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)
```

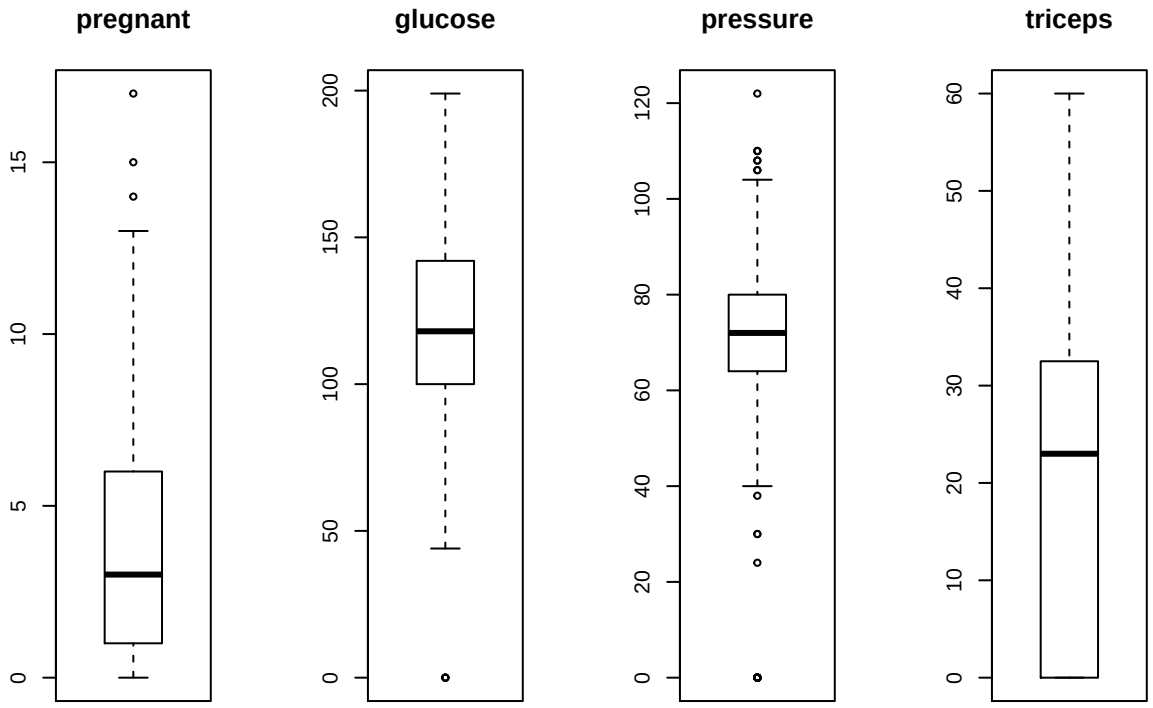
```
##     freq percentage
## neg  400   65.04065
## pos  215   34.95935
```

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

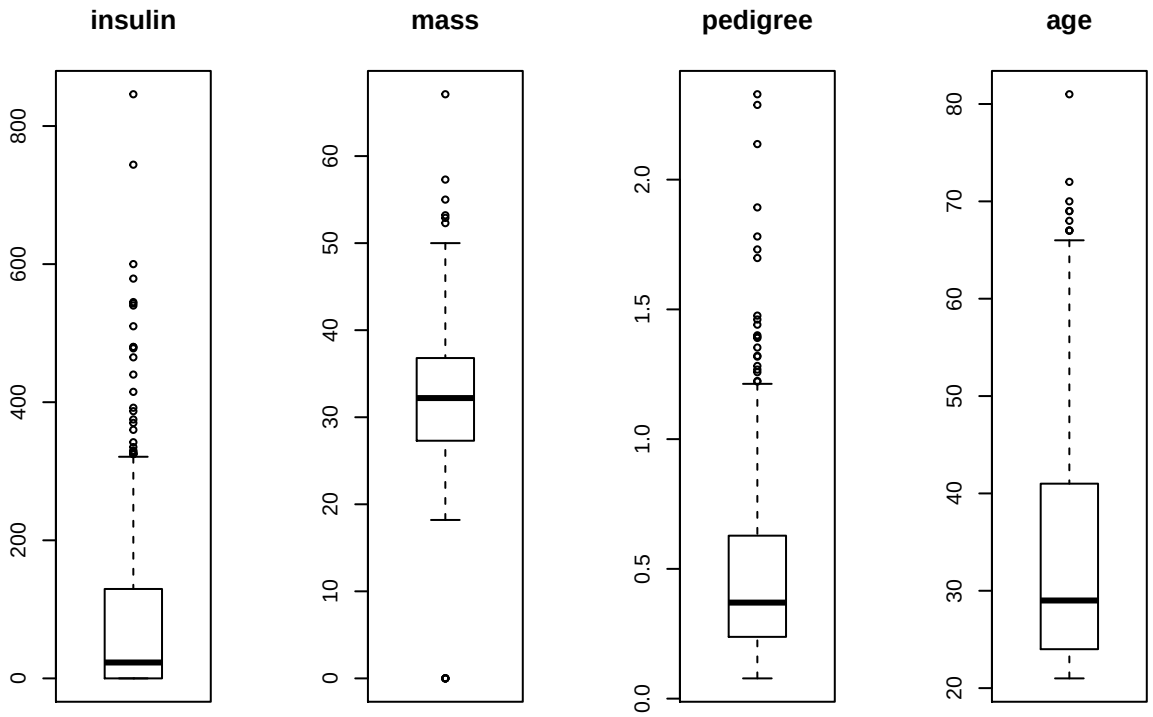
```
##     pregnant         glucose         pressure         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.:100.0   1st Qu.: 64.00   1st Qu.: 0.00
##  Median : 3.000   Median :118.0   Median : 72.00   Median :23.00
##  Mean   : 3.855   Mean   :121.6   Mean   : 69.57   Mean   :20.49
##  3rd Qu.: 6.000   3rd Qu.:142.0   3rd Qu.: 80.00   3rd Qu.:32.50
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :60.00
##     insulin           mass          pedigree           age
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:27.30   1st Qu.:0.2380   1st Qu.:24.00
##  Median : 23.00   Median :32.20   Median :0.3700   Median :29.00
##  Mean   : 79.65   Mean   :32.22   Mean   :0.4679   Mean   :33.41
##  3rd Qu.:129.50   3rd Qu.:36.80   3rd Qu.:0.6280   3rd Qu.:41.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.3290   Max.   :81.00
##  diabetes
##  neg:400
##  pos:215
##
##
##
##
```

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

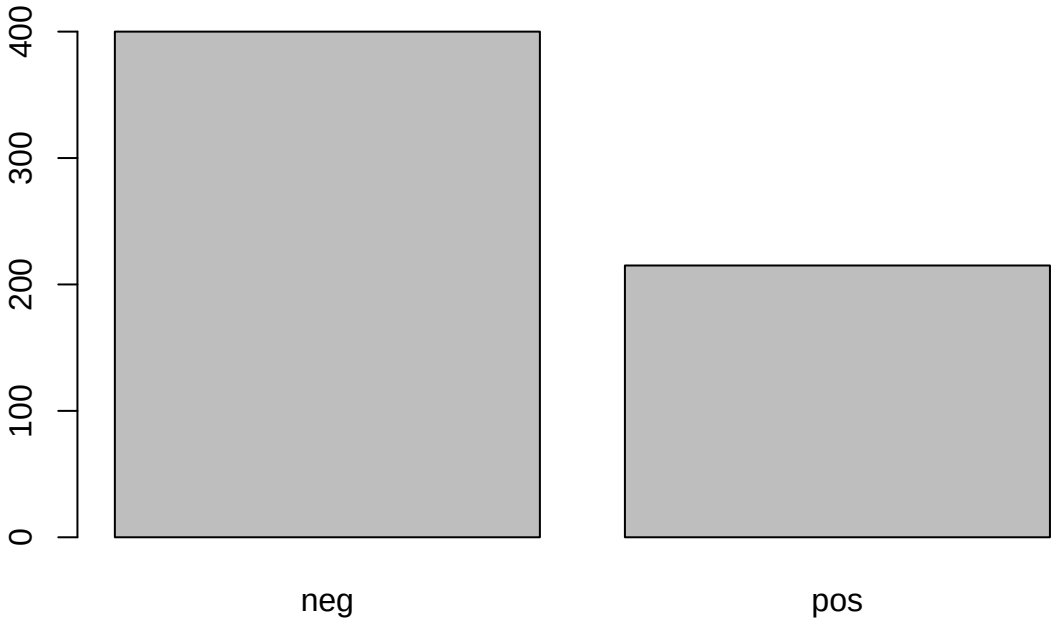
```
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 1:4) {
  boxplot(x[,i], main=names(PimaIndiansDiabetes)[i])
}
```



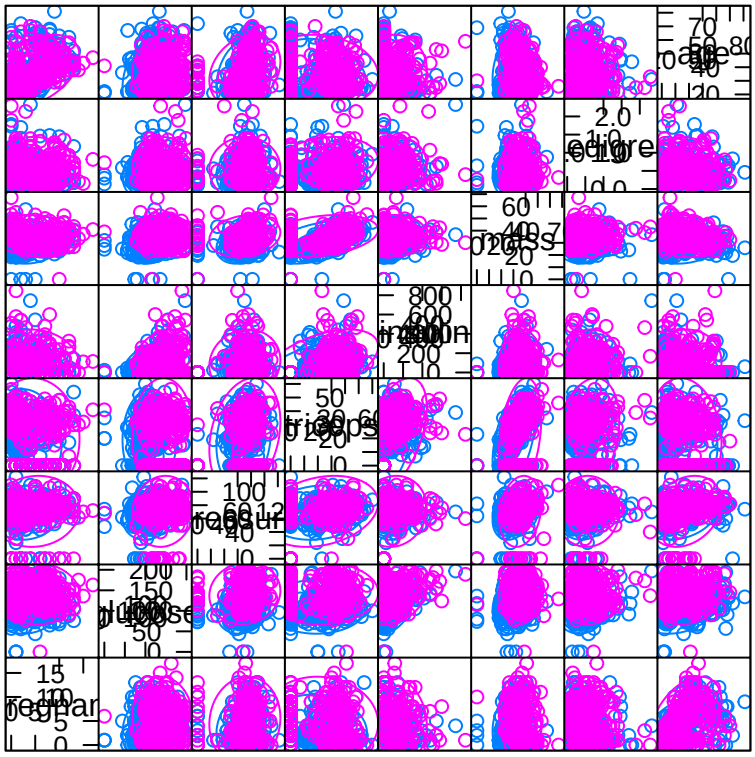

