Clustering on Images

Isaac Baguisa

2025-03-20

K-means

- Run K-means on each dataset with k = number of types to determine if type can be recovered
- Tune for the optimal number of clusters
- Try K-means++
- Clustering results – visualization (interesting pairs of features), CH plots

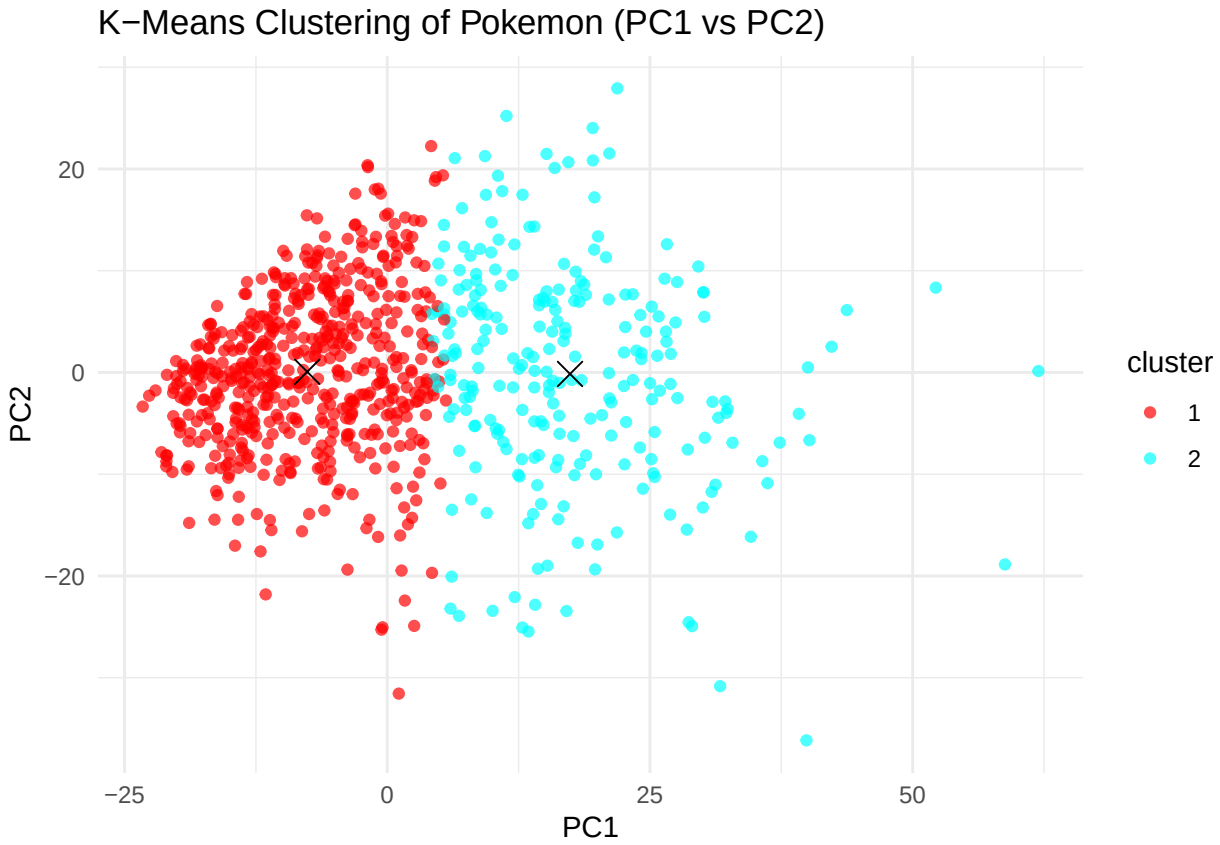
```
library(cluster)
library(factoextra)
library(ggplot2)
library(ggfortify)
library(tidyverse)
library(VIM)
library(gridExtra)
load("../Data/pokemon.RData")
load("../Data/dr_pokemon2.RData")
```

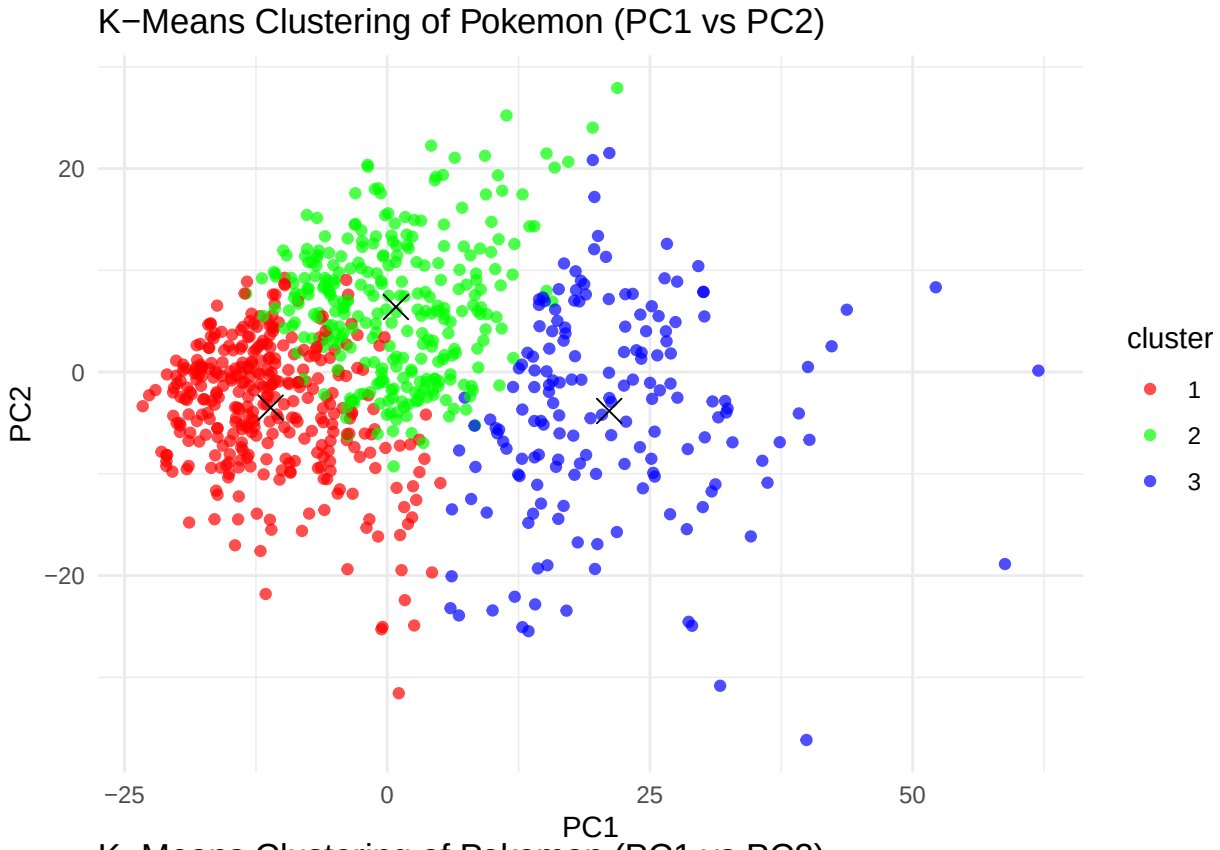
```
pca_data <- dr_images[,-1]  # Remove image_path column
k_types <- length(unique(stats$type1))
```

