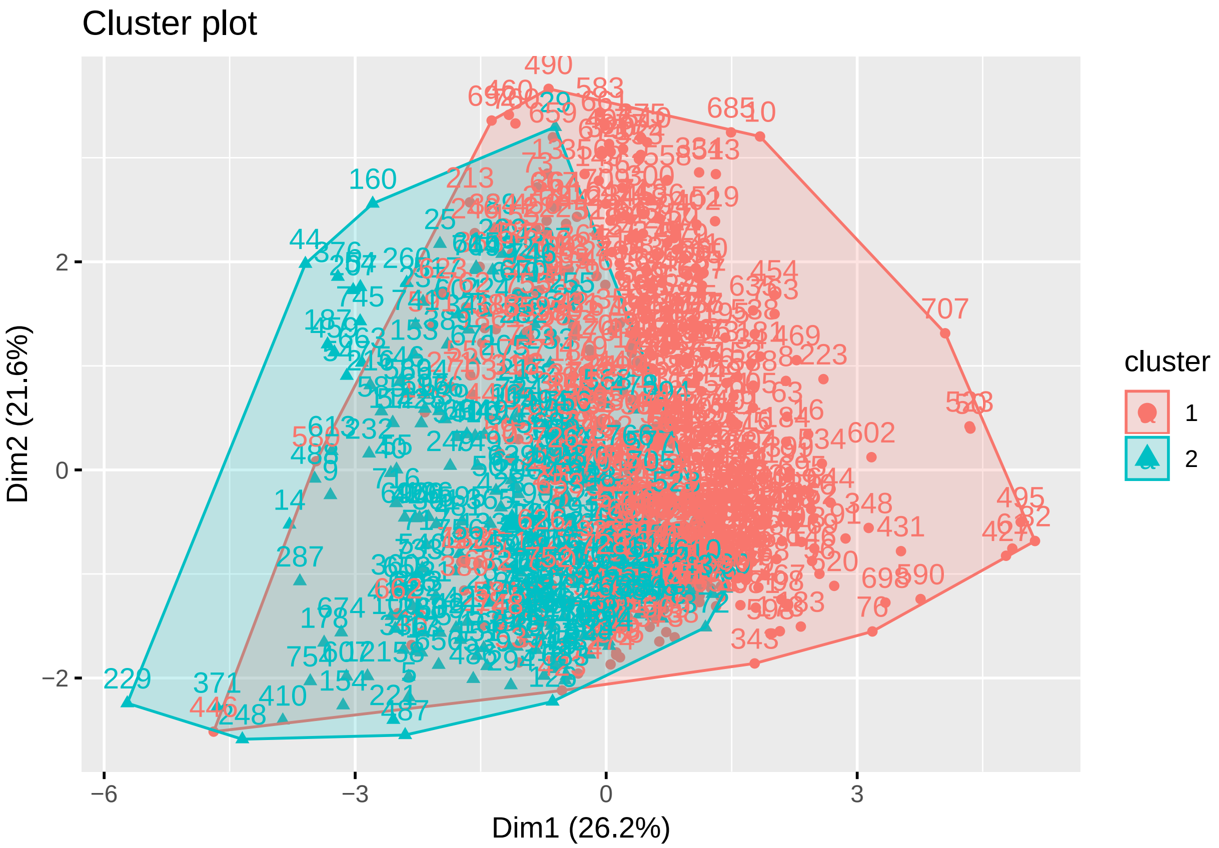
```
## sub_grp
##   1   2
## 500 268
```

```
## sub_grp
```

```
plot(fit.ha, cex = 0.6)
rect.hclust(fit.ha, k = 2, border = 2:5)
```



```
fviz_cluster(list(data = PimaIndiansDiabetes[,1:8], cluster = sub_grp))
```

