Clustering on Stats

Isaac Baguisa

2025-03-20

Before k-means, PCA on stats dataset only

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, dendrogram

```
library(ggplot2)
library(cluster)
library(factoextra)
library(ggfortify)
library(tidyverse)
library(VIM)
library(caret)
load("../Data/pokemon.RData")
stats_numeric <- stats %>%
  select(-c(abilities, capture_rate, classfication, japanese_name, name,
            name.simple, type1, type2, image_path, has_img))
#Imputation on missing data using KNN
stats_numeric_impKNN <- kNN(stats_numeric)
#Remove cols with variance = 0
# Check the variance of each column
variances <- apply(stats_numeric_impKNN, 2, var)
zero_var_col <- which(variances == 0)
stats_numeric_impKNN <- stats_numeric_impKNN[, -zero_var_col]
```

PCA on stats

```
pca_stats <- prcomp(stats_numeric_impKNN, center = TRUE, scale. = TRUE)
pca_stats_df <- as.data.frame(pca_stats$x)
colnames(pca_stats_df) <- paste0("PC", 1:ncol(pca_stats_df))
loadings <- pca_stats$rotation
```

```
var_explained <- summary(pca_stats)$importance[2, ]  # Proportion of variance explained
cumulative_var <- cumsum(var_explained)
(cumulative_var)
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
## 0.16578 0.26587 0.33782 0.40198 0.46428 0.51601 0.56327 0.60756 0.64868 0.68521
##    PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20
## 0.71773 0.74928 0.77765 0.80222 0.82357 0.84430 0.86400 0.88133 0.89763 0.91129
##    PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28    PC29    PC30
## 0.92404 0.93477 0.94465 0.95358 0.96206 0.96982 0.97674 0.98268 0.98758 0.99150
##    PC31    PC32    PC33    PC34    PC35    PC36    PC37
## 0.99407 0.99633 0.99816 0.99975 1.00000 1.00000 1.00000
```

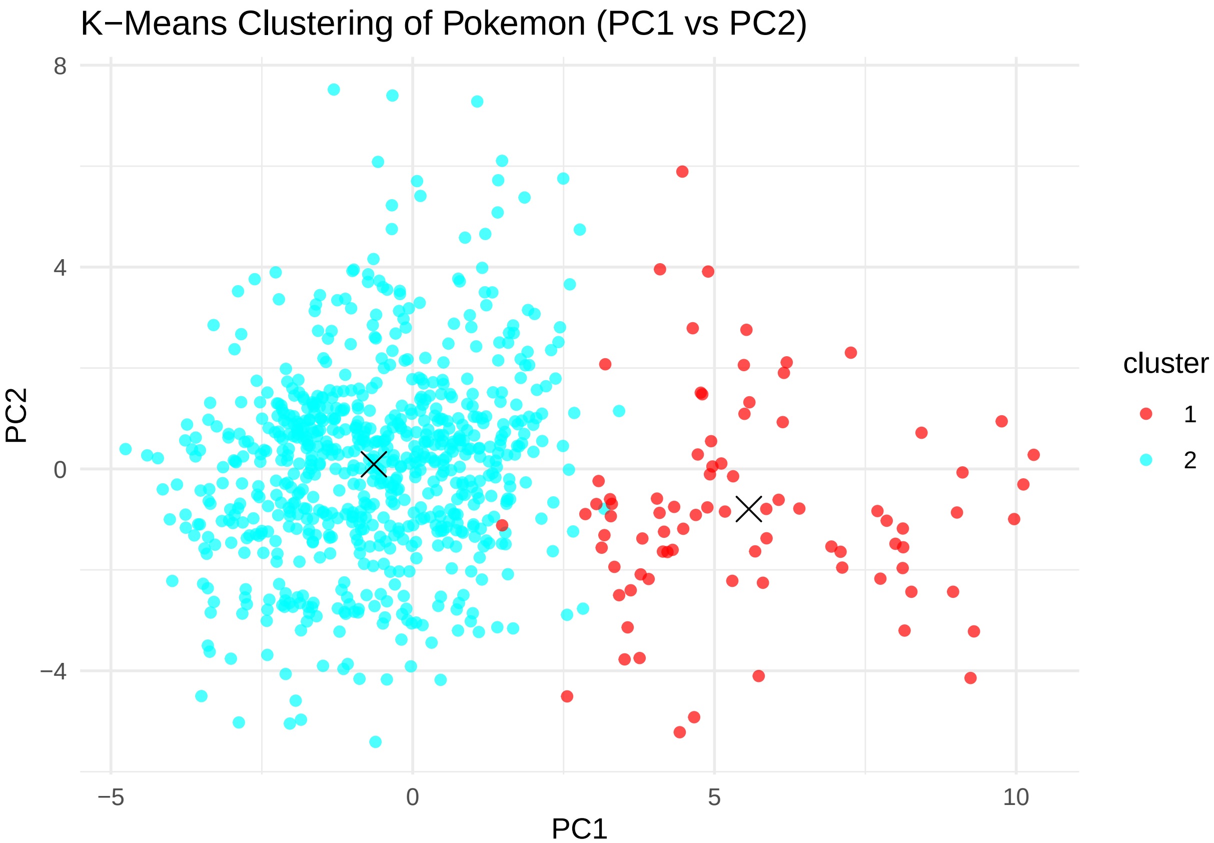
```
# Keep 20 PCs (90% VE)
pca_stats_df <- pca_stats_df[,1:20]
```

