DA, CA, Cluster analysis

Pei-Yu Chen

2024-10-28

Table of contents

DA

LDA (Linear Discriminant Analysis)

資料集: iris

```r
library(MASS)
data(iris)
#    (X)    (y)
X <- iris[, 1:4]   #
y <- iris[, 5]     #
#
set.seed(123)
train_index <- sample(1:nrow(iris), 0.7 * nrow(iris))   #   70%
X_train <- X[train_index, ]
y_train <- y[train_index]
X_test <- X[-train_index, ]
y_test <- y[-train_index]
#   lda   LDA
```

```
lda_model <- lda(y_train ~ ., data = data.frame(X_train, y_train))
#
predictions <- predict(lda_model, newdata = data.frame(X_test))$class
#
confusion_matrix <- table(Predicted = predictions, Actual = y_test)
print("Confusion Matrix:")
```

```
[1] "Confusion Matrix:"
```

```
print(confusion_matrix)
```

```
            Actual
Predicted    setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         17         0
  virginica       0          1        13
```

```
#
accuracy <- mean(predictions == y_test)
cat("Accuracy:", accuracy, "\n")
```

```
Accuracy: 0.9777778
```

QDA (Quadratic Discriminant Analysis)

資料集: wine

```
library(rattle)
```

```
    tibble

    bitops

Warning:   'bitops'   R   4.3.3

Rattle: A free graphical interface for data science with R.
Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
```

```
data(wine)
qda_model <- qda(Type ~ ., data = wine)
qda_pred <- predict(qda_model, wine)
predicted_classes <- qda_pred$class
confusion_matrix <- table(predicted_classes, wine$Type)
print(confusion_matrix)
```

```
predicted_classes  1  2  3
                1 59  1  0
                2  0 70  0
                3  0  0 48
```

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy: ", round(accuracy, 4)))
```

```
[1] "Accuracy:  0.9944"
```

RDA (Regularized Discriminant Analysis)

資料集: iris

```
library(glmnet)
```

```
    Matrix


  'Matrix'

     'package:bitops':

    %&%
```

```
Loaded glmnet 4.1-8
```

```
data(iris)
#     (X)    (y)
X <- as.matrix(iris[, 1:4])  #
y <- as.factor(iris[, 5])     #
#
set.seed(123)
train_index <- sample(1:nrow(X), 0.7 * nrow(X))  #   70%
```

```
X_train <- X[train_index, ]
y_train <- y[train_index]
X_test <- X[-train_index, ]
y_test <- y[-train_index]
#   glmnet   RDA
model <- glmnet(X_train, y_train, alpha = 0.5, family = "multinomial")
#         lambda
cv_model <- cv.glmnet(X_train, y_train, alpha = 0.5, family = "multinomial")
#     lambda
best_lambda <- cv_model$lambda.min
cat("Best lambda:", best_lambda, "\n")
```

```
Best lambda: 0.0002451586
```

```
#     lambda
predictions <- predict(model, s = best_lambda, newx = X_test, type = "class")
#
confusion_matrix <- table(Predicted = predictions, Actual = y_test)
print(confusion_matrix)
```

```
            Actual
Predicted    setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         17         0
  virginica       0          1        13
```

```
#
accuracy <- mean(predictions == y_test)
cat("Accuracy:", accuracy, "\n")
```

```
Accuracy: 0.9777778
```

CA

SCA (Simple Correspondence Analysis)

資料集: HairEyeColor

```
#
dev.new()

#   ca package
library(ca)
```

```
#   HairEyeColor
hair_eye <- margin.table(HairEyeColor, 1:2)

#
sca_model <- ca(hair_eye)

print(sca_model)
```

```
 Principal inertias (eigenvalues):
           1        2        3       
Value      0.208773 0.022227 0.002598
Percentage 89.37%   9.52%    1.11%   


 Rows:
             Black     Brown       Red    Blond
Mass      0.182432  0.483108  0.119932 0.214527
ChiDist   0.551192  0.159461  0.354770 0.838397
Inertia   0.055425  0.012284  0.015095 0.150793
Dim. 1   -1.104277 -0.324463 -0.283473 1.828229
Dim. 2    1.440917 -0.219111 -2.144015 0.466706