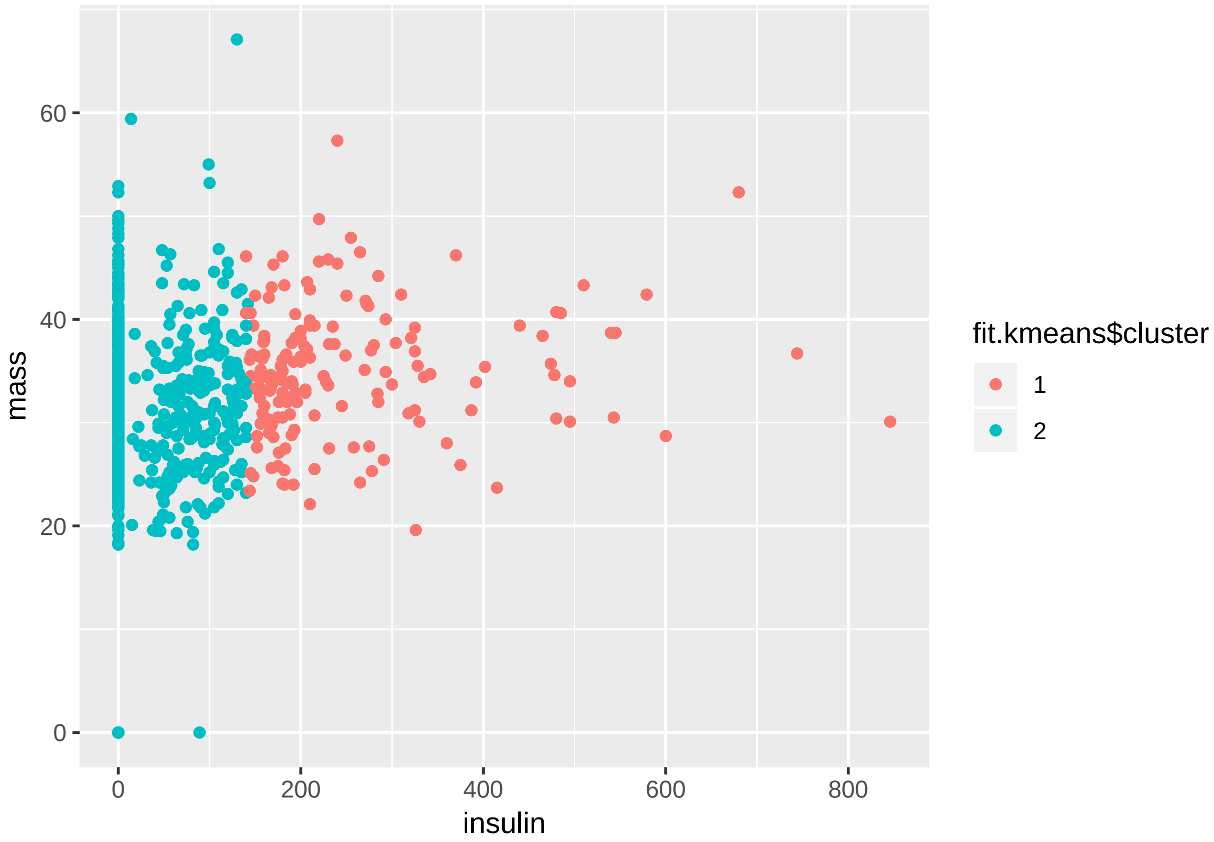


Getting insights from K-Means Clustering

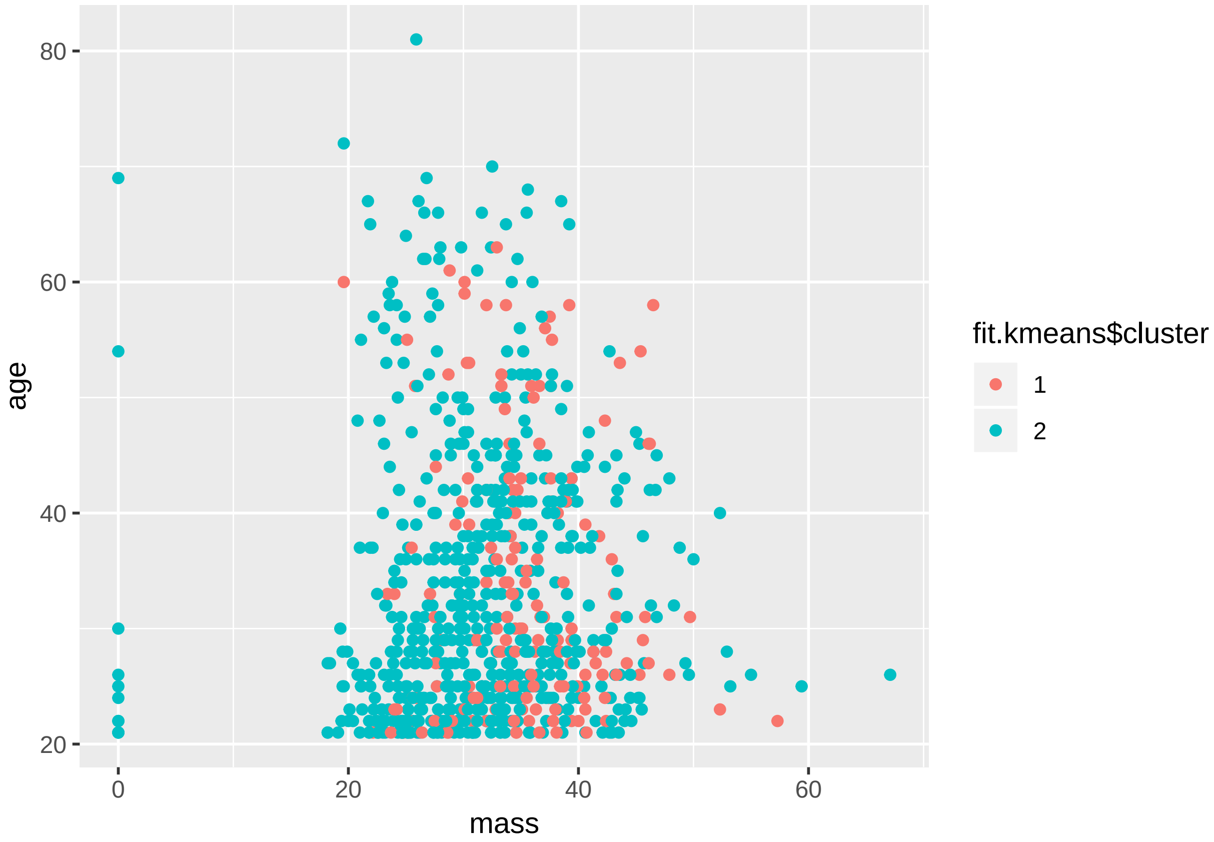
```
table(fit.kmeans$cluster, PimaIndiansDiabetes$diabetes)
```

```
##
##       neg pos
##     1  79  86
##     2 421 182
```

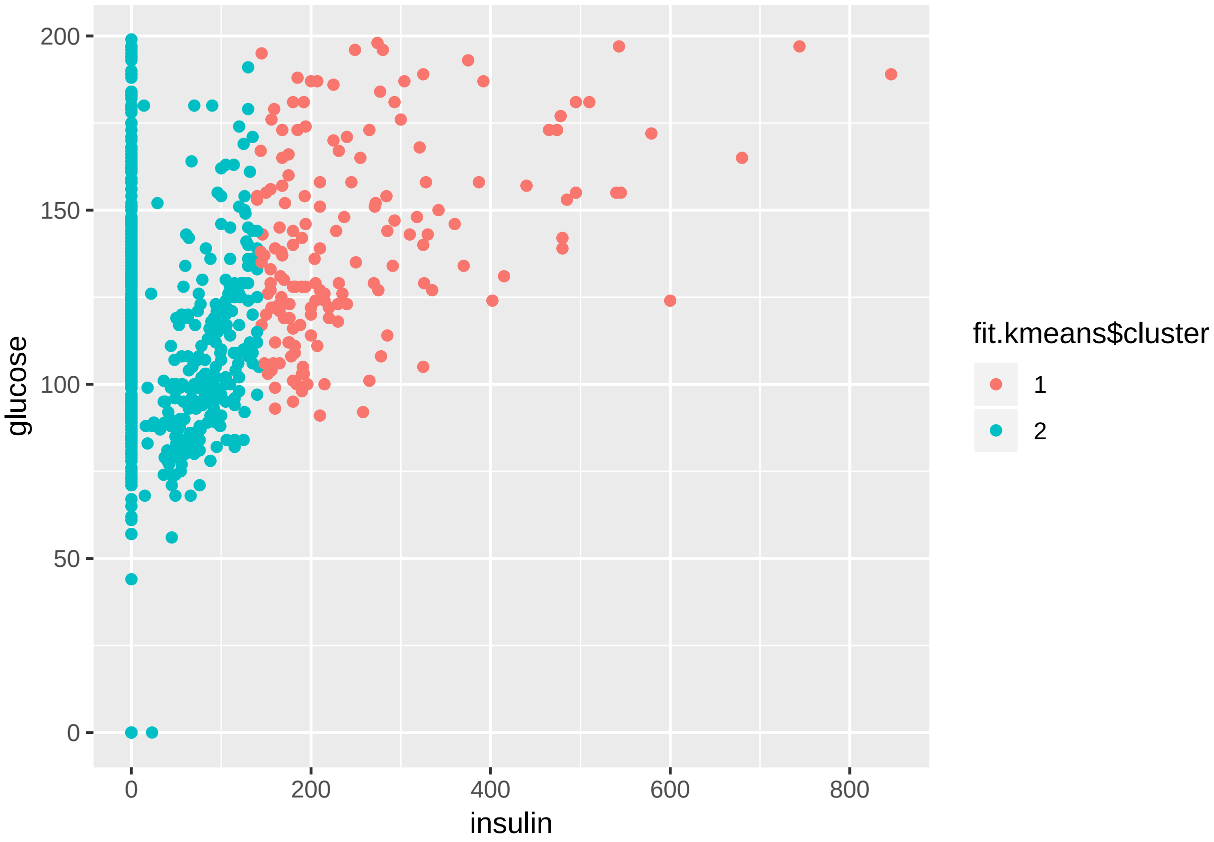