```
# boxplot for each attribute on one image
par(mfrow=c(1,4))
  for(i in 5:8) {
  boxplot(x[,i], main=names(PimaIndiansDiabetes)[i])
}
```




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

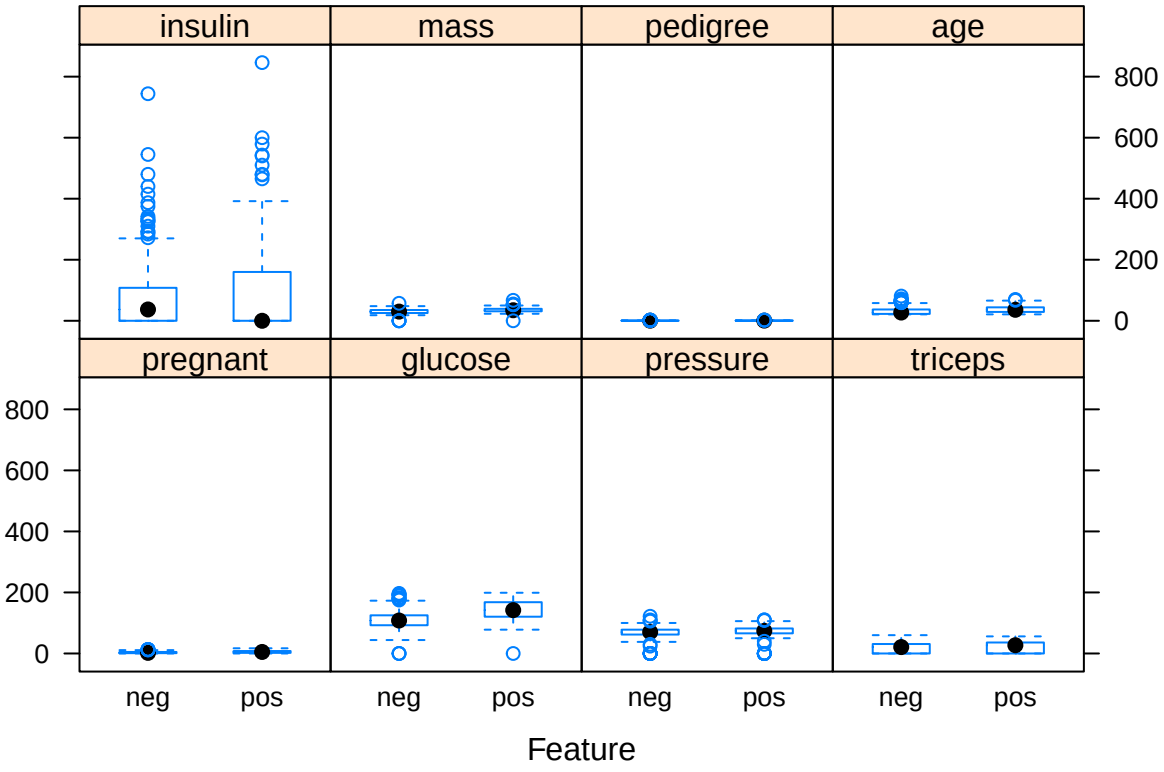