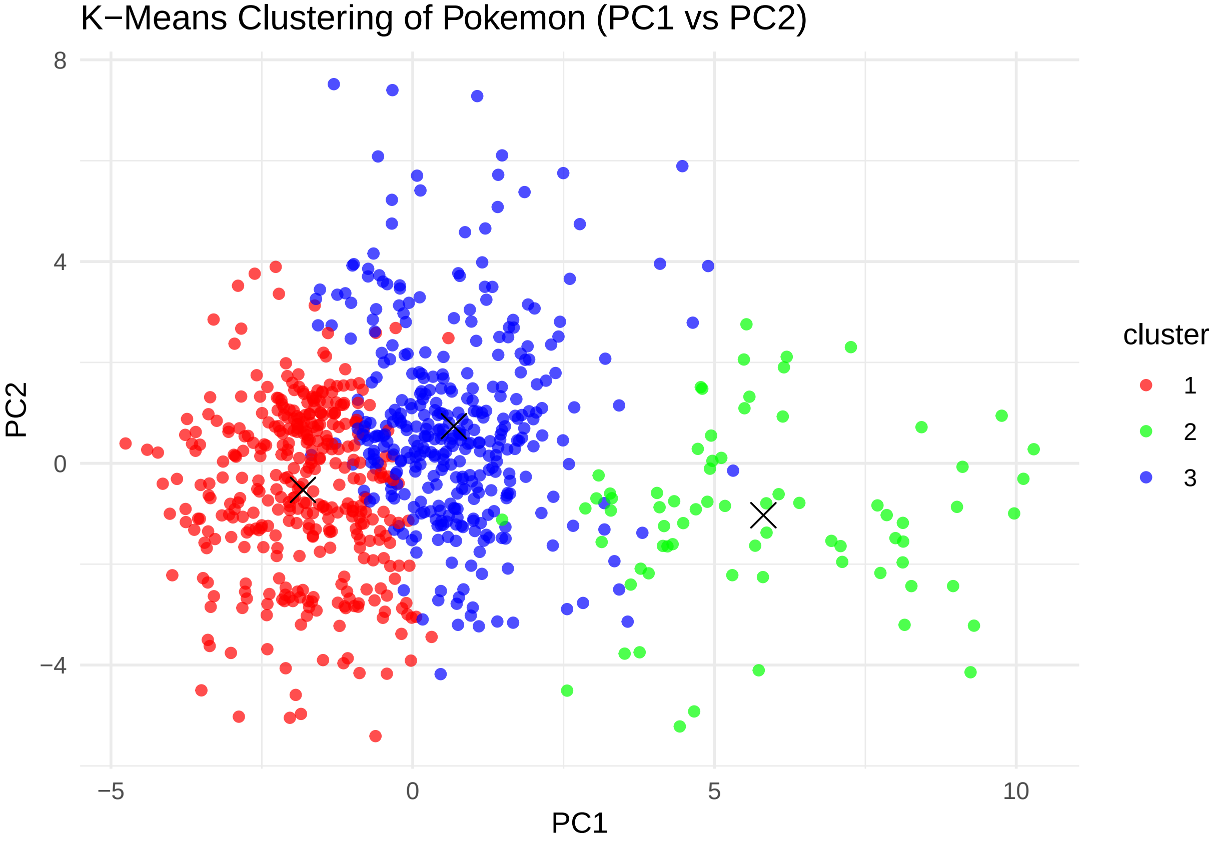
Visualize different K values on stats data

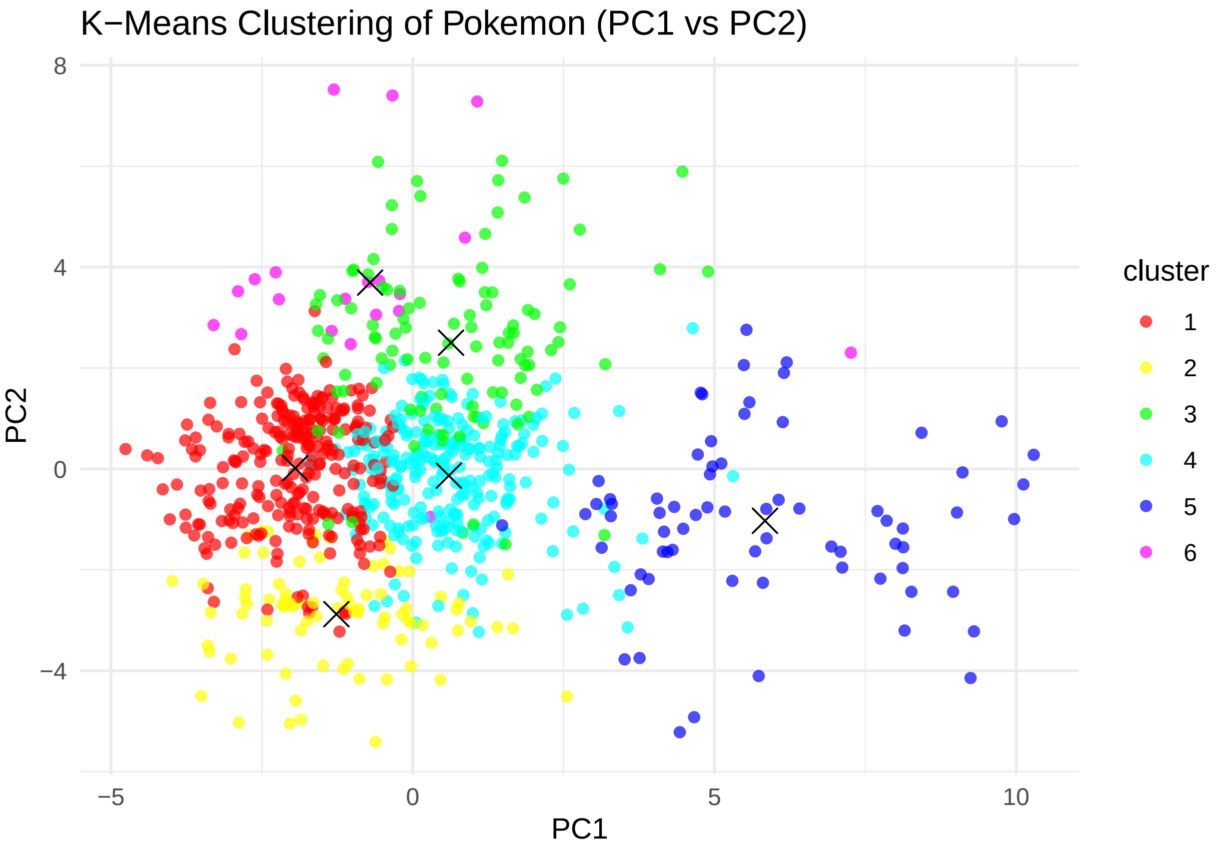
```
k_types <- length(unique(stats$type1))
kmeans_stats <- kmeans(pca_stats_df, centers = k_types, nstart = 25)
```