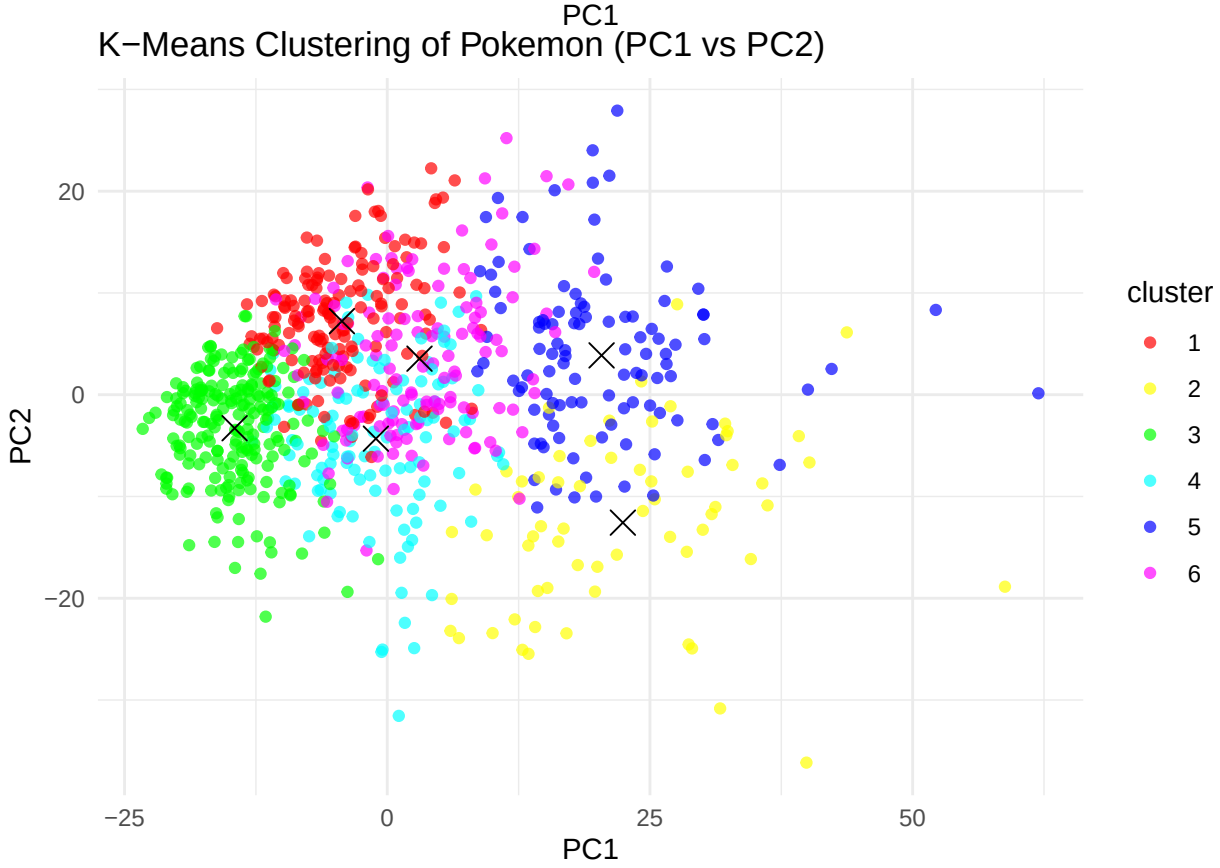
Visualize different K values on image data

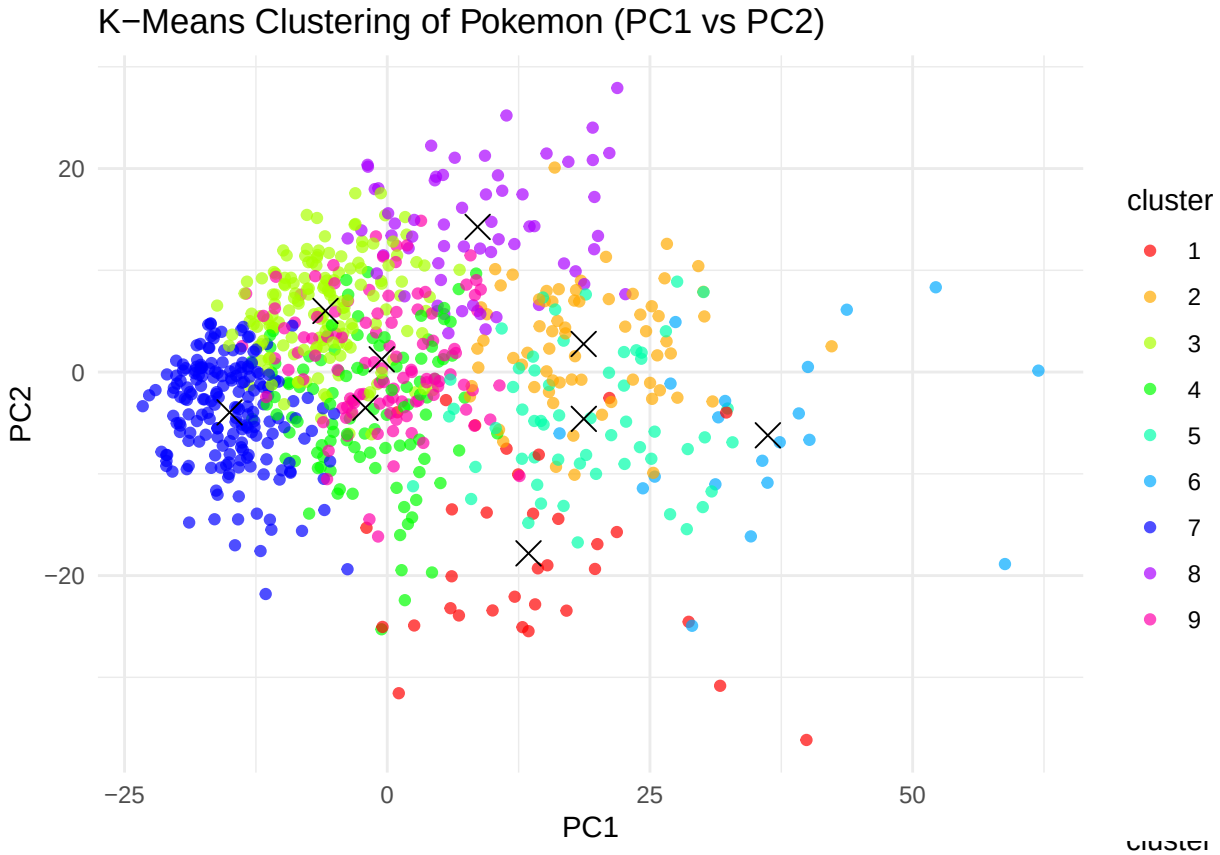
```
# Helper function lecture 7
scatterplot = function(X, M, cluster, label = FALSE){
  X_df <- data.frame(X, cluster = as.factor(cluster))
  M_df <- data.frame(M)

  if (length(unique(cluster)) == 1) {
    plt <- ggplot(X_df, aes(x = PC1, y = PC2)) +
      geom_point() +
      geom_point(data = M_df, aes(x = PC1, y = PC2), shape = 4, size = 4, color = "red") +
      labs(title = "Scatterplot of Pokemon Clusters")

    if (label) {
      plt <- plt + geom_text(aes(label = stats$name), nudge_x = 0.1, size = 3)
    }
    return(plt)
  }
  else {
    ggplot(X_df, aes(x = PC1, y = PC2, color = cluster)) +
      geom_point(alpha = 0.7) +
      geom_point(data = M_df, aes(x = PC1, y = PC2), shape = 4, size = 4, color = "black") +
      scale_color_manual(values = rainbow(length(unique(cluster)))) +
```