```
# scatterplot matrix
featurePlot(x=x, y=y, plot="ellipse")
```



Scatter Plot Matrix

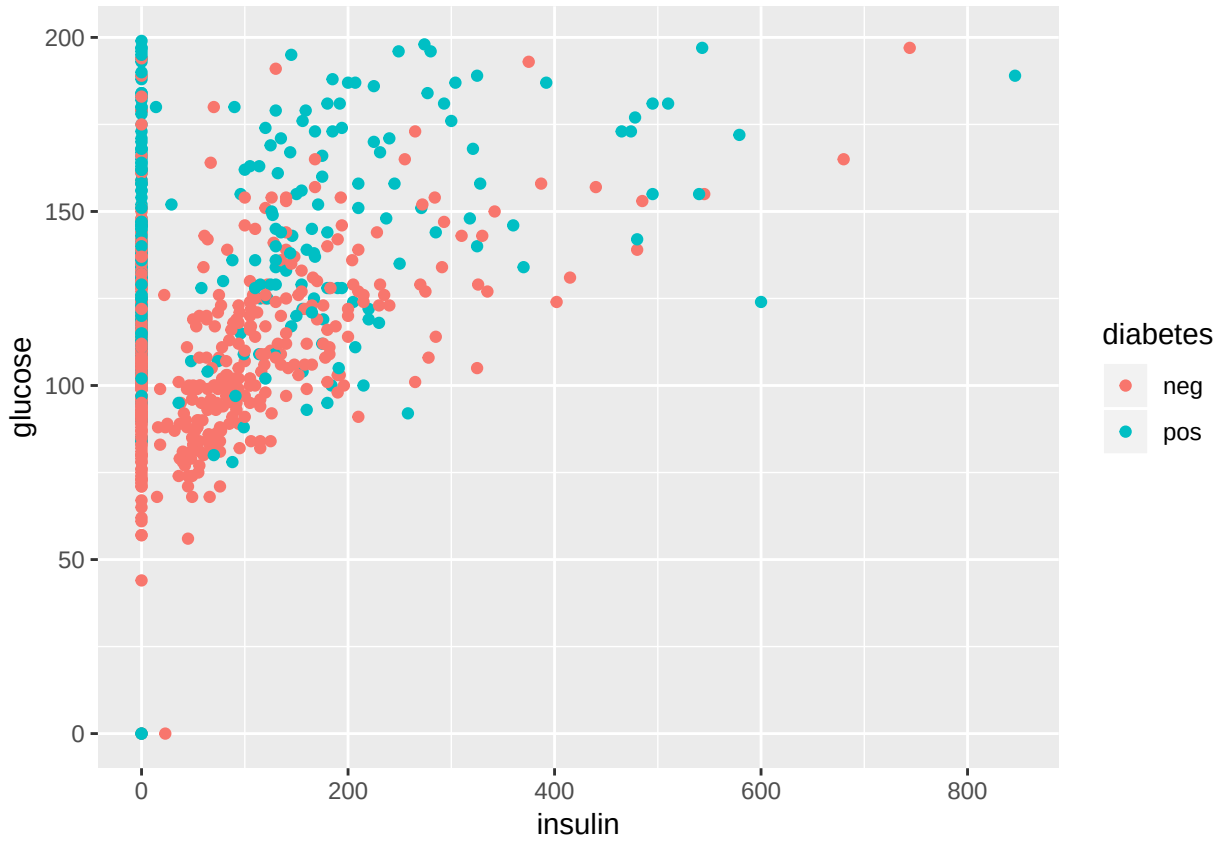
```
# box and whisker plots for each attribute
featurePlot(x=x, y=y, plot="box")
```



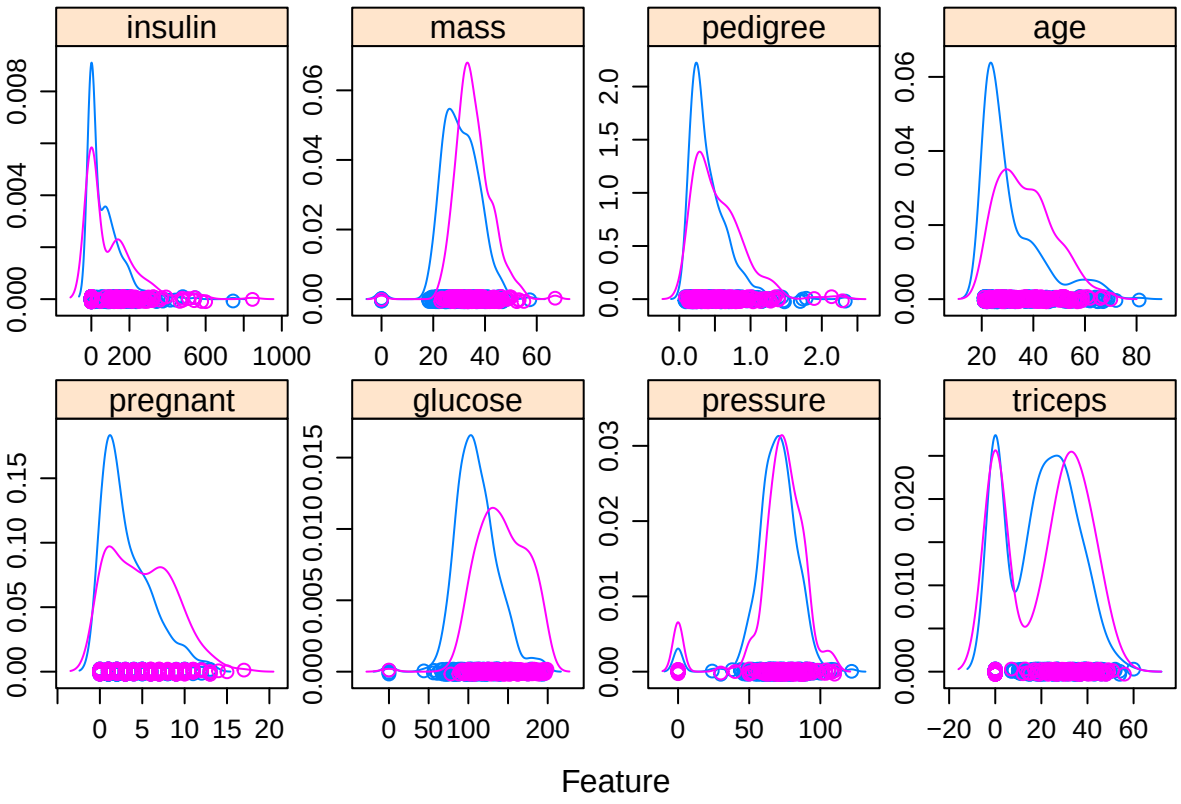
```
ggplot(PimaIndiansDiabetes, aes(insulin,mass, color = diabetes)) + geom_point()
```