K-means on k=18 visualization

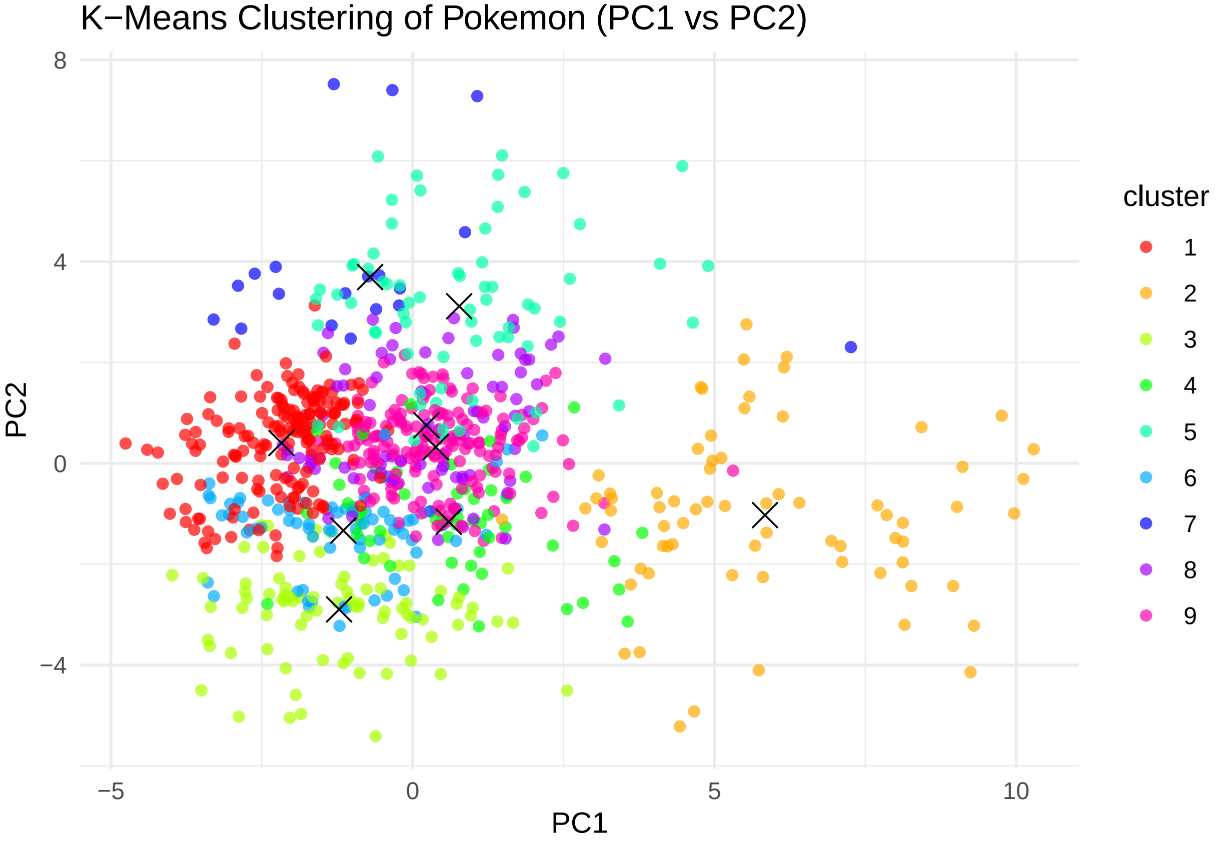
```
# Helper function lecture 7
scatterplot = function(X, M, cluster, label = FALSE){
  X_df <- data.frame(X, cluster = as.factor(cluster))
  M_df <- data.frame(M)

  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_stats <- kmeans(pca_stats_df, centers = iter, nstart = 25)
  print(scatterplot(pca_stats_df, kmeans_stats$centers, kmeans_stats$cluster))
}
```


K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
8
4
0
−4
−5
0
5
10
PC1
cluster
1
2