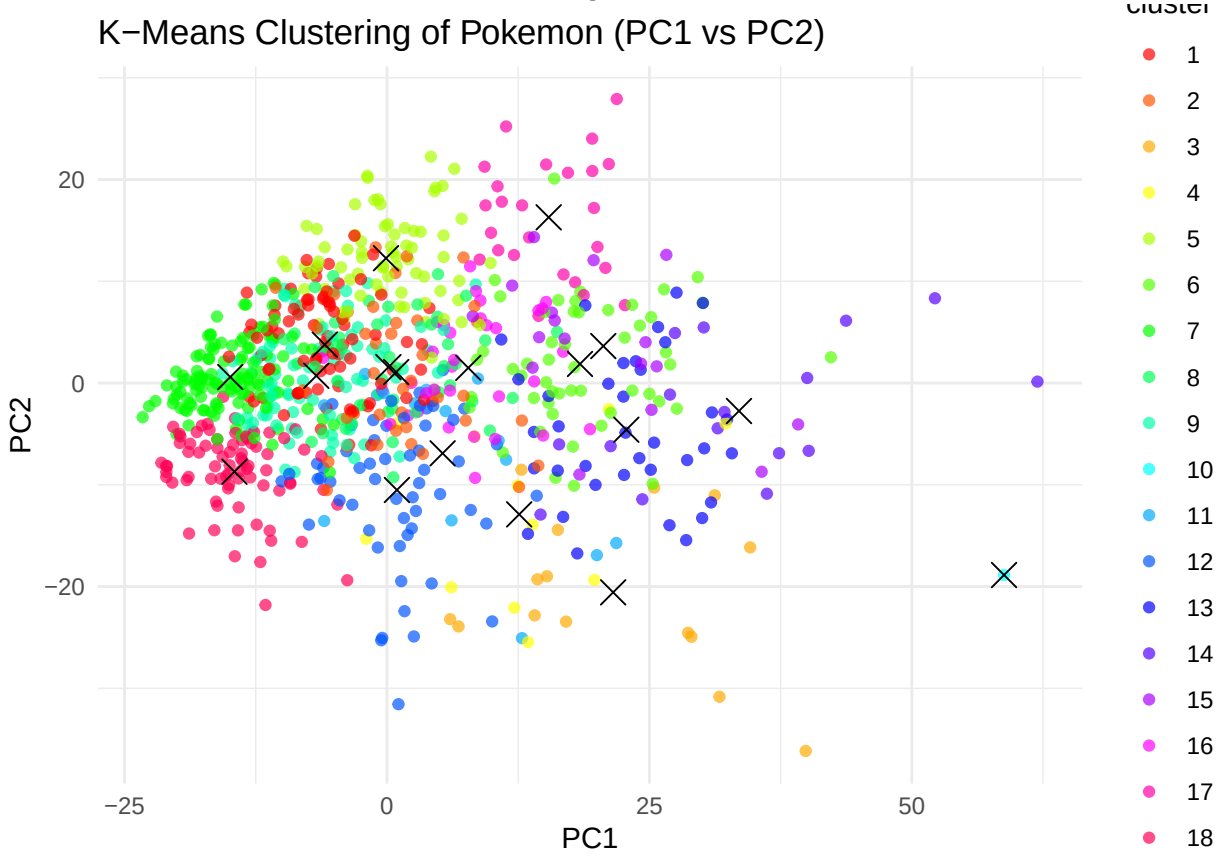
```
      theme_minimal() +
      labs(title = "K-Means Clustering of Pokemon (PC1 vs PC2)", x = "PC1", y = "PC2") +
      theme(legend.position = "right")
  }
}

ks <- c(2, 3, 6, 9, 18)

# Fix layout later
for(iter in ks){
  kmeans_imgs <- kmeans(pca_data, centers = iter, nstart = 25)
  print(scatterplot(pca_data, kmeans_imgs$centers, kmeans_imgs$cluster))
}
```



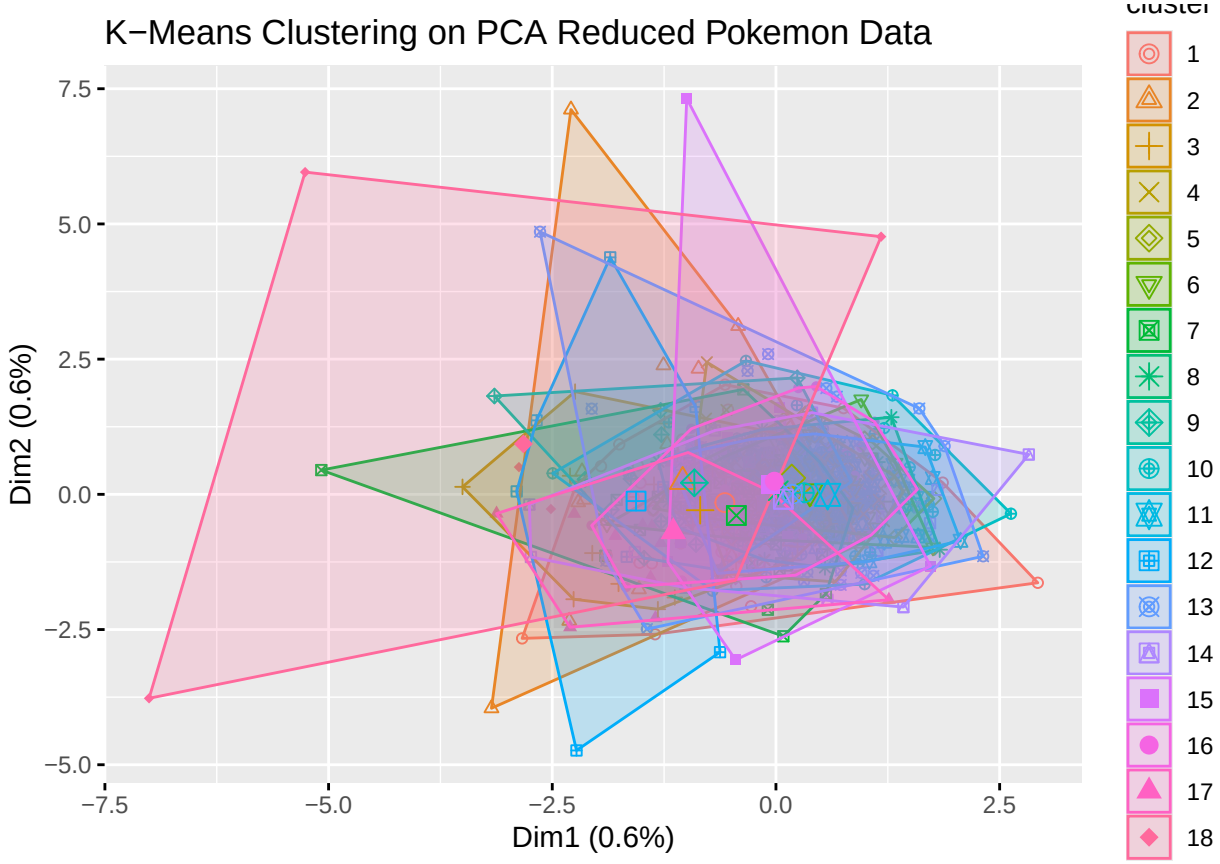

K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
PC1
20
0
−20
−25
0
25
50
cluster
1
2
3


K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
PC1
20
0
−20
−25
0
25
50
cluster
1
2
3
4
5
6