```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, mass, color = fit.kmeans$cluster)) + geom_point()
```



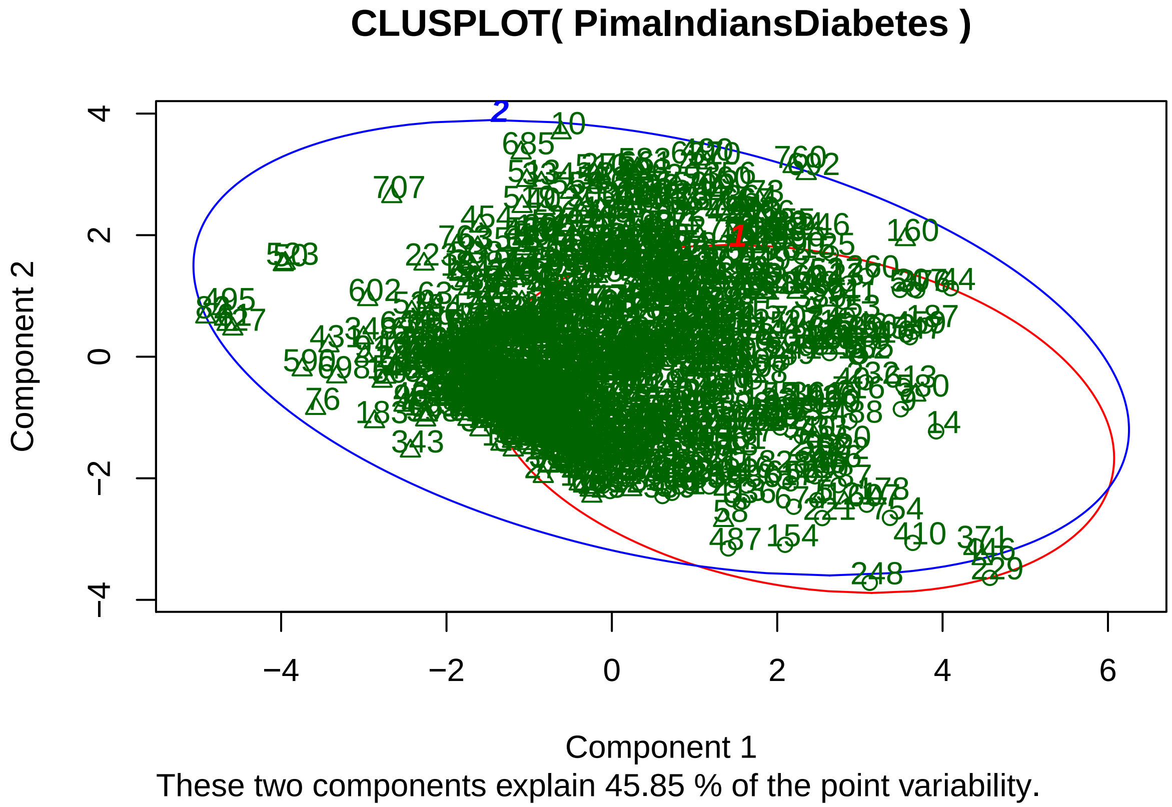
```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(PimaIndiansDiabetes, aes(mass, age, color = fit.kmeans$cluster)) + geom_point()