 Columns:
             Brown     Blue     Hazel     Green
Mass      0.371622 0.363176  0.157095  0.108108
ChiDist   0.500487 0.553684  0.288654  0.385727
Inertia   0.093086 0.111337  0.013089  0.016085
Dim. 1   -1.077128 1.198061 -0.465286  0.354011
Dim. 2    0.592420 0.556419 -1.122783 -2.274122
```

```
#
plot(sca_model)
```

MCA (Multiple Correspondence Analysis)

資料集: survey

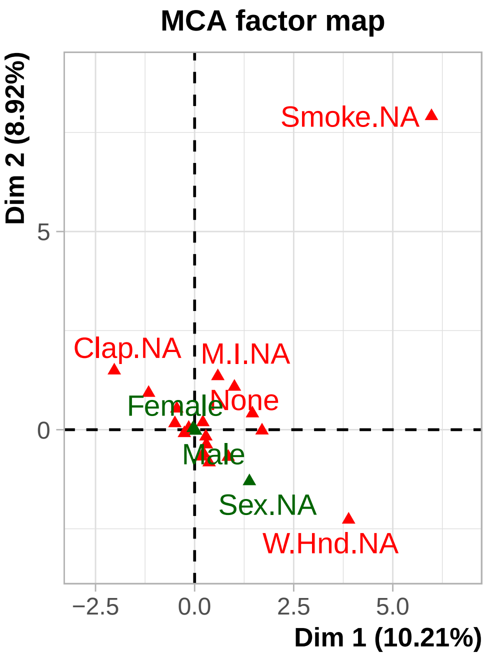
```
library(FactoMineR)
```

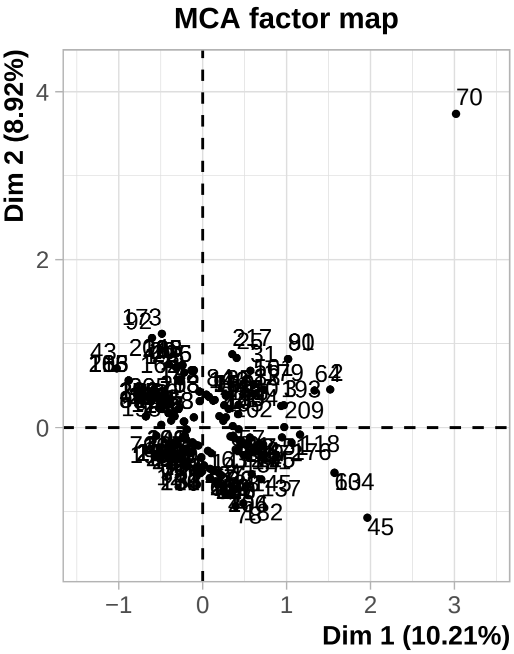
```
Warning:    'FactoMineR'    R    4.3.2
```

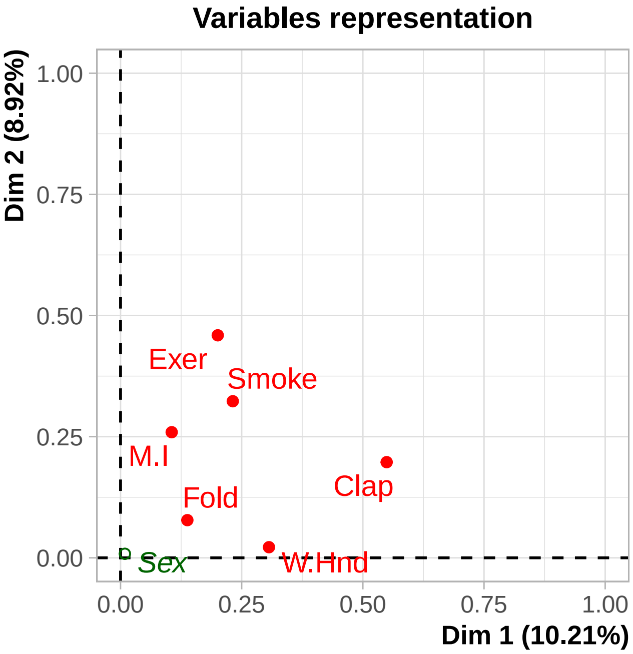
```r
library(ggplot2)
library(ggrepel)
library(MASS)

data(survey)
#       MCA
mca_data <- survey[, c("Sex", "W.Hnd", "Fold", "Clap", "Exer", "Smoke", "M.I")]
#
mca_model <- MCA(mca_data, quali.sup = 1)
```