```
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)
```

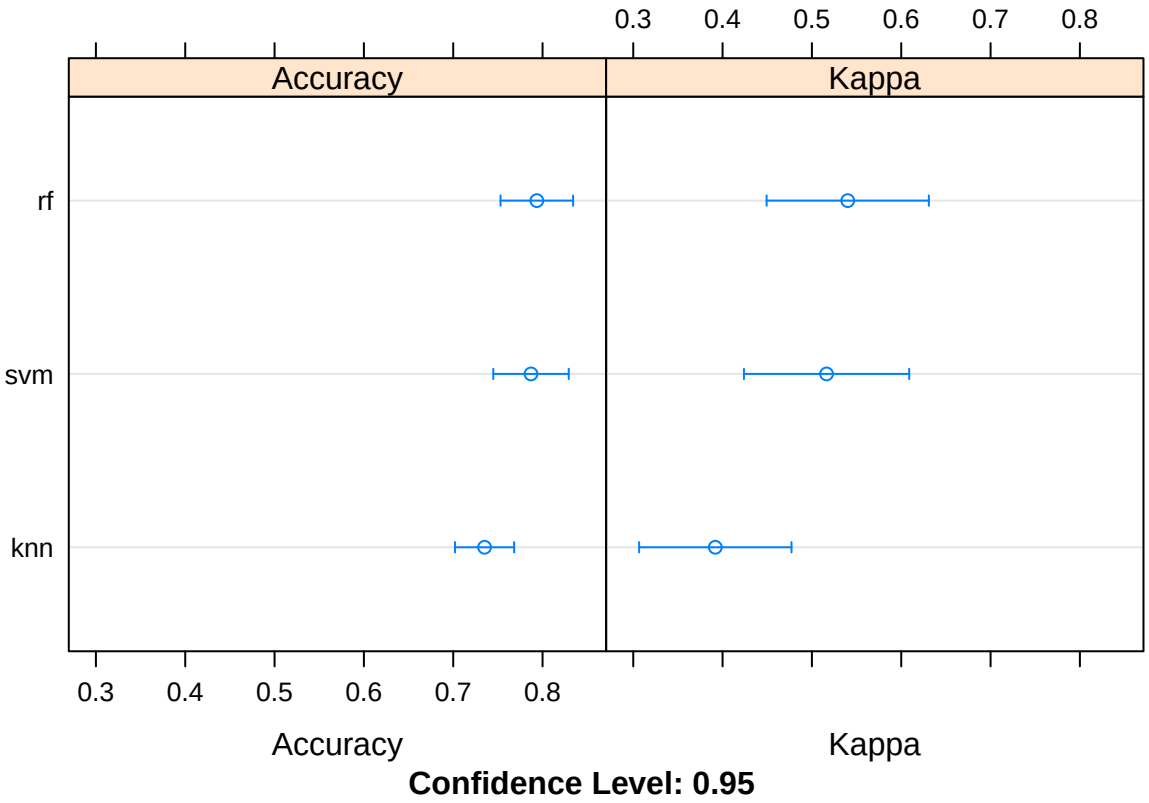
Comparison of the Classification Algorithms

```
# 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 0.6612903 0.7125859 0.7295082 0.7350873 0.7673850 0.8032787    0
## svm 0.6612903 0.7741935 0.7868852 0.7870703 0.8163670 0.8709677    0
## rf  0.6774194 0.7652697 0.7968006 0.7936013 0.8387097 0.8524590    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## knn 0.1968912 0.3405823 0.3833313 0.3918680 0.4866384 0.5542022    0
## svm 0.2677165 0.4628713 0.5055416 0.5164152 0.5954488 0.7061611    0
## rf  0.3231441 0.4616766 0.5437285 0.5401147 0.6582350 0.6743697    0
```

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



Insights from the best model

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

```
## Random Forest
##
## 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:
##
##   mtry  Accuracy   Kappa
##   2     0.7871232  0.5179103
##   5     0.7870968  0.5205188
##   8     0.7936013  0.5401147
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 8.
```

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

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction neg pos
##        neg  83  26
##        pos  17  27
##
##               Accuracy : 0.719
##                 95% CI : (0.6407, 0.7886)
##    No Information Rate : 0.6536
##    P-Value [Acc > NIR] : 0.05152
##
##                  Kappa : 0.3535
##  Mcnemar's Test P-Value : 0.22247
##
##             Sensitivity : 0.8300
##             Specificity : 0.5094
##          Pos Pred Value : 0.7615
##          Neg Pred Value : 0.6136
##              Prevalence : 0.6536
##          Detection Rate : 0.5425
##    Detection Prevalence : 0.7124
##       Balanced Accuracy : 0.6697
##
##        '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
```

```r
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

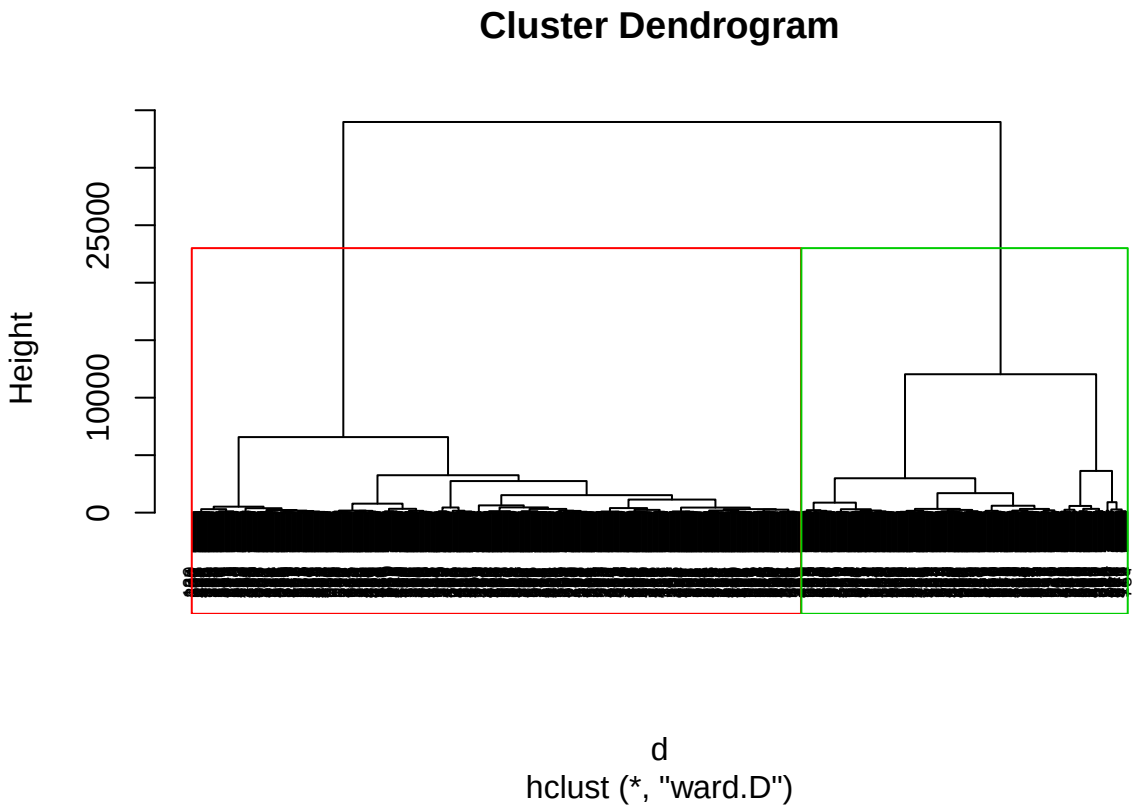
```r
# Cut tree into 4 groups
sub_grp <- cutree(fit.ha, k = 2)