```



```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, glucose, color = fit.kmeans$cluster)) + geom_point()
```



```r
clusplot(PimaIndiansDiabetes, fit.kmeans$cluster, color=TRUE, shade=FALSE, labels=2, lines=0)
```

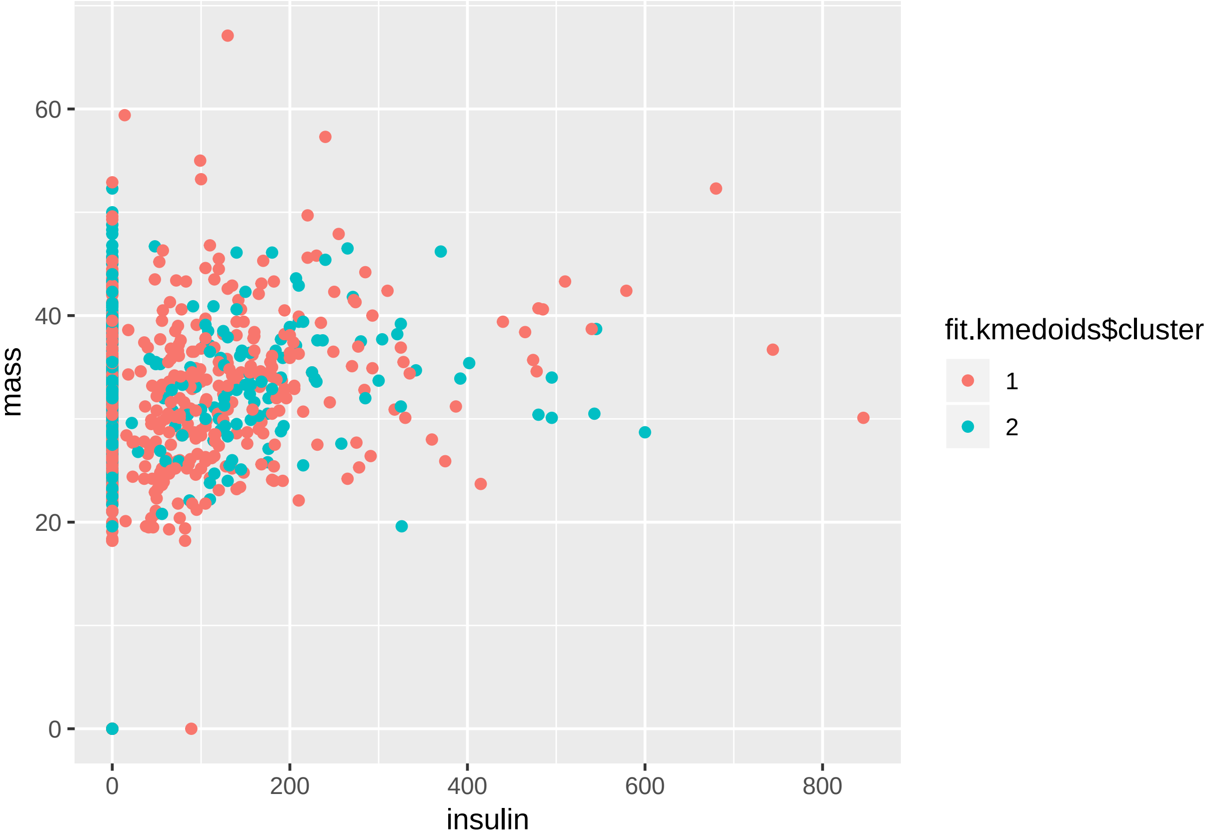