```
Warning: ggrepel: 16 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



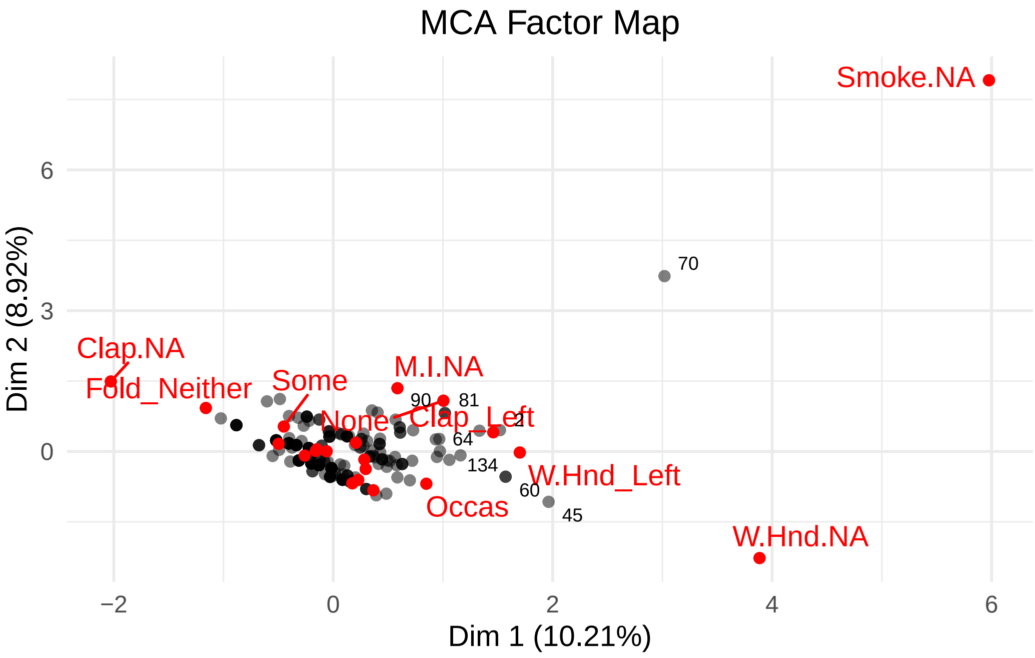




```
#   MCA
mca_vars <- data.frame(mca_model$var$coord, Variable = rownames(mca_model$var$coord))
mca_inds <- data.frame(mca_model$ind$coord, Individual = rownames(mca_model$ind$coord))
#     ggplot2
ggplot() +
  geom_point(data = mca_inds, aes(x = Dim.1, y = Dim.2), color = "black", alpha = 0.5) +  #
  geom_point(data = mca_vars, aes(x = Dim.1, y = Dim.2), color = "red") +  #
  geom_text_repel(data = mca_vars, aes(x = Dim.1, y = Dim.2, label = Variable), color = "red")
  geom_text_repel(data = mca_inds, aes(x = Dim.1, y = Dim.2, label = Individual), size = 2.5,
  labs(title = "MCA Factor Map", x = paste0("Dim 1 (", round(mca_model$eig[1, 2], 2), "%)"),
       y = paste0("Dim 2 (", round(mca_model$eig[2, 2], 2), "%)")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
Warning: ggrepel: 11 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