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

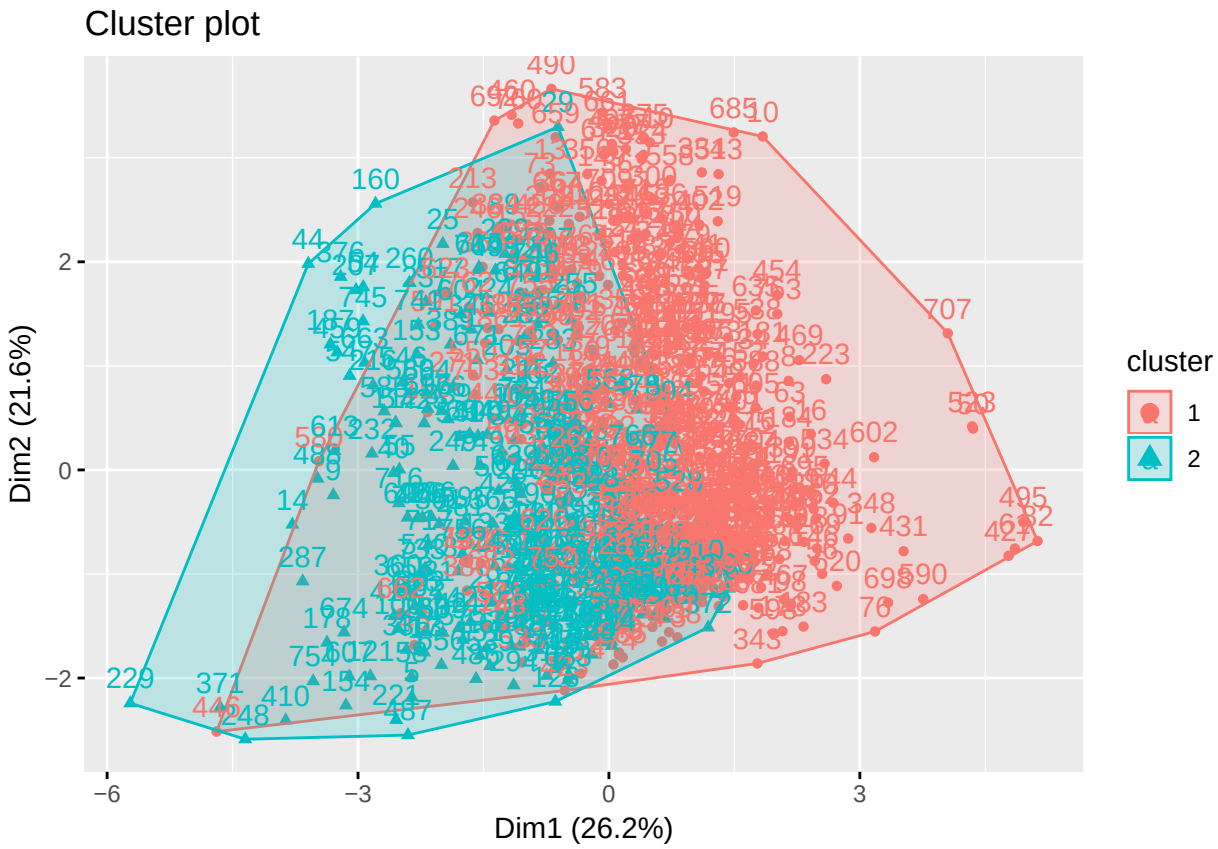
```
## sub_grp
##   1   2
## 500 268
```

```
## sub_grp
```

```r
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))
```