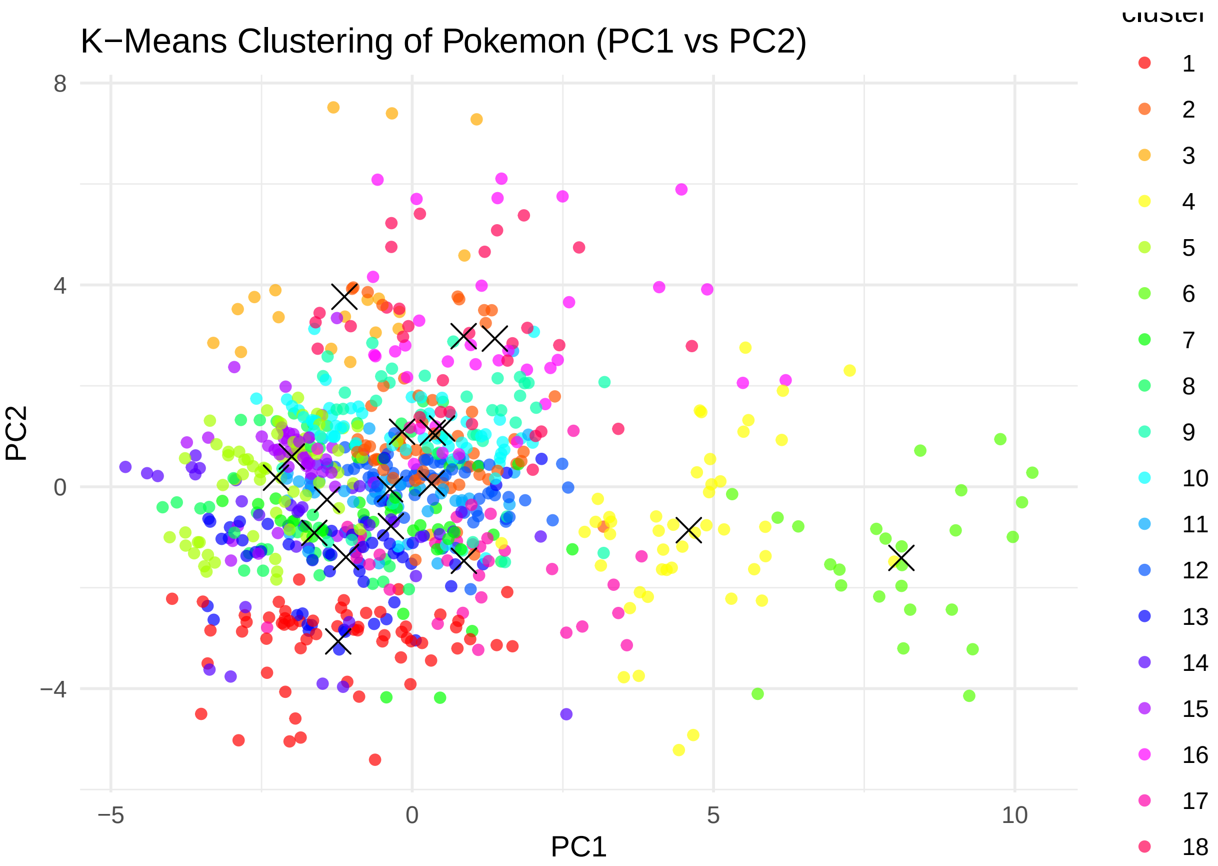

K−Means Clustering of Pokemon (PC1 vs PC2)
PC2
8
4
0
−4
−5
0
5
10
PC1
cluster
1
2
3



K−Means Clustering of Pokemon (PC1 vs PC2)

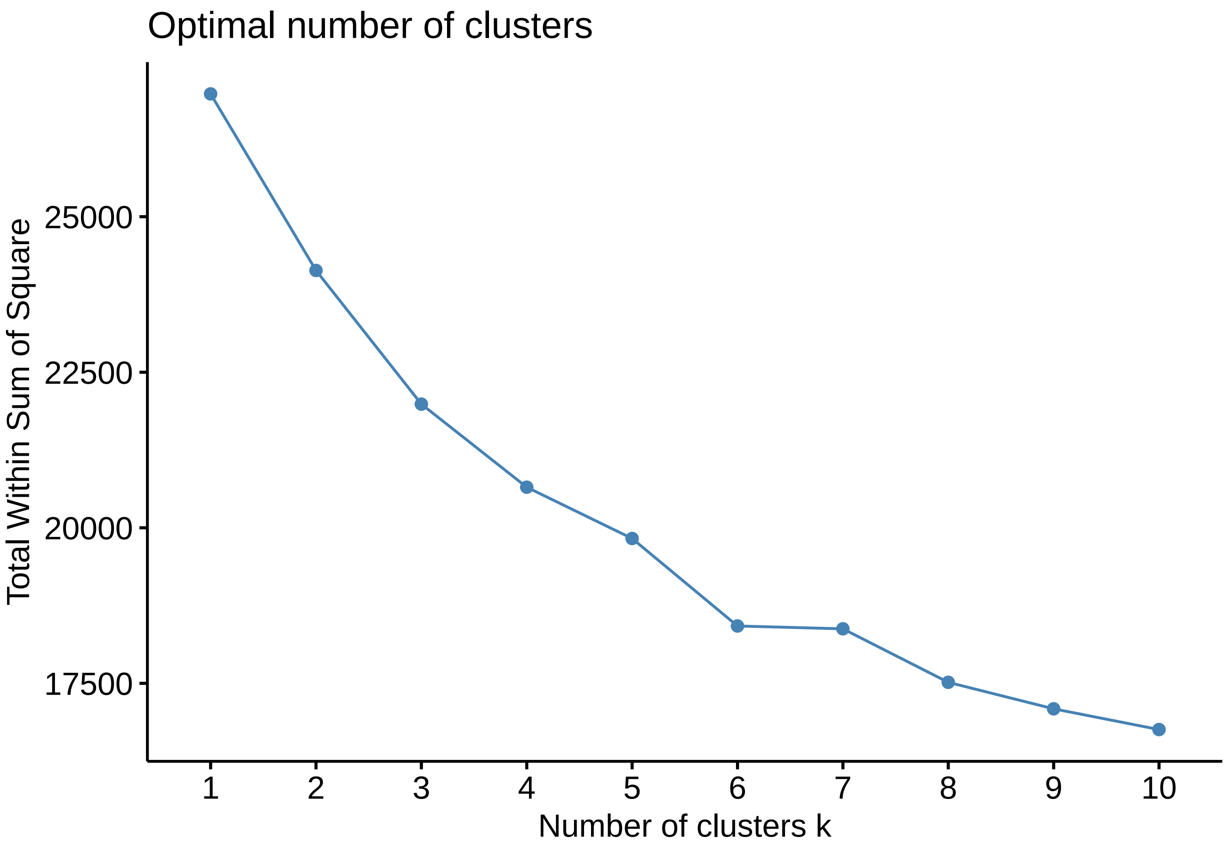


K−Means Clustering of Pokemon (PC1 vs PC2)