K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
PC1
20
0
−20
−25
0
25
50
cluster
1
2
3
4
5
6
7
8
9

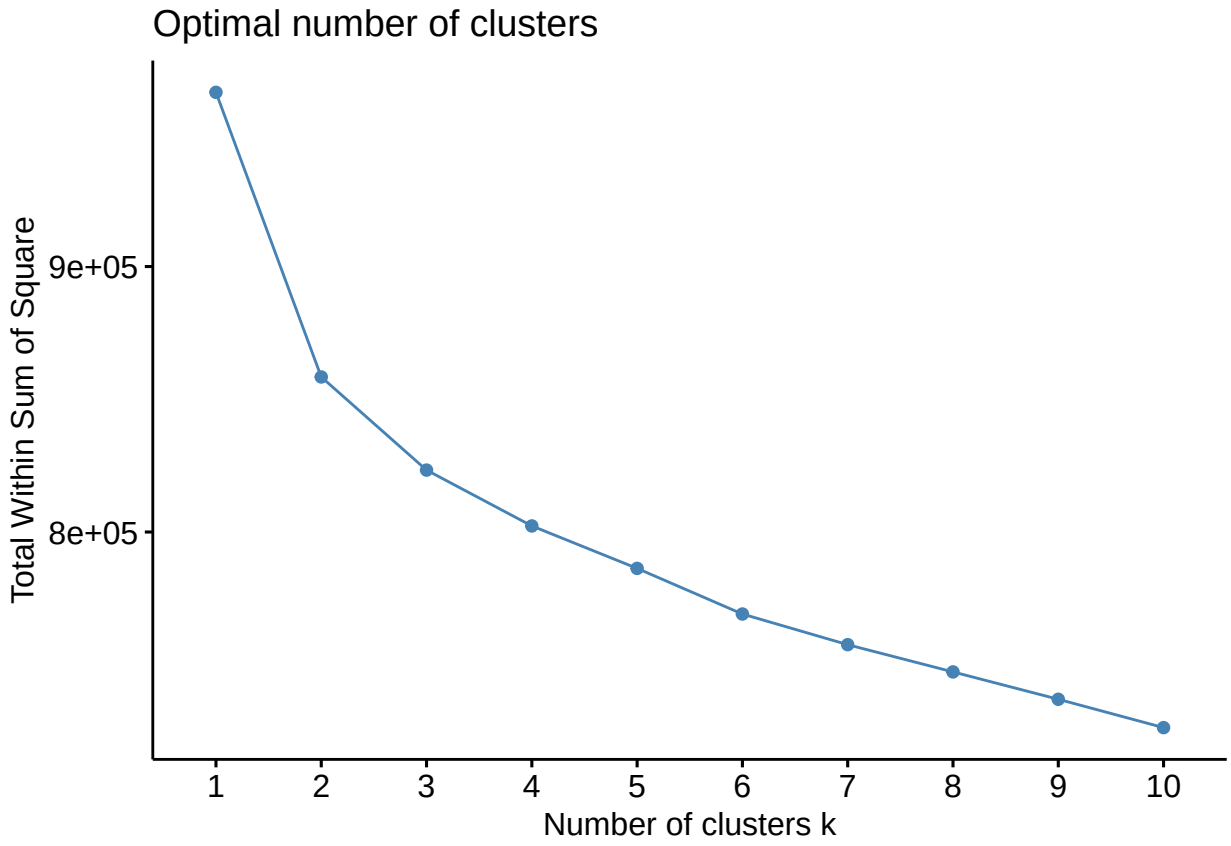

K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
PC1
20
0
−20
−25
0
25
50
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

```r
kmeans_imgs <- kmeans(pca_data, centers = k_types, nstart = 25)
dr_images$cluster_kmeans <- factor(kmeans_imgs$cluster)
fviz_cluster(kmeans_imgs, data = pca_data, geom = "point", ellipse.type = "convex") +
  ggtitle("K-Means Clustering on PCA Reduced Pokemon Data")
```



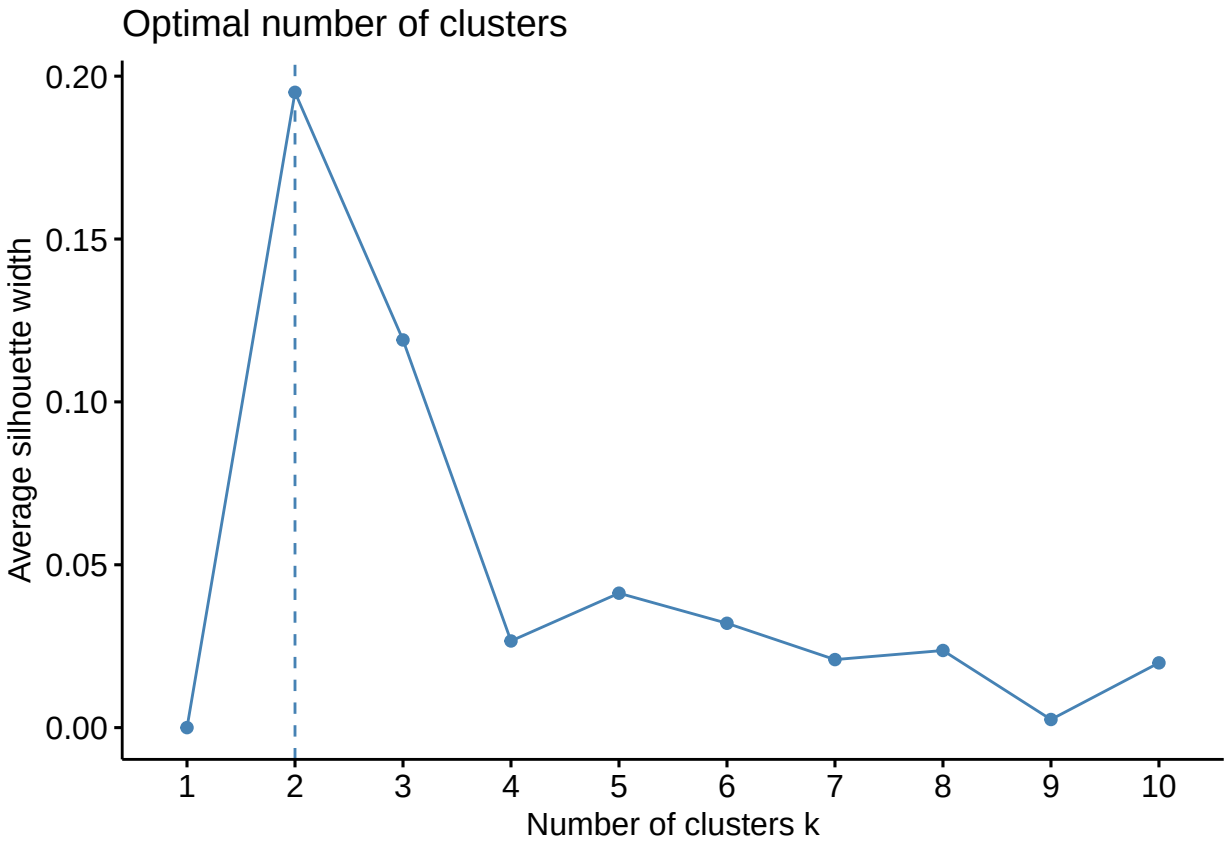
Tune for optimal k

```r
fviz_nbclust(pca_data, kmeans, method = "wss")
```



Elbow method: Optimal K = 2 or 3

```
fviz_nbclust(pca_data, kmeans, method = "silhouette")
```