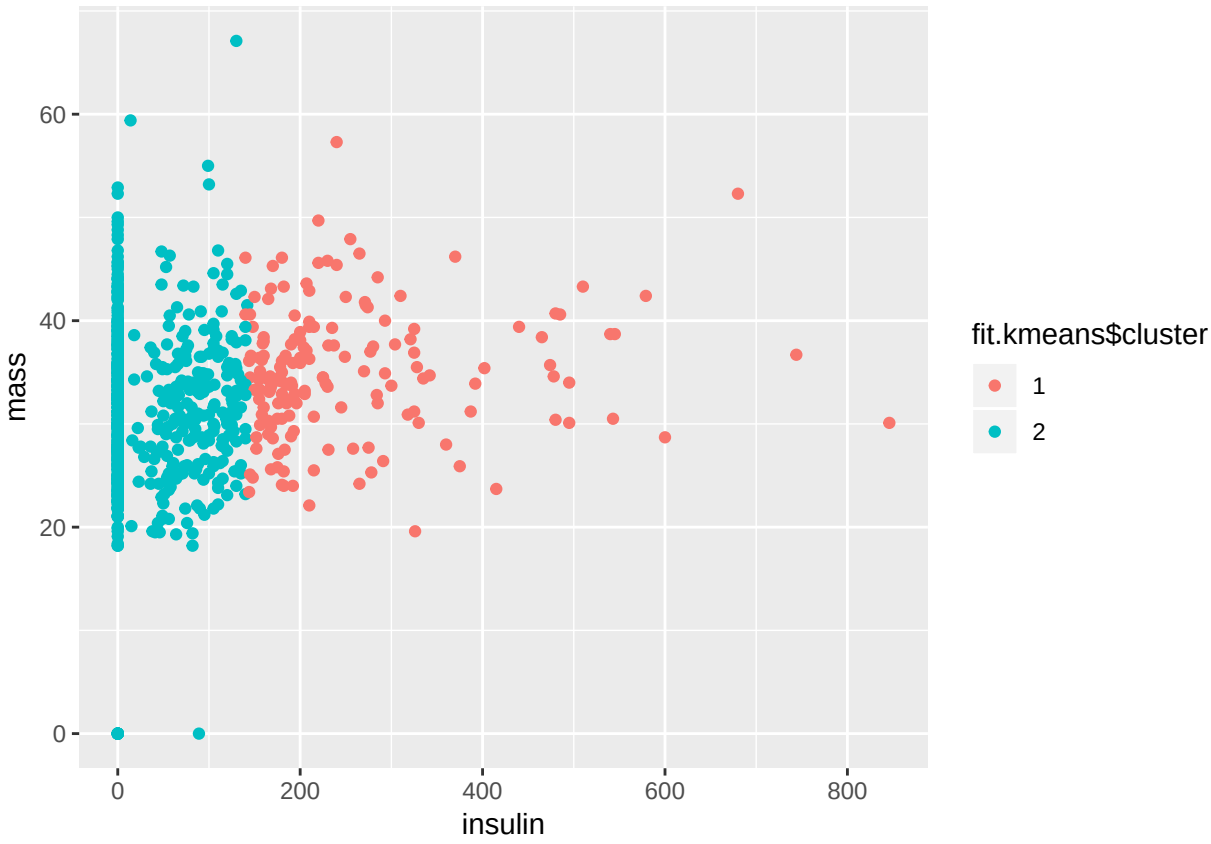


Getting insights from K-Means Clustering

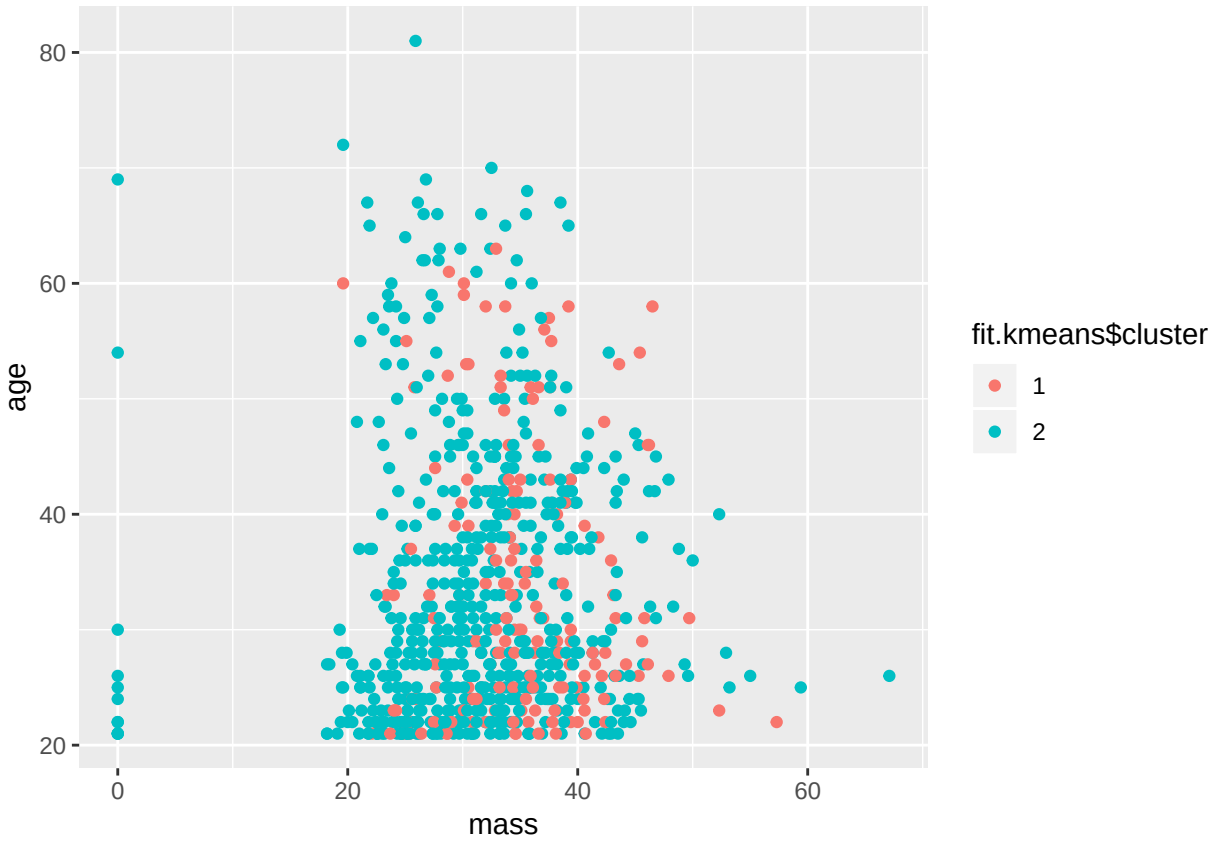
```
table(fit.kmeans$cluster, PimaIndiansDiabetes$diabetes)
```