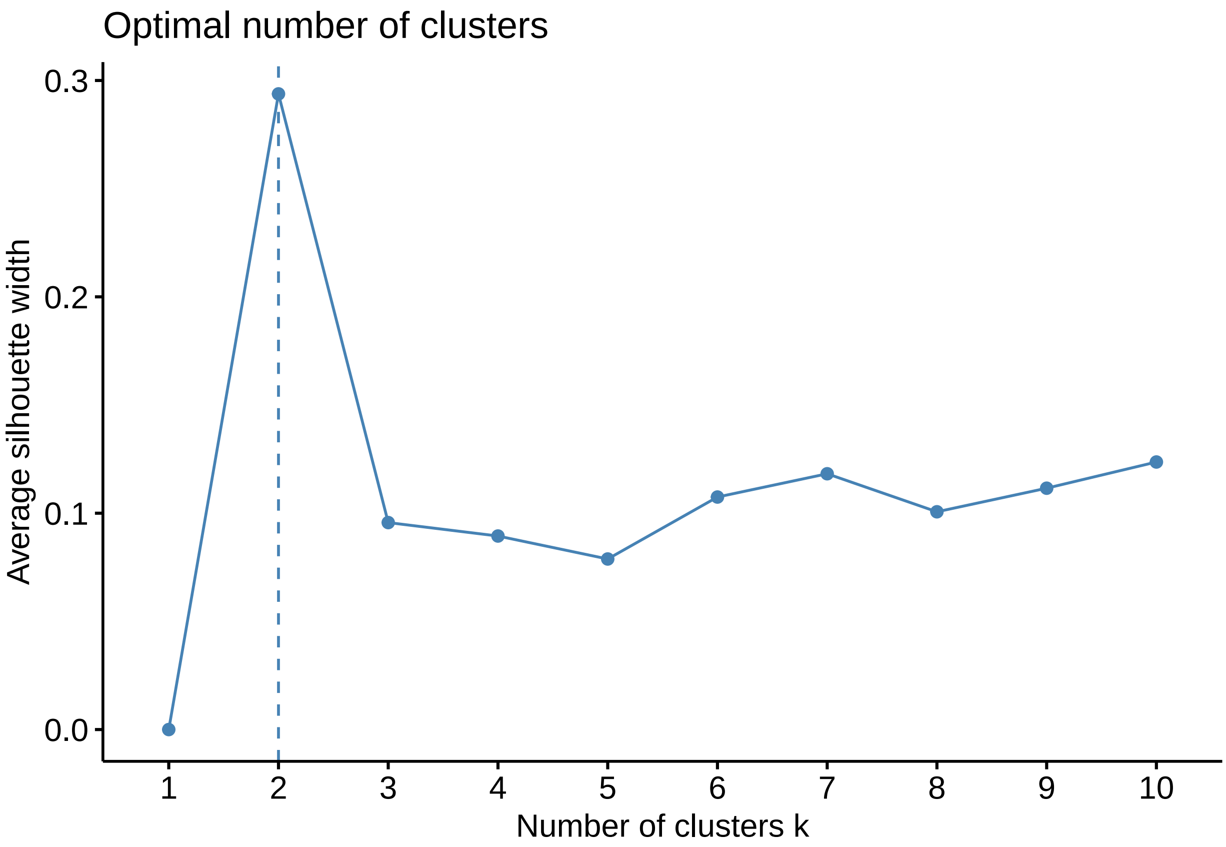
Tune for optimal k

```
fviz_nbclust(pca_stats_df, kmeans, method = "wss", algorithm = "Lloyd")
```



Elbow method: Optimal K = 3 or 4

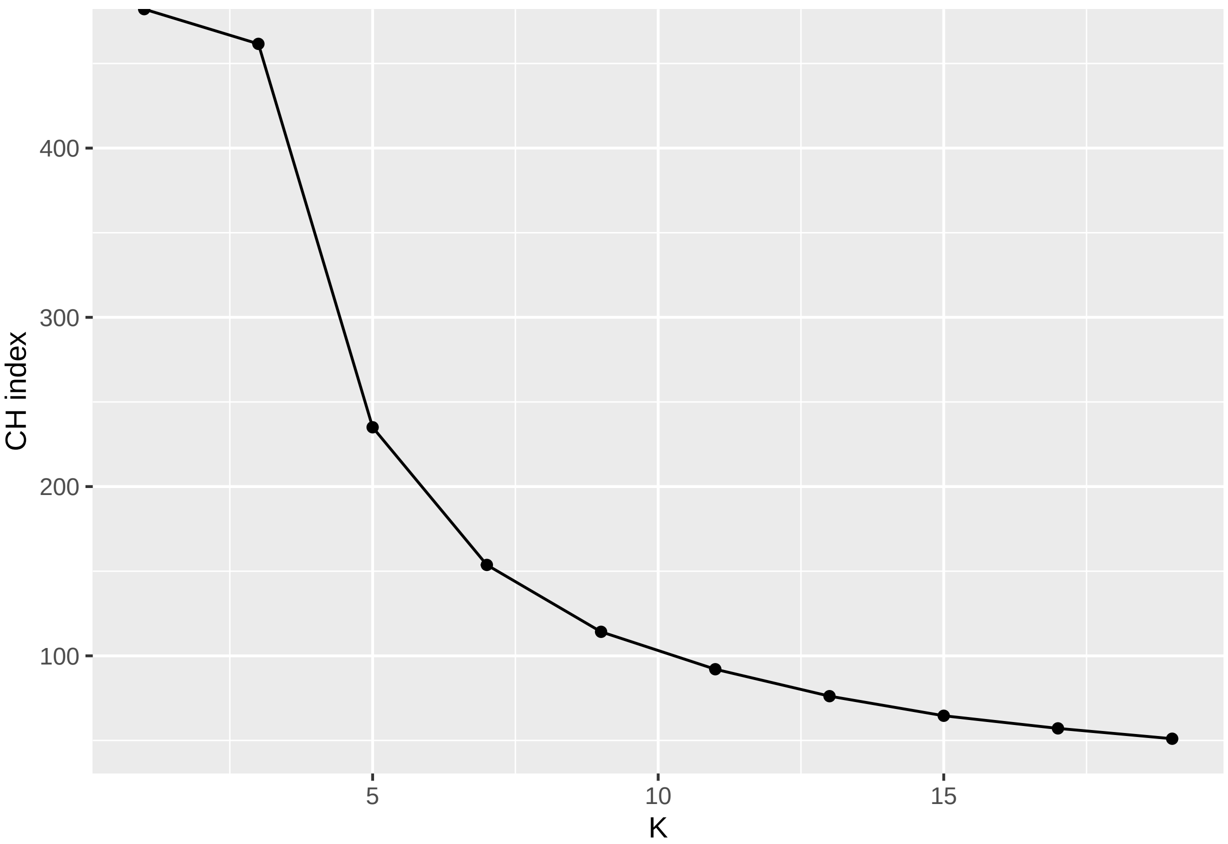
```
fviz_nbclust(pca_stats_df, kmeans, method = "silhouette")
```