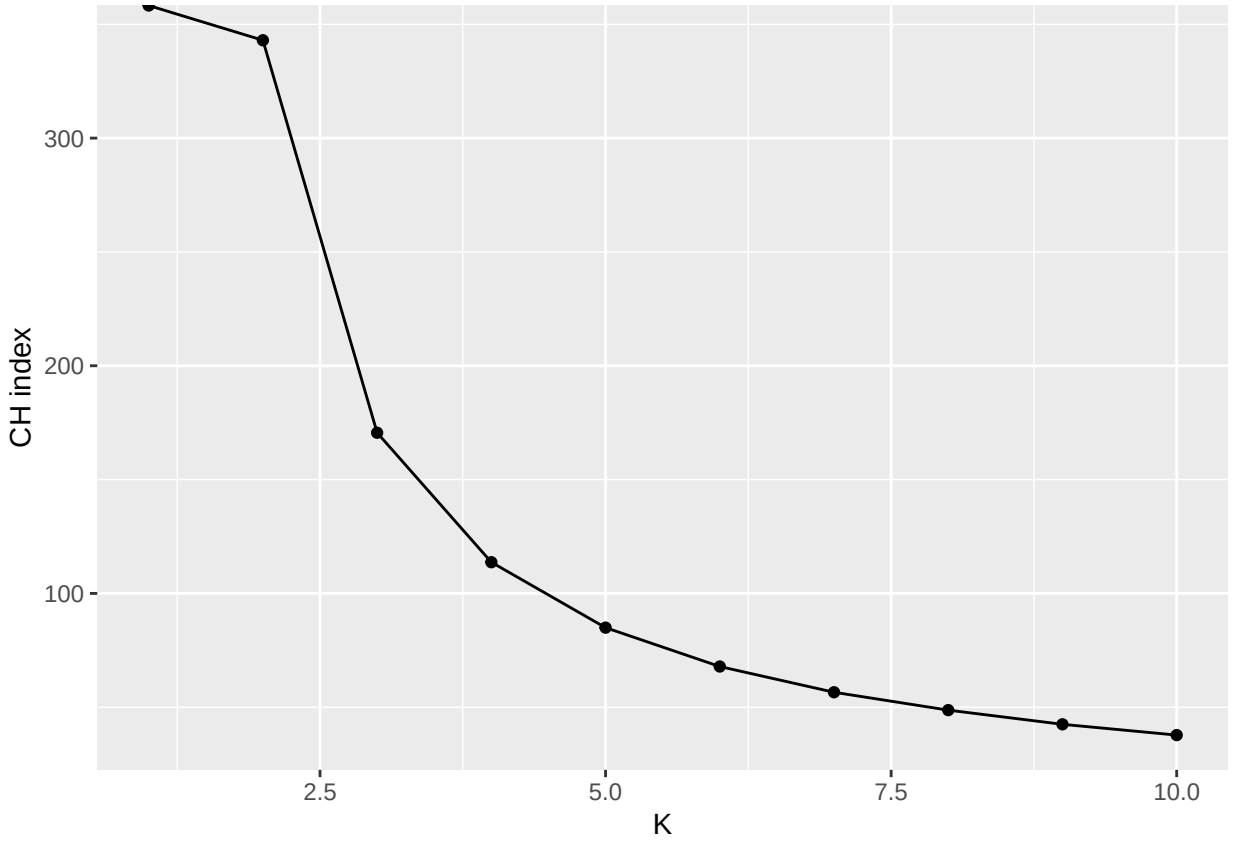


Silhouette method: Optimal K = 2

CH index

```
CHs = c()
Ks = seq(1, 10, 1)

for(K in Ks){
  KM = kmeans(pca_data, centers = k_types, nstart = 25)

  # Between-cluster sum of squares
  B = KM$betweenss

  # Within-cluster sum of squares
  W = KM$tot.withinss

  # Number of data points
  n = nrow(pca_data)

  # Calculate the Calinski-Harabasz index
  CH = (B / (K - 1)) / (W / (n - K))

  # Append the CH index for the current K to the list
  CHs = c(CHs, CH)
}

df = data.frame(K = Ks, CH = CHs)
ggplot(df, aes(K, CH)) +
  geom_point() +
```

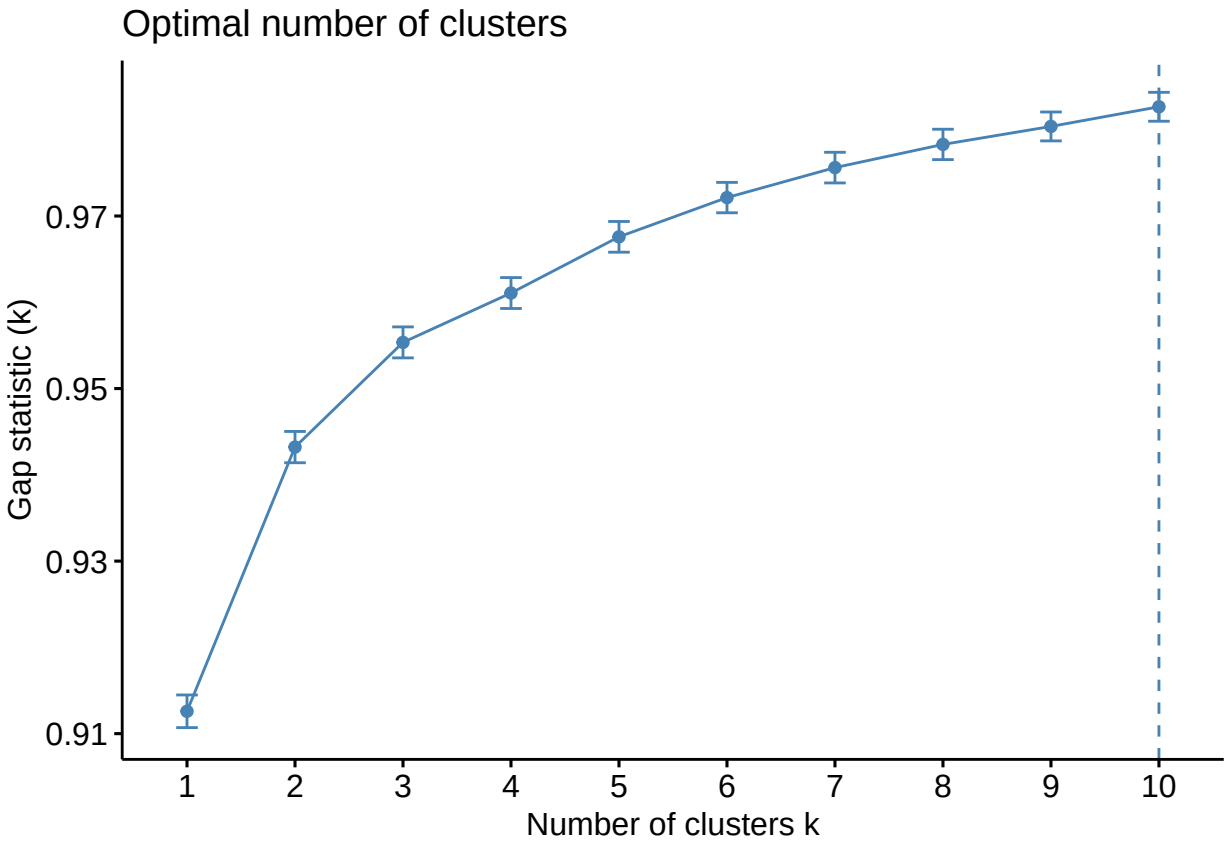
```
  geom_line() +
  ylab("CH index")
```