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

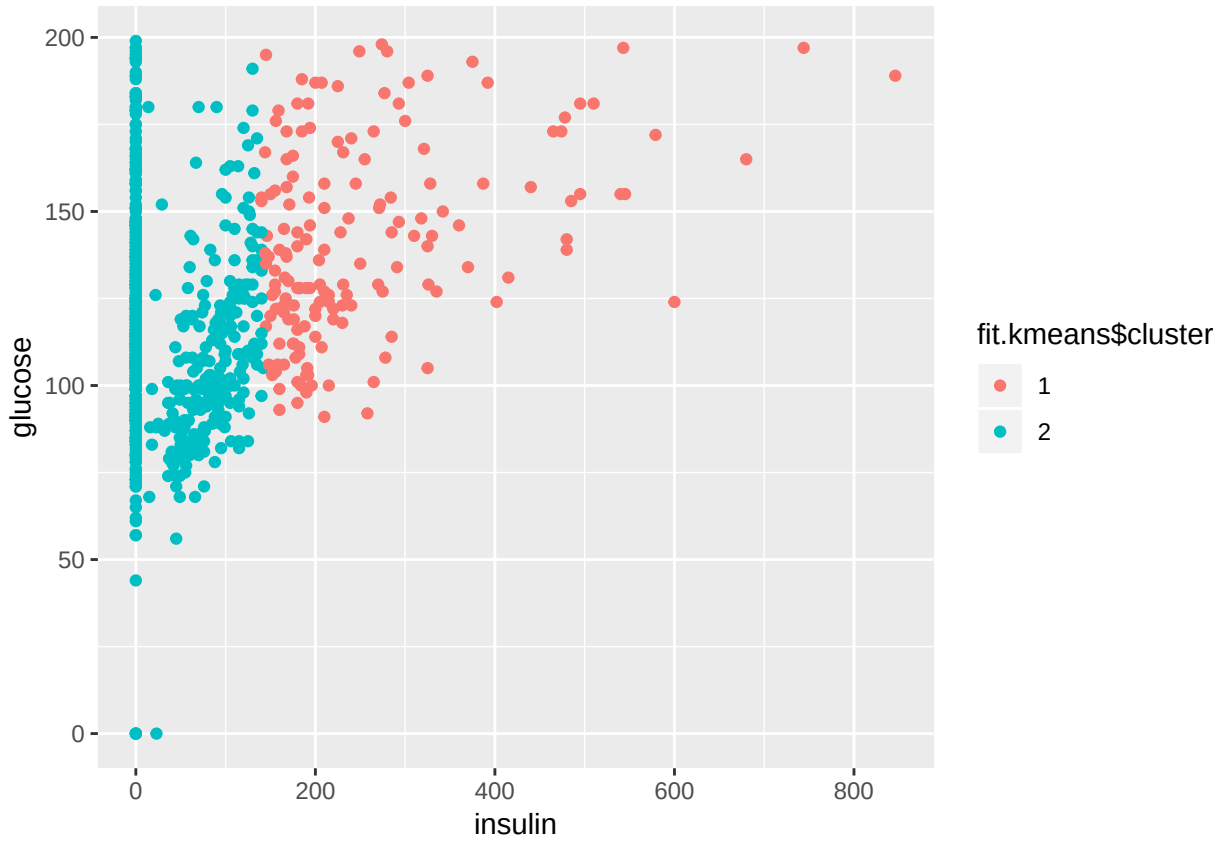
```
fit.kmeans$cluster <- as.factor(fit.kmeans$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, mass, color = fit.kmeans$cluster)) + geom_point()
```




```
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()
```