```
Warning: ggrepel: 229 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



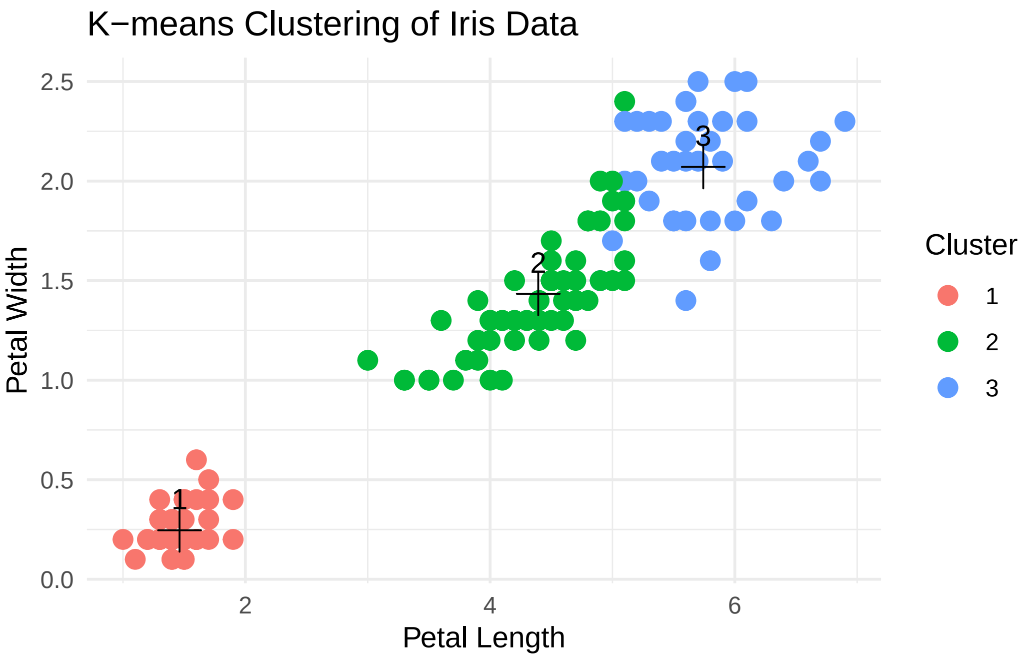
Cluster Analysis

K-means Clustering

資料集: iris

```
#
library(ggplot2)
#   iris
data(iris)
#   Species
iris_data <- iris[, -5]
# K-means     3
```

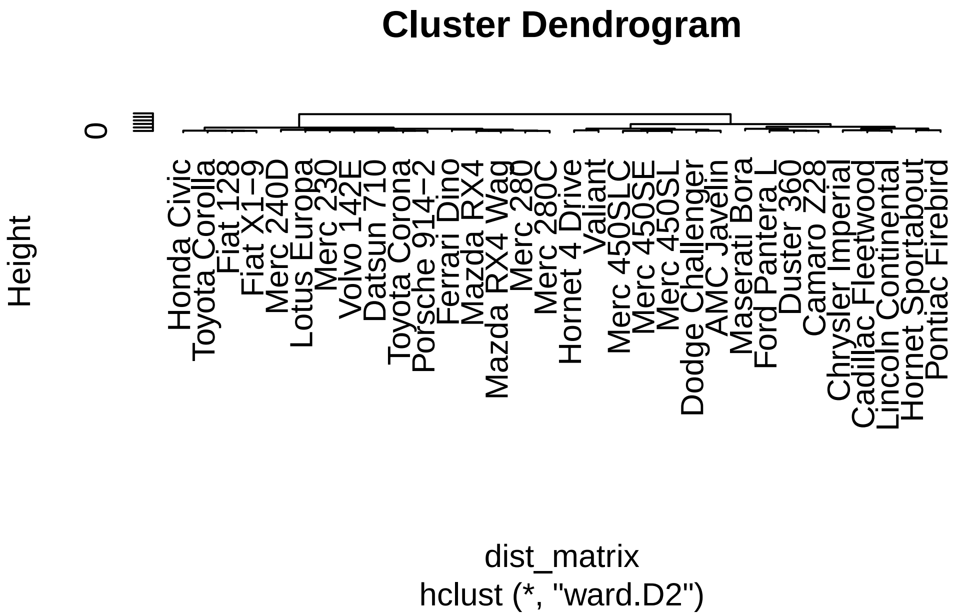
```r
set.seed(123)
kmeans_model <- kmeans(iris_data, centers = 3)
#
iris_data$Cluster <- as.factor(kmeans_model$cluster)
#
centers_df <- as.data.frame(kmeans_model$centers)
centers_df$Cluster <- factor(1:3)
#    K-means
ggplot(iris_data, aes(x = Petal.Length, y = Petal.Width, color = Cluster)) +
  geom_point(size = 3) +
  #
  geom_point(data = centers_df, aes(x = Petal.Length, y = Petal.Width),
             color = "black", size = 5, shape = 3) +
  #
  geom_text(data = centers_df,
            aes(x = Petal.Length, y = Petal.Width, label = Cluster),
            vjust = -1, color = "black") +
  labs(title = "K-means Clustering of Iris Data",
       x = "Petal Length", y = "Petal Width") +
  theme_minimal()
```