CH Index: Optimal K = 1

Gap statistics

```
gapstat_img = clusGap(pca_data, FUN = kmeans, nstart = 50, K.max = 10, B = 50)
fviz_gap_stat(gapstat_img, maxSE = list(method = "Tibs2001SEmax", SE.factor = 1))
```



Gap statistics: Optimal K = 10

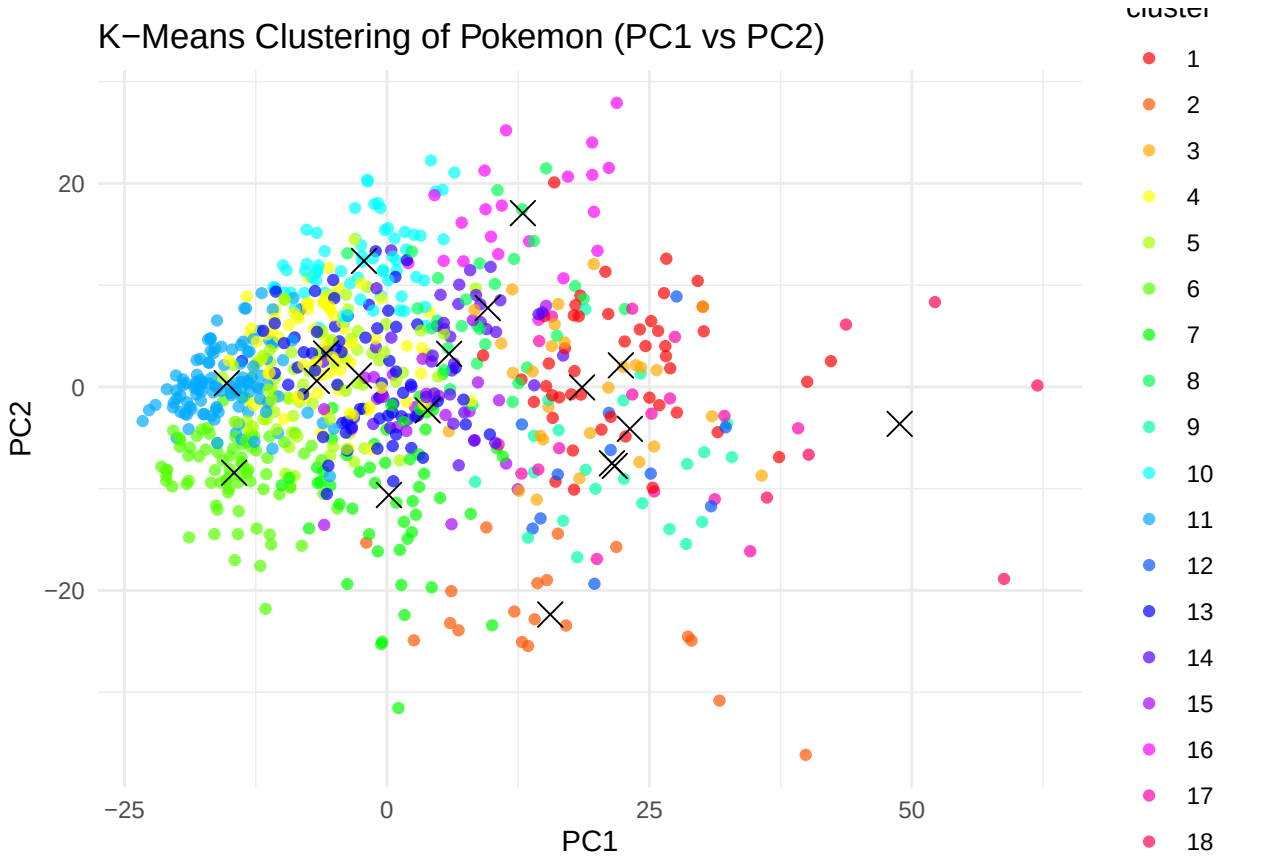
Compare K-means and K-means++

K-means with K-means++ initialization

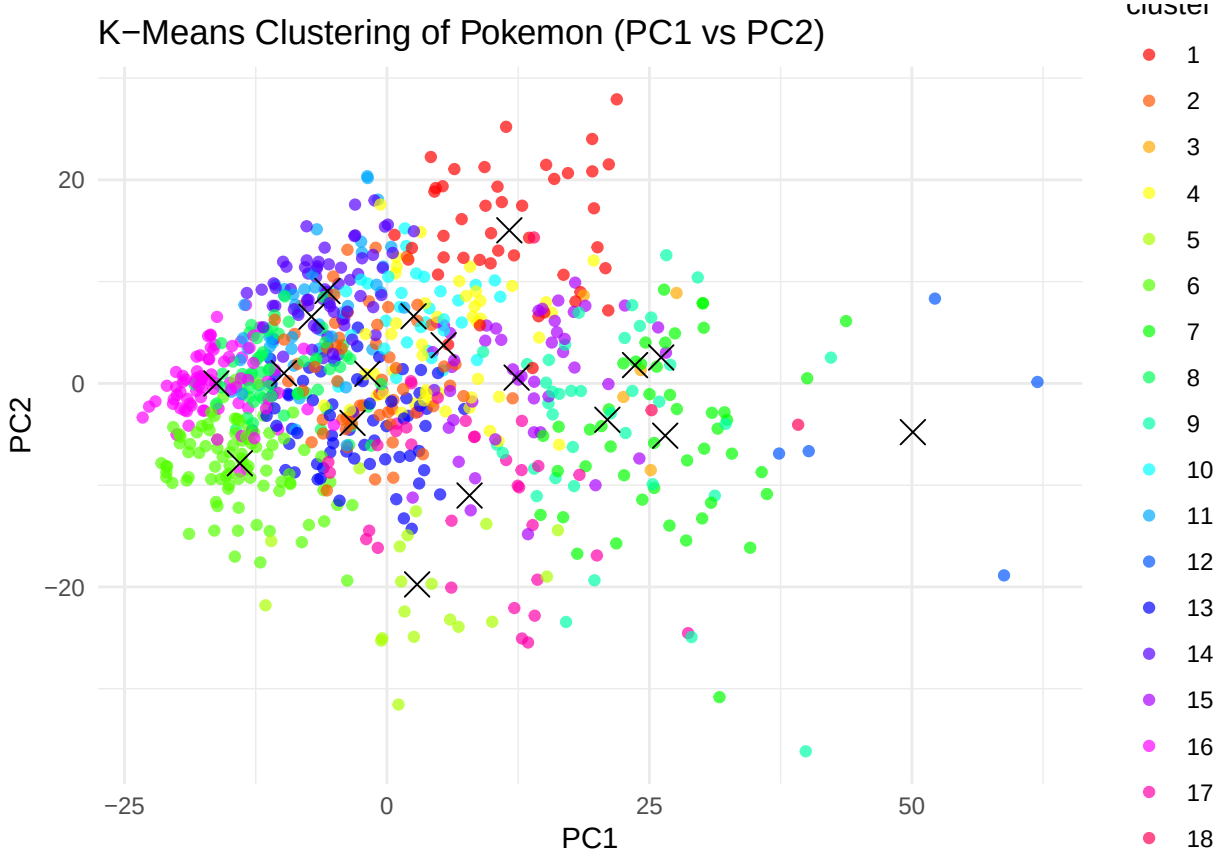
```
set.seed(2101)

#Ref: lecture 7
# Randomly select the first centroid
n <- nrow(pca_data)
M <- pca_data[sample(1:n, 1), , drop = FALSE]

# Select remaining k-1 centroids using weighted probability
for (i in 2:k_types) {
  # Compute distance from each point to the nearest centroid
  D <- as.matrix(dist(rbind(M, pca_data)))[2:(n+1), 1]

  # Probability for each point to be chosen as the next centroid
  P <- D^2 / sum(D^2)

  # Select the next centroid based on P
  pick <- sample(1:n, 1, prob = P)
  M <- rbind(M, pca_data[pick, , drop = FALSE])
}

kmeans_pp <- kmeans(pca_data, centers = M, algorithm = "Lloyd")
```