Getting insights from K-Medoids Clustering

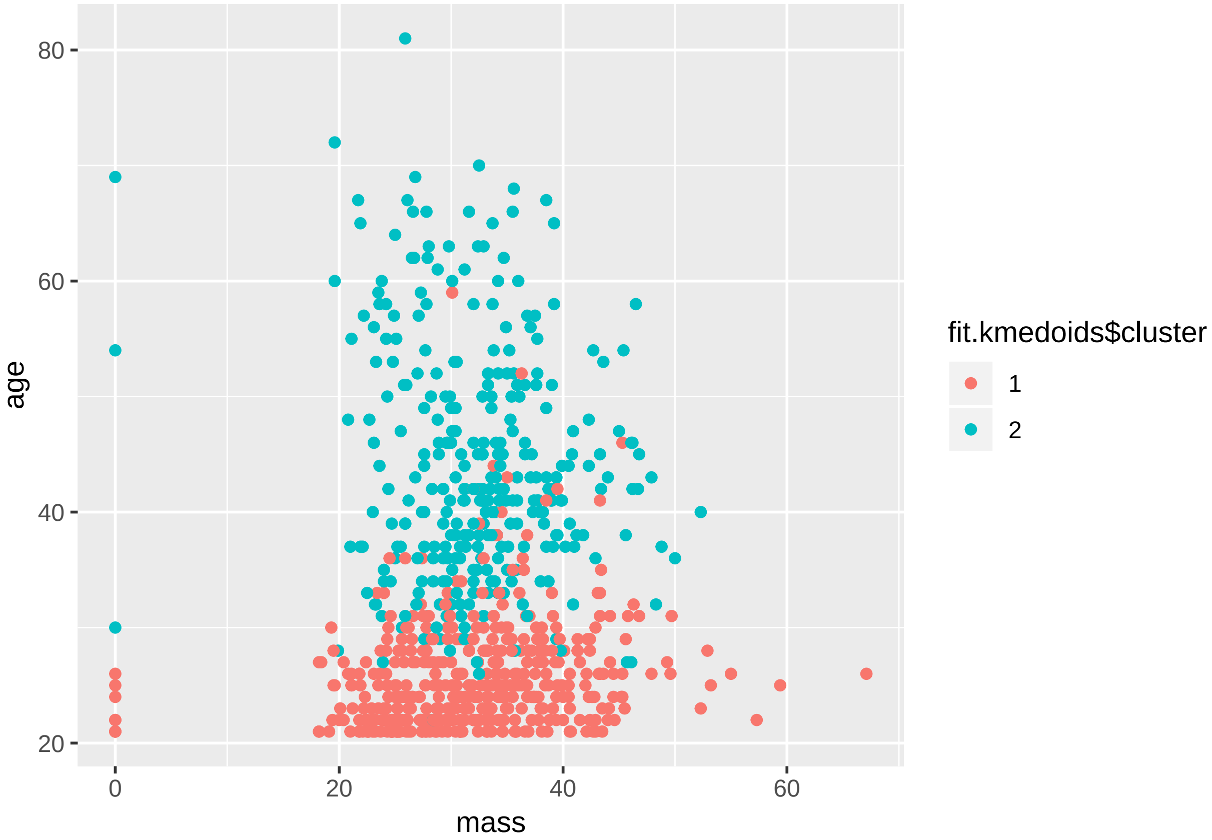
```
table(fit.kmedoids$cluster, PimaIndiansDiabetes[,9])
```

```
##
##      neg pos
##   1 342 109
##   2 158 159
```

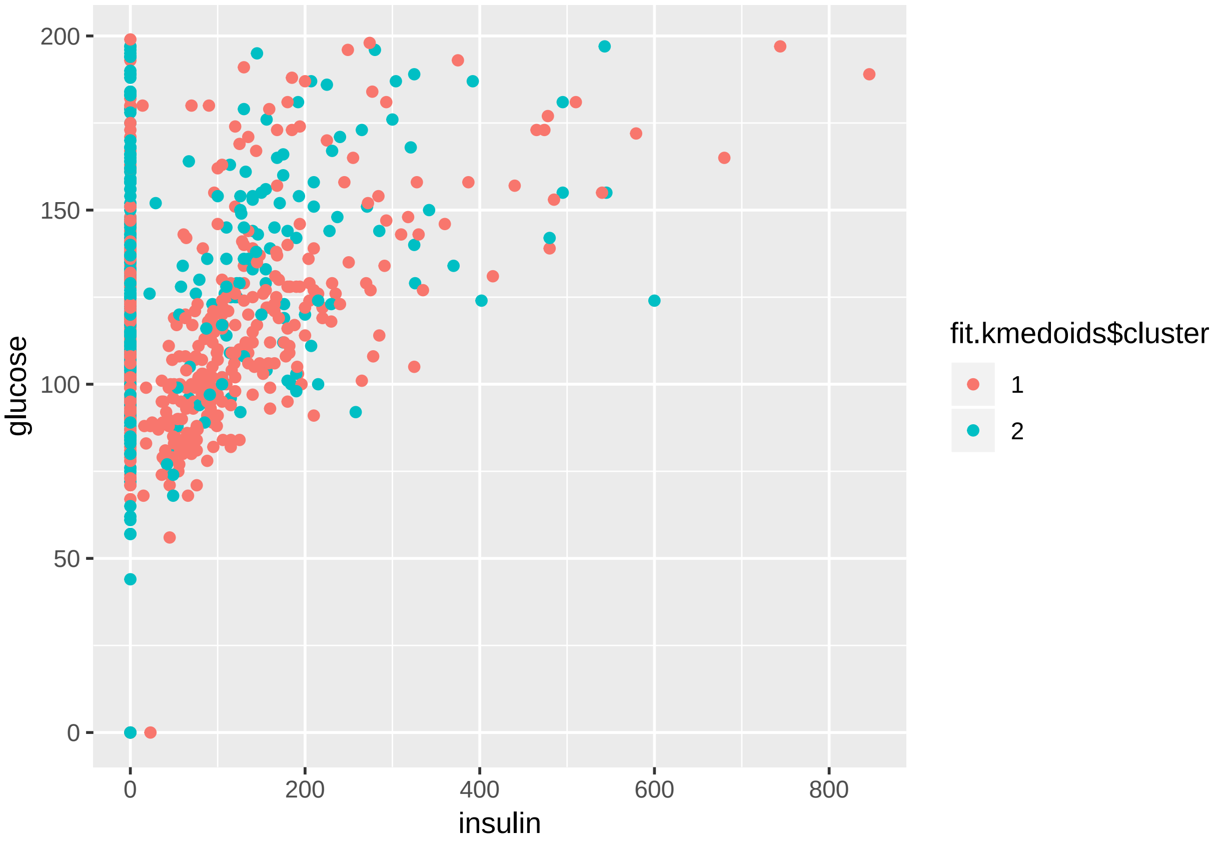