Silhouette method: Optimal K = 2

CH index for K-means

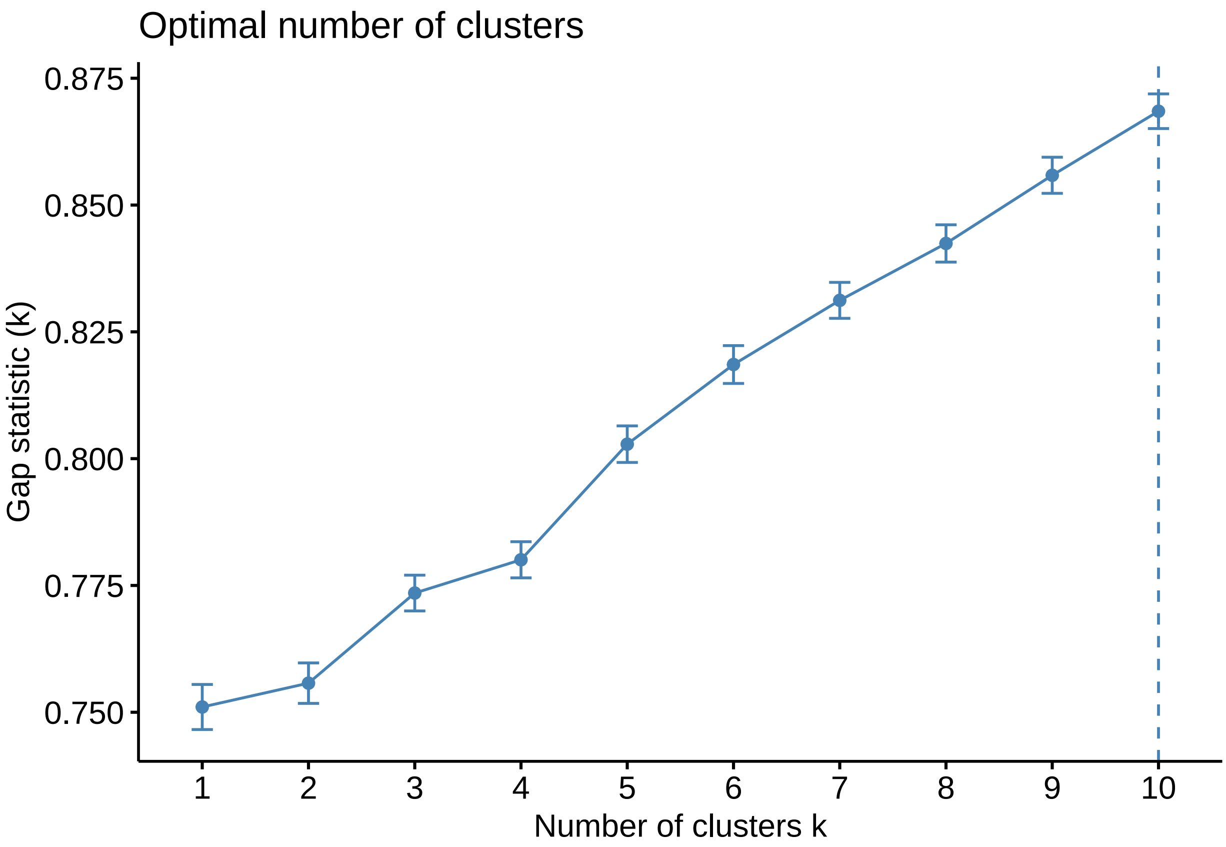
```
CHs = c()
Ks = seq(1, 20, 2)

for(K in Ks){
  KM = kmeans(pca_stats_df, 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_stats_df)

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