Hierarchical Clustering

資料集: mtcars

```r
#   mtcars
data(mtcars)
#
dist_matrix <- dist(mtcars)
#   Ward's
hclust_model <- hclust(dist_matrix, method = "ward.D2")
#
plot(hclust_model)
```



```r
#     4
groups <- cutree(hclust_model, k = 4)
print(groups)  #
```

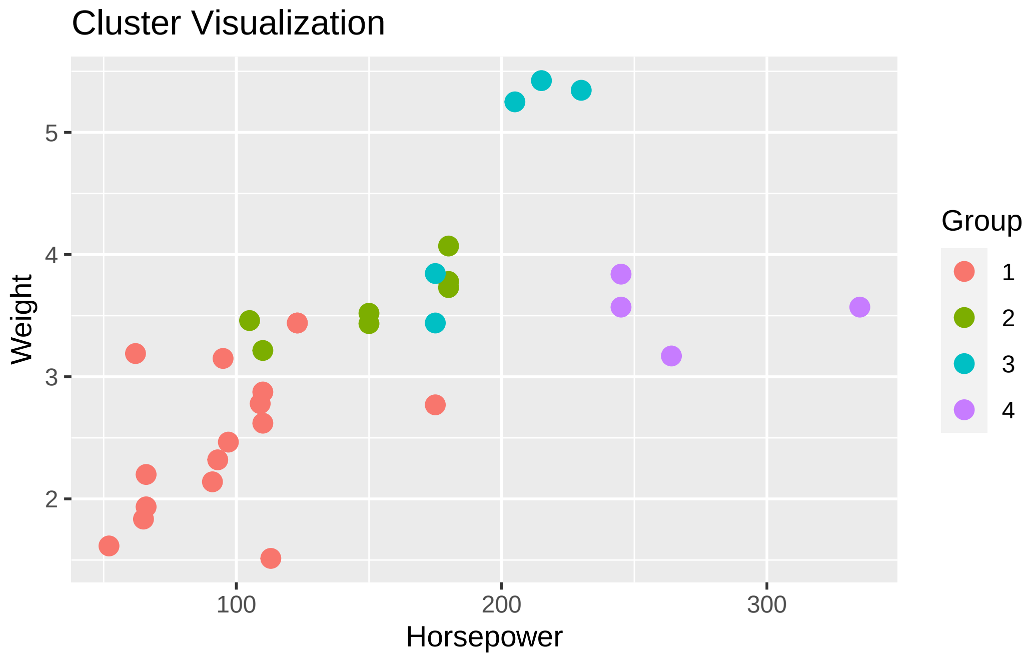
```
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive
                  1                   1                   1                   2
  Hornet Sportabout             Valiant          Duster 360           Merc 240D
                  3                   2                   4                   1
           Merc 230            Merc 280           Merc 280C          Merc 450SE
                  1                   1                   1                   2
         Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
                  2                   2                   3                   3
  Chrysler Imperial            Fiat 128         Honda Civic      Toyota Corolla
                  3                   1                   1                   1
      Toyota Corona    Dodge Challenger         AMC Javelin          Camaro Z28
                  1                   2                   2                   4
   Pontiac Firebird           Fiat X1-9       Porsche 914-2        Lotus Europa
                  3                   1                   1                   1
```

```
     Ford Pantera L        Ferrari Dino       Maserati Bora          Volvo 142E
                  4                   1                   4                   1
```

```
#
mtcars$group <- groups
#
aggregate(mtcars[, -ncol(mtcars)], by = list(Group = mtcars$group), FUN = mean)
```

```
  Group      mpg      cyl     disp       hp     drat       wt     qsec
1     1 24.50000 4.625000 122.2938  96.8750 4.002500 2.518000 18.54312
2     2 17.01429 7.428571 276.0571 150.7143 2.994286 3.601429 18.11857
3     3 14.68000 8.000000 426.4000 200.0000 3.078000 4.660800 17.45800
4     4 14.60000 8.000000 340.5000 272.2500 3.675000 3.537500 15.08750
         vs     am  gear     carb
1 0.7500000 0.6875 4.125 2.437500
2 0.2857143 0.0000 3.000 2.142857
3 0.0000000 0.0000 3.000 3.200000
4 0.0000000 0.5000 4.000 5.000000
```

```
library(ggplot2)
#      hp   wt
ggplot(mtcars, aes(x = hp, y = wt, color = factor(group))) +
  geom_point(size = 3) +
  labs(title = "Cluster Visualization", x = "Horsepower", y = "Weight", color = "Group")
```



DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

資料集: iris

```r
library(dplyr)
```

```
   'dplyr'

     'package:MASS':

    select

     'package:stats':

    filter, lag

     'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(dbscan)
```

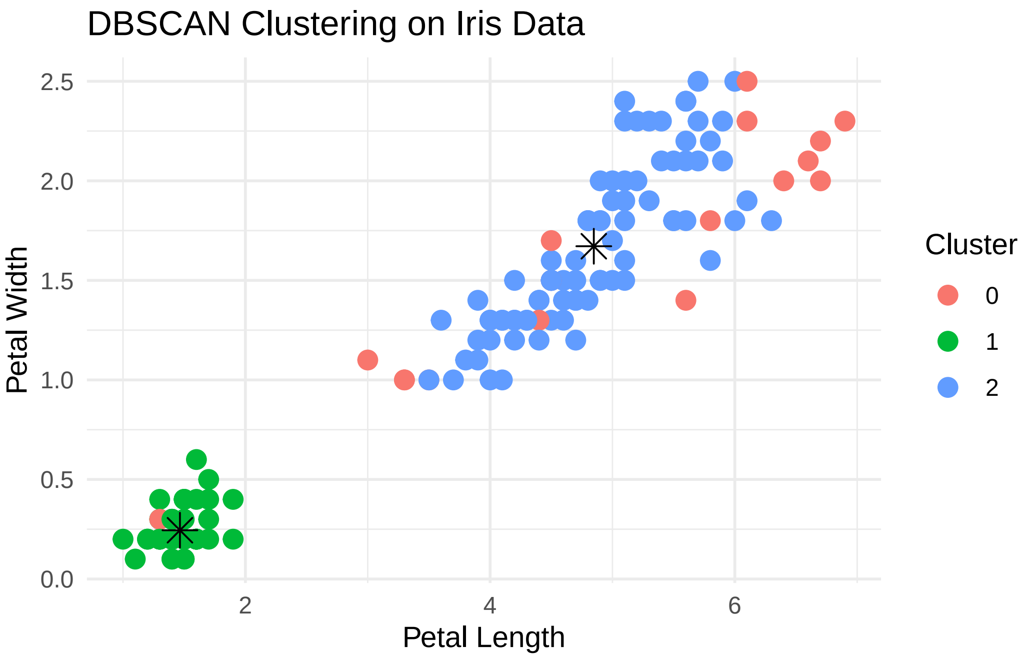
```
Warning:   'dbscan'   R   4.3.3


   'dbscan'

     'package:stats':

    as.dendrogram
```

```r
#   Species
iris_data <- iris[, -5]
#   DBSCAN
dbscan_model <- dbscan(iris_data, eps = 0.5, minPts = 5)
#       dataframe
iris_dbscan <- data.frame(iris_data, Cluster = as.factor(dbscan_model$cluster))
#
centroids <- iris_dbscan %>%
  filter(Cluster != 0) %>%
  group_by(Cluster) %>%
  summarise(
    Petal.Length = mean(Petal.Length),
    Petal.Width = mean(Petal.Width)
  )
#   DBSCAN
ggplot(iris_dbscan, aes(x = Petal.Length, y = Petal.Width, color = Cluster)) +
  geom_point(size = 3) +
  geom_point(data = centroids, aes(x = Petal.Length, y = Petal.Width),
             color = "black", size = 4, shape = 8) +
  labs(title = "DBSCAN Clustering on Iris Data",
       x = "Petal Length", y = "Petal Width") +
  theme_minimal() +
  theme(legend.position = "right")
```


DBSCAN Clustering on Iris Data
Petal Width
Petal Length
Cluster