Compare K-means and K-means++ plots

```
scatterplot(pca_data, kmeans_imgs$centers, kmeans_imgs$cluster)
```

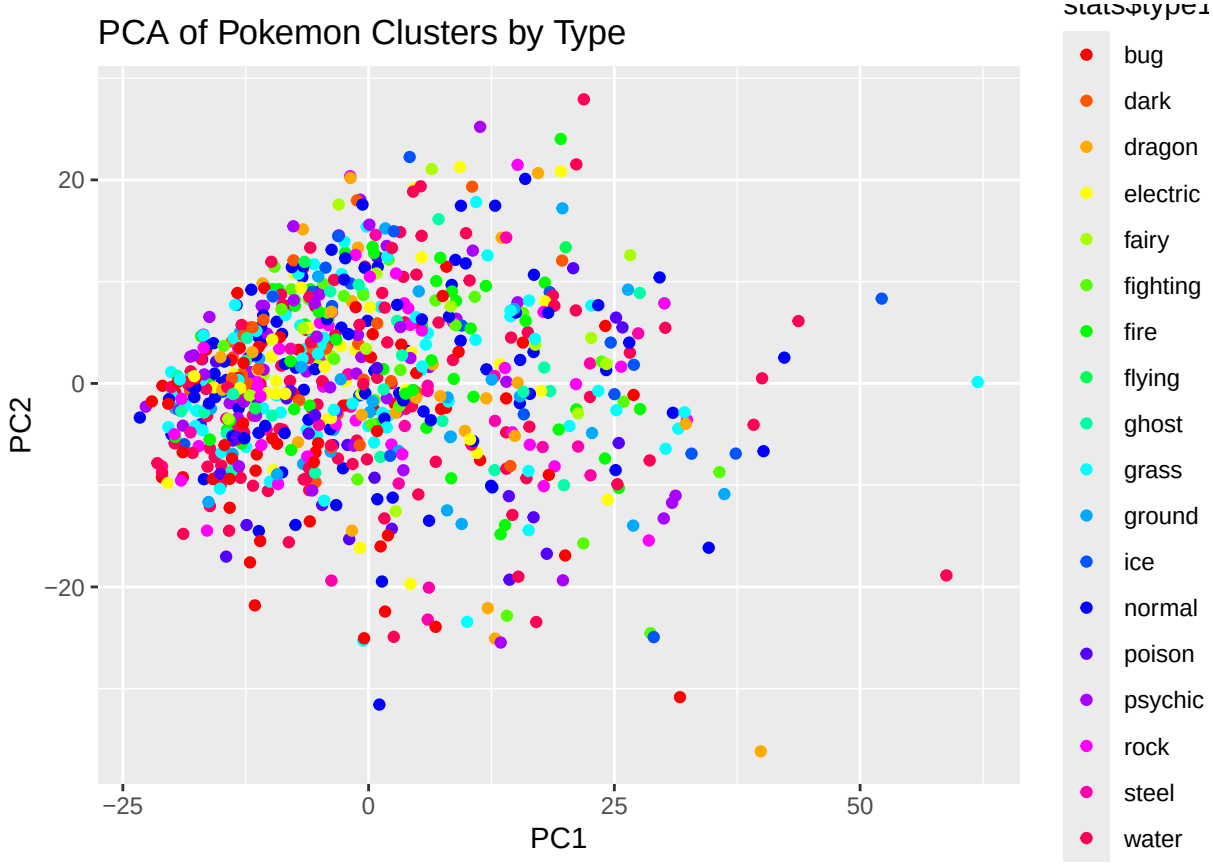



```
scatterplot(pca_data, kmeans_pp$centers, kmeans_pp$cluster)
```



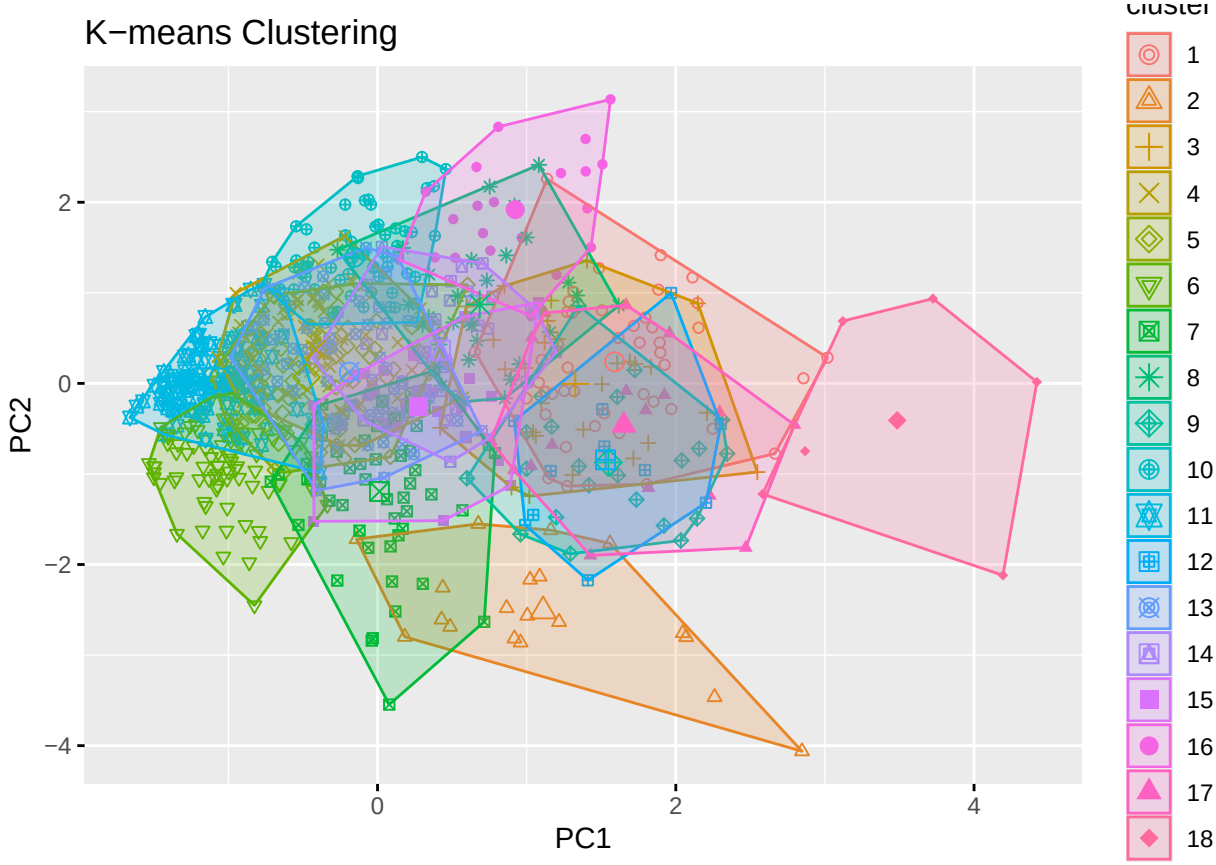
Visualize cluster on actual labels

```
ggplot(data = pca_data, aes(x = PC1, y = PC2, color = stats$type1)) +
  geom_point() +
  labs(title = "PCA of Pokemon Clusters by Type",
       x = "PC1", y = "PC2") +
  scale_color_manual(values = rainbow(length(unique(stats$type1))))
```