Gap stats: Optimal K = 10

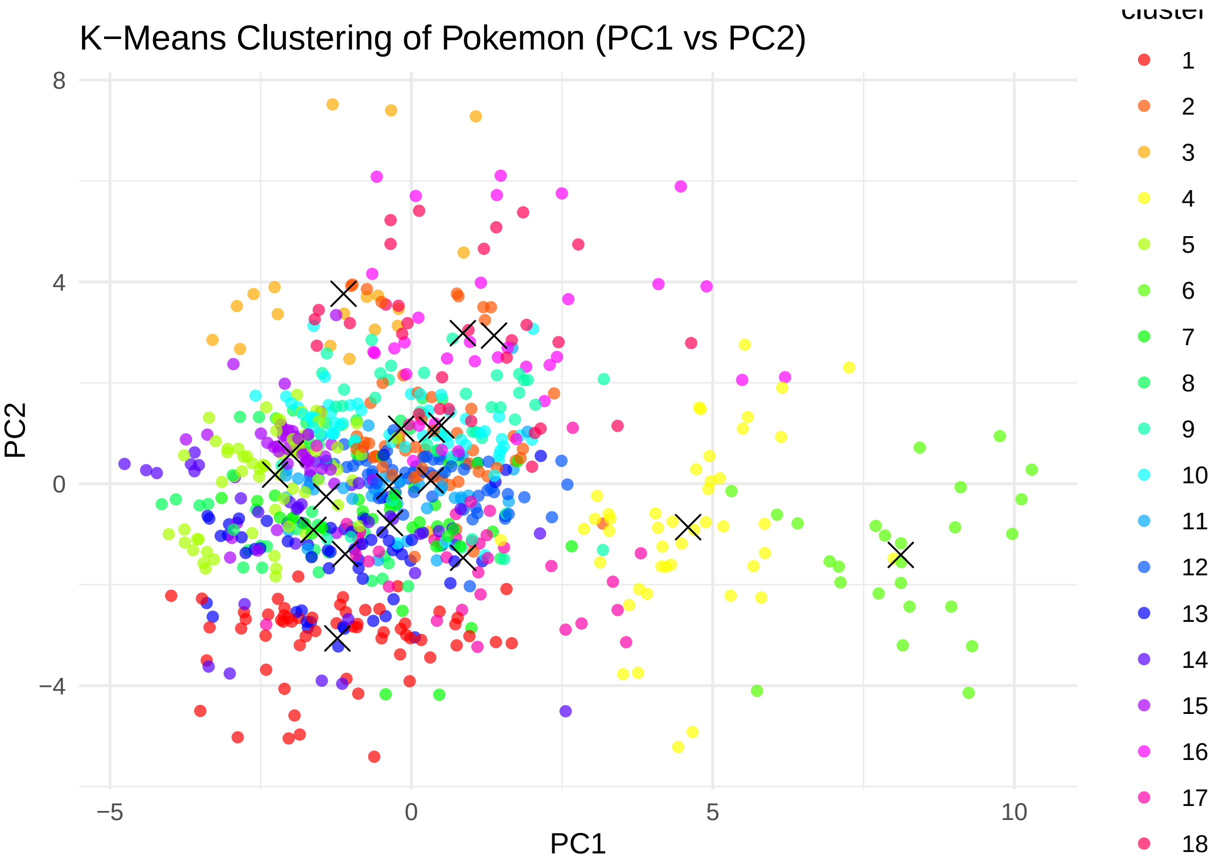
Compare K-means and K-means++

K-means with K-means++ initialization

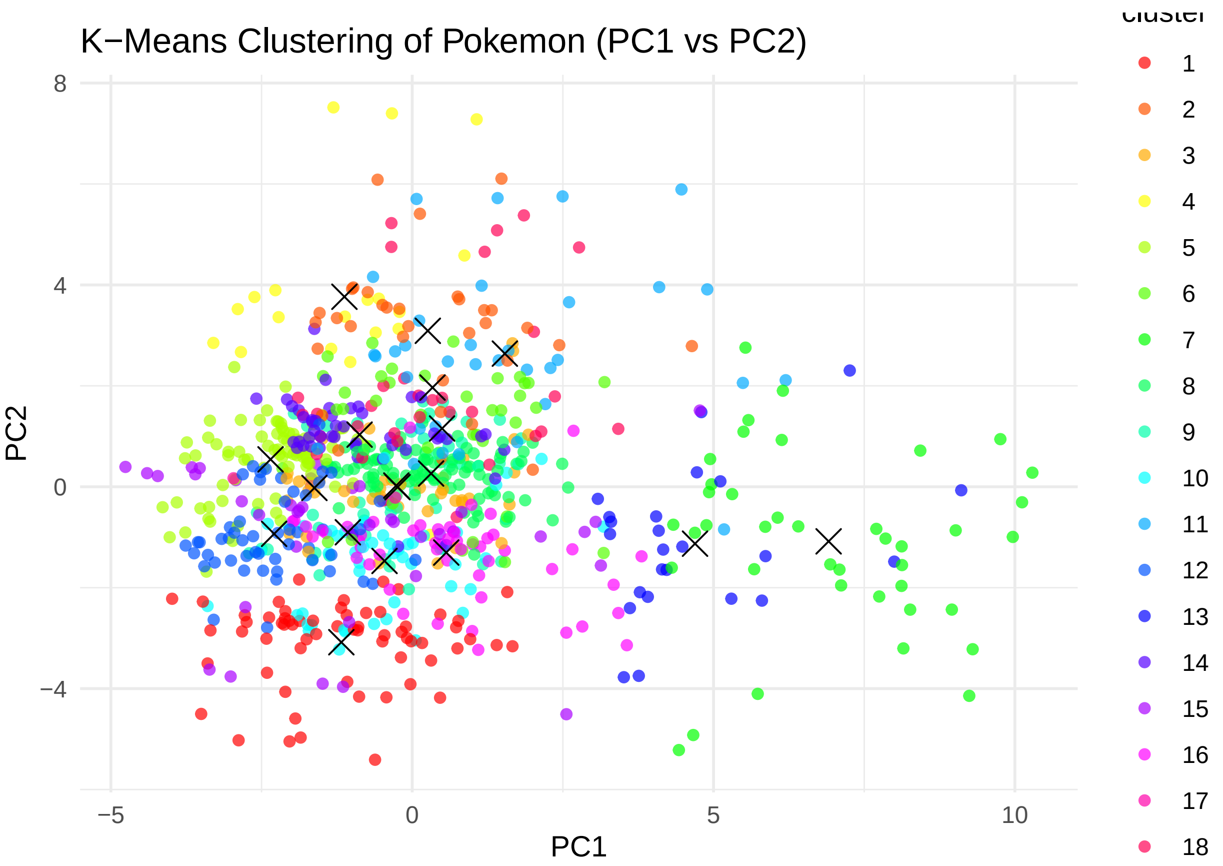
```
#Ref: lecture 7
# Randomly select the first centroid
n <- nrow(pca_stats_df)
M <- pca_stats_df[sample(1:n, 1), , drop = FALSE]