```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, mass, color = fit.kmedoids$cluster)) + geom_point()
```



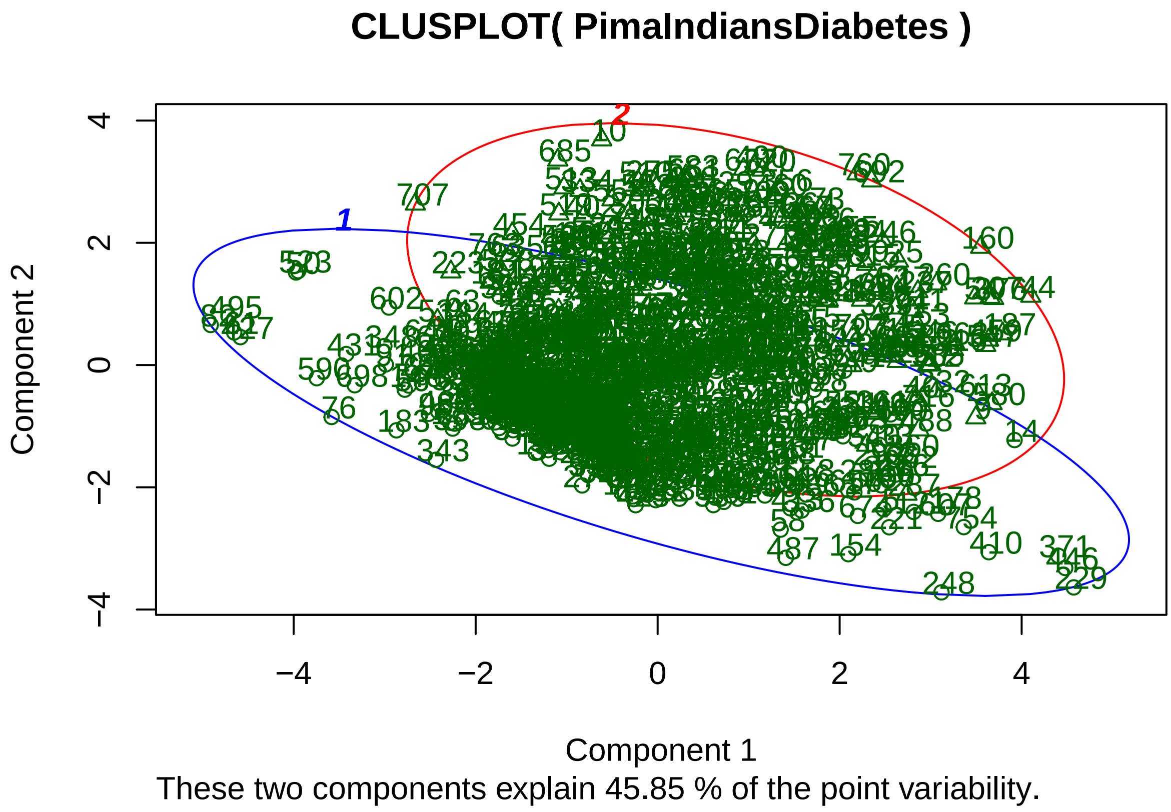
```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(PimaIndiansDiabetes, aes(mass, age, color = fit.kmedoids$cluster)) + geom_point()
```



```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, glucose, color = fit.kmedoids$cluster)) + geom_point()
```



```
clusplot(PimaIndiansDiabetes, fit.kmedoids$cluster, color=TRUE, shade=FALSE, labels=2, lines=0)
```



Conclusion

With better accuracy and kappa measures, SVM has outperformed other competitors on Glass Dataset while Hierarchical Agglomerative Clustering is the winner when compared with K-Means and K-Medoids Clustering on Glass Dataset as it has clustered data better evident from the Cluster Plot and Cluster Dendrogram.