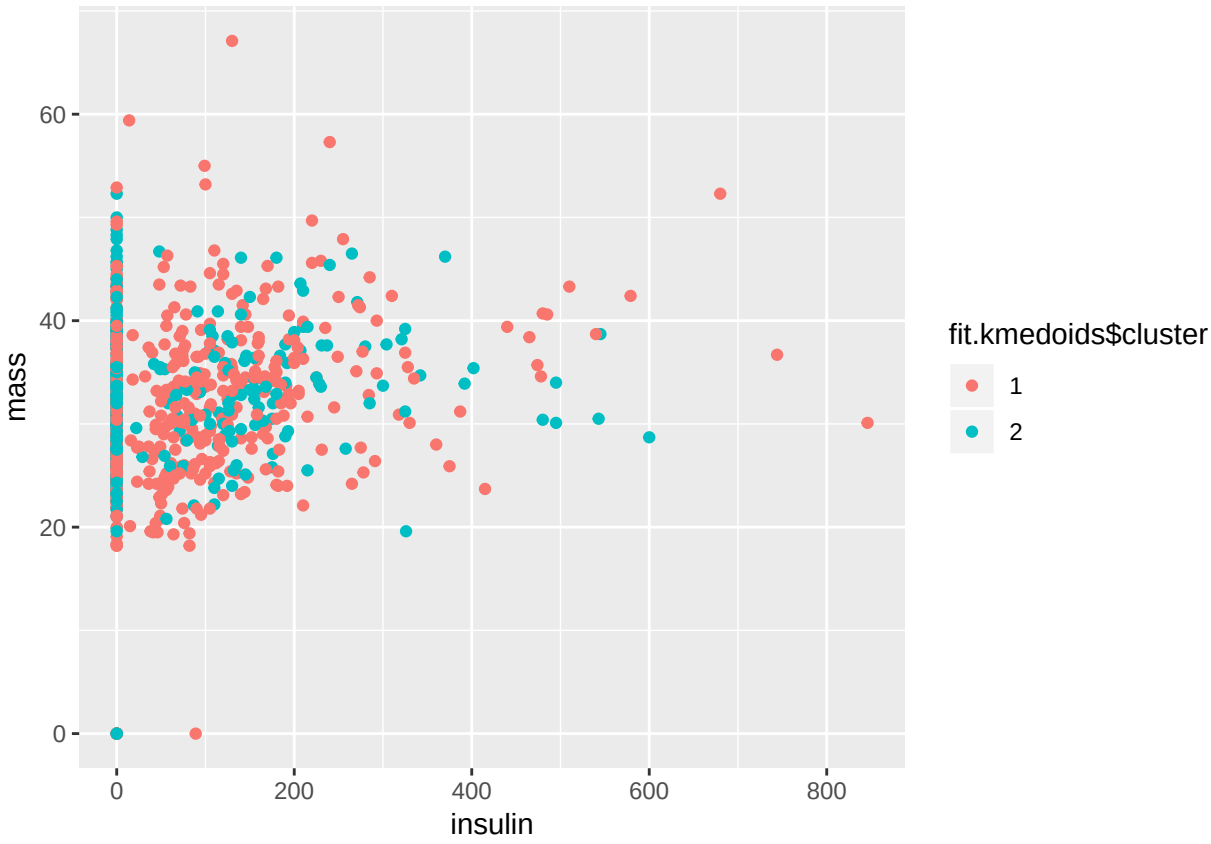


Getting insights from K-Medoids Clustering

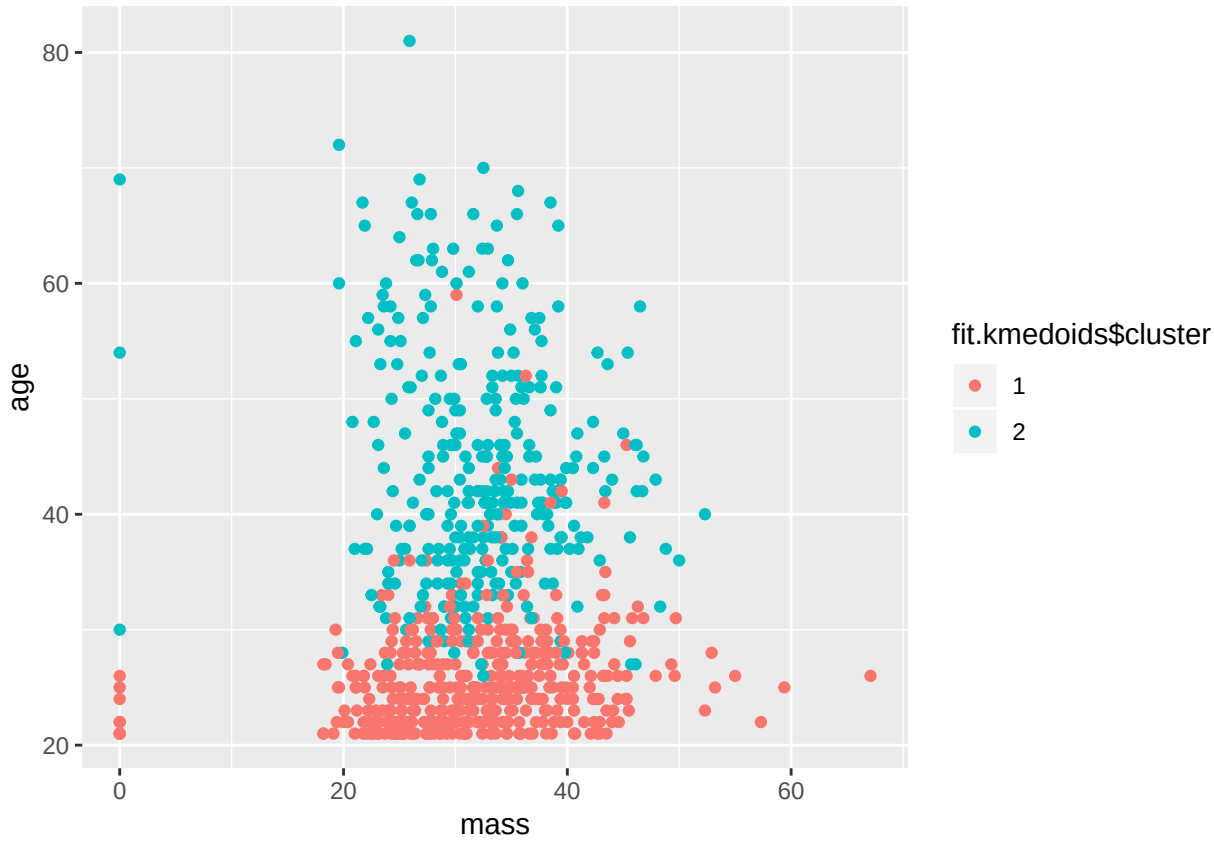
```
(fastiris <- table(fit.kmedoids$cluster, PimaIndiansDiabetes[,9]))
```

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

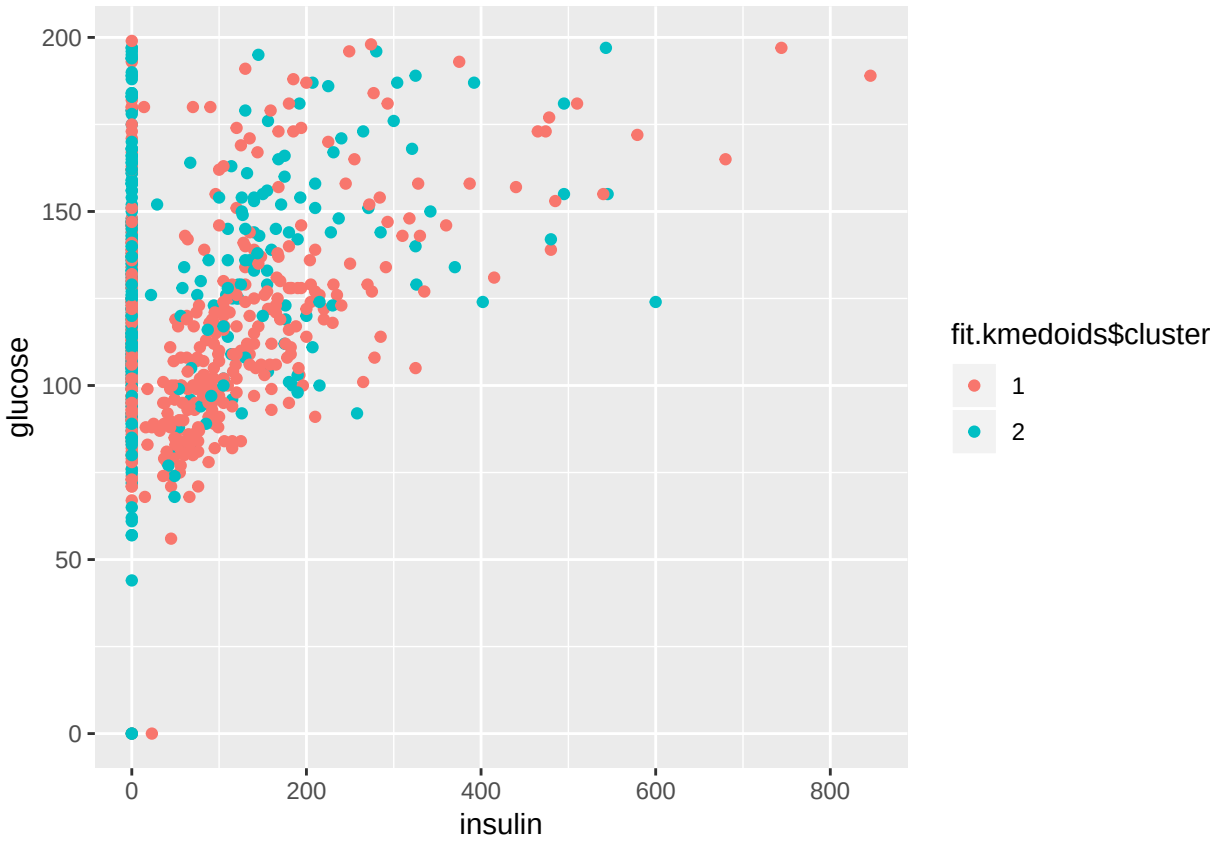
```
fit.kmedoids$cluster <- as.factor(fit.kmedoids$cluster)
ggplot(PimaIndiansDiabetes, aes(insulin, mass, color = fit.kmedoids$cluster)) + geom_point()
```




```
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()
```



Conclusion

With better accuracy and kappa measures, Random Forest 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.