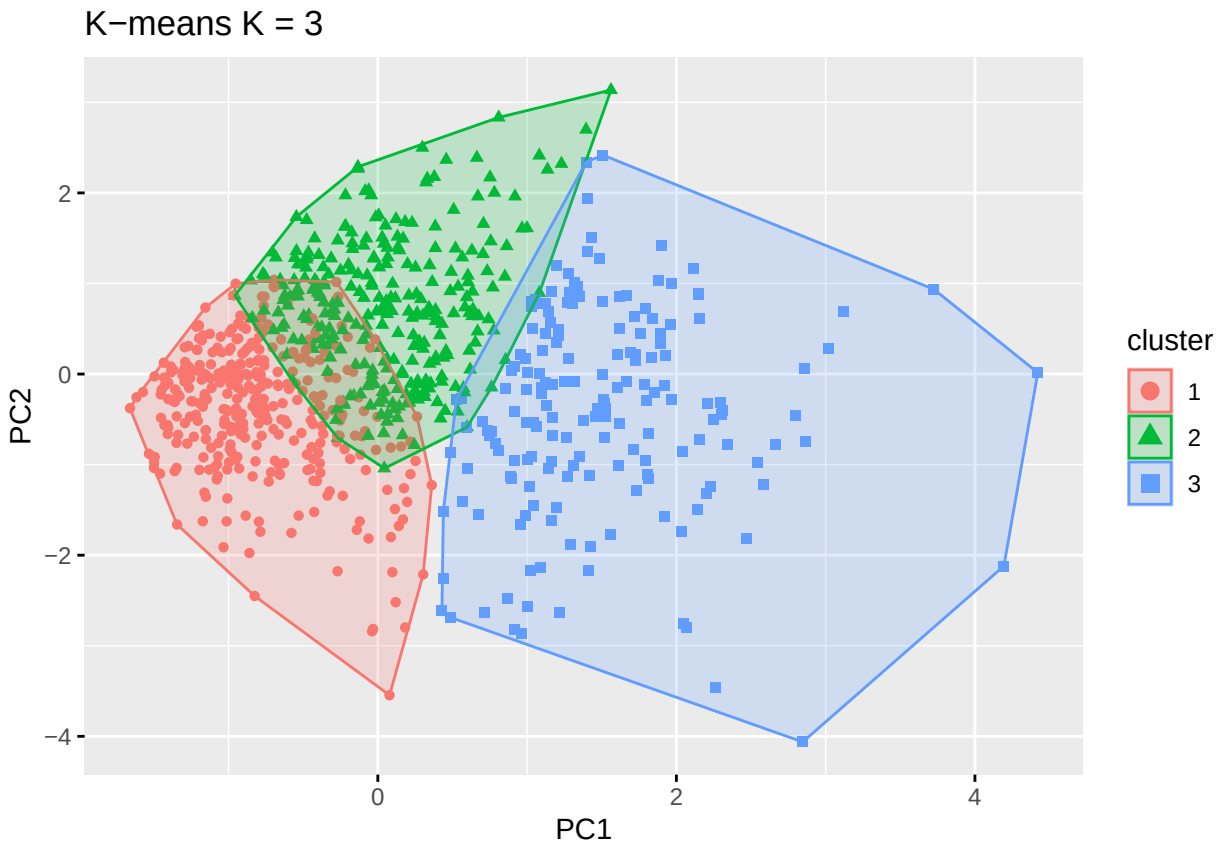
K-means results (using the first two principal components for visualization)

```
fviz_cluster(kmeans_imgs, data = pca_data[,1:2], geom = "point",
             main = "K-means Clustering")
```



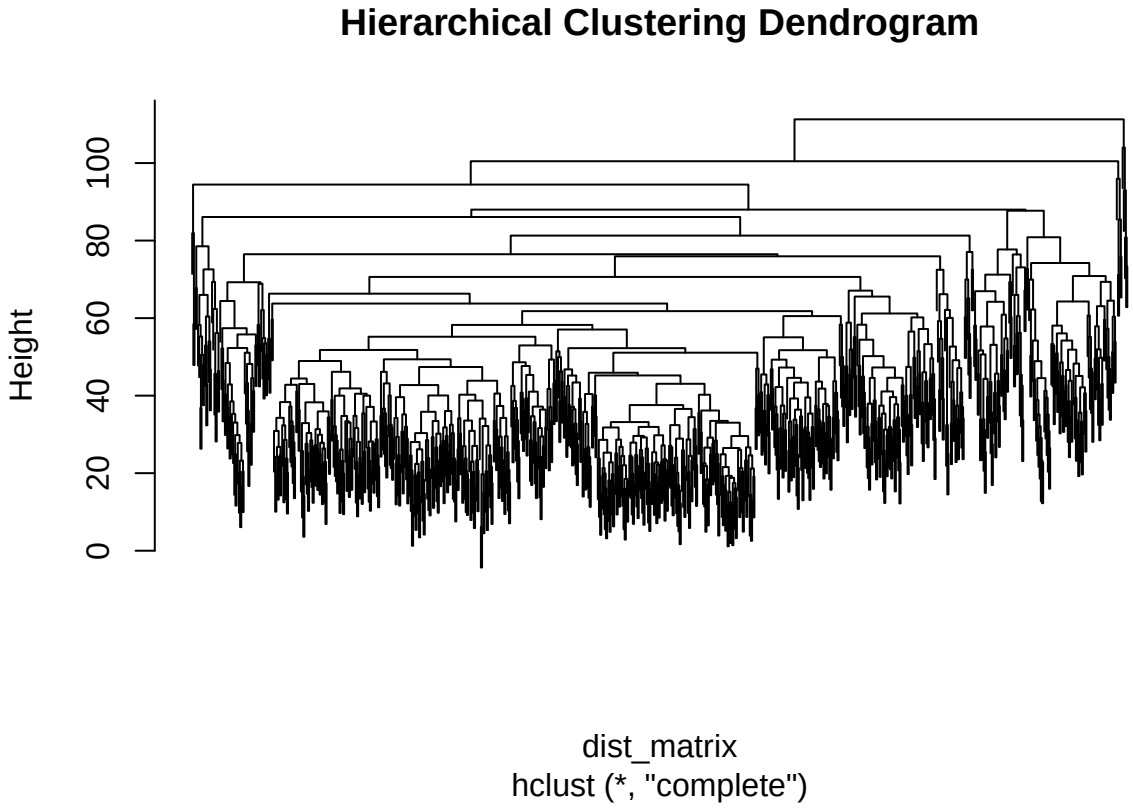
K means on optimal K = 3 (using elbow method)

```
kmeans_imgs_optimal <- kmeans(pca_data, centers = 3, nstart = 25)
fviz_cluster(kmeans_imgs_optimal, data = pca_data[,1:2], geom = "point", main = "K-means K = 3")
```



Hierarchial clustering

```
dist_matrix <- dist(pca_data, method="euclidean")
hclust_pca <- hclust(dist_matrix, method = "complete")
plot(hclust_pca, labels = FALSE, main = "Hierarchical Clustering Dendrogram")
```


Hierarchical Clustering Dendrogram
Height
0
20
40
60
80
100
dist_matrix
hclust (*, "complete")