for (i in 2:k_types) {
  # Dist from each point to the nearest centroid
  D <- as.matrix(dist(rbind(M, pca_stats_df)))[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_stats_df[pick, , drop = FALSE])
}

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

Compare K-means and K-means++ plots

```
scatterplot(pca_stats_df, kmeans_stats$centers, kmeans_stats$cluster)
```

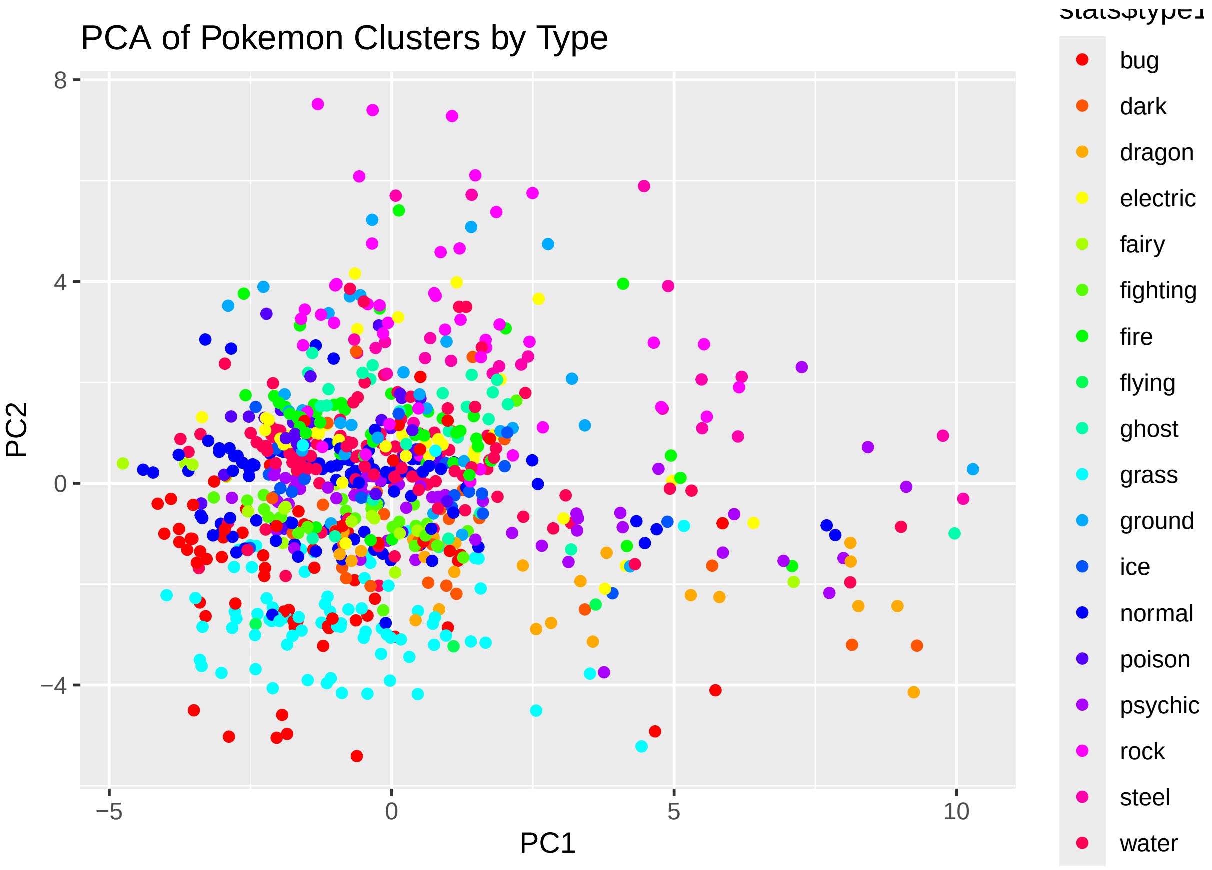


```
scatterplot(pca_stats_df, kmeans_pp$centers, kmeans_pp$cluster)
```



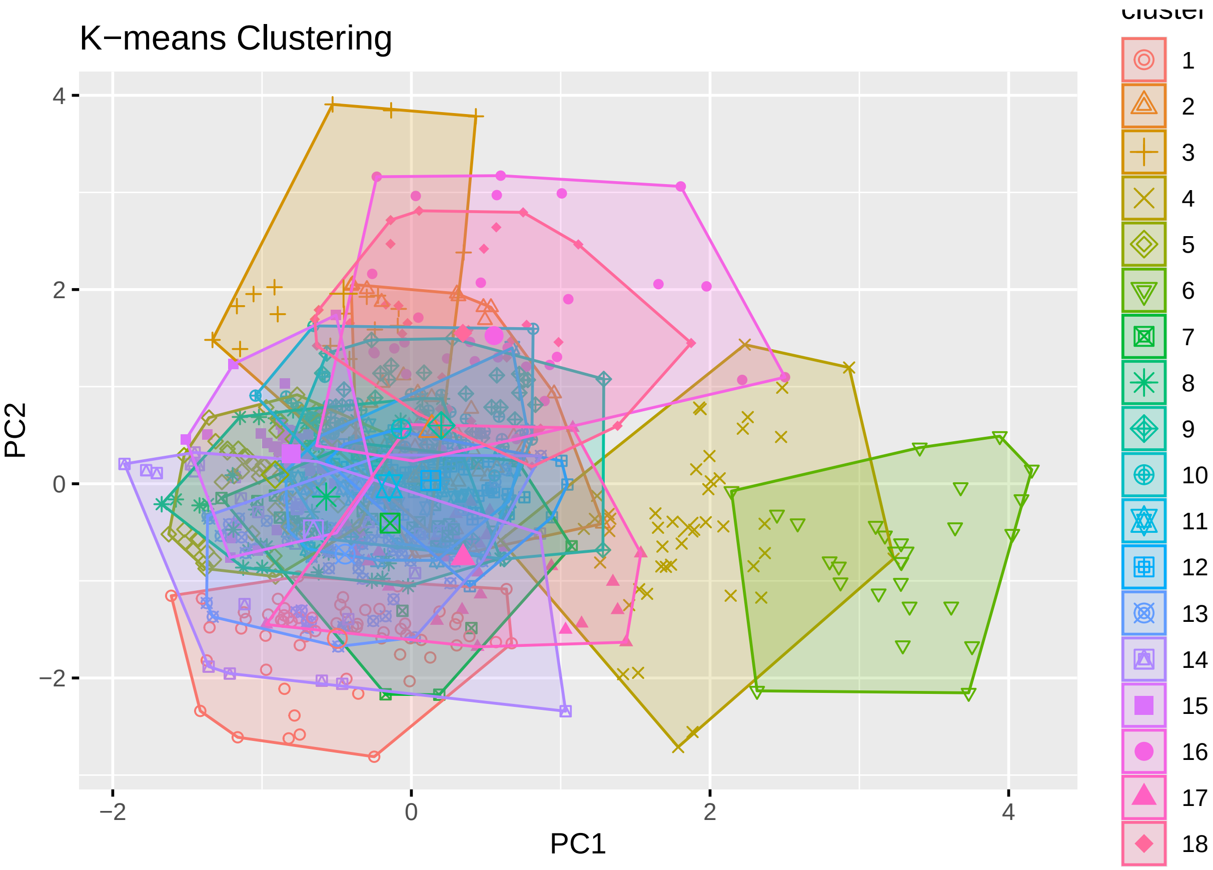
Visualize cluster on actual labels

```
ggplot(data = pca_stats_df, 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))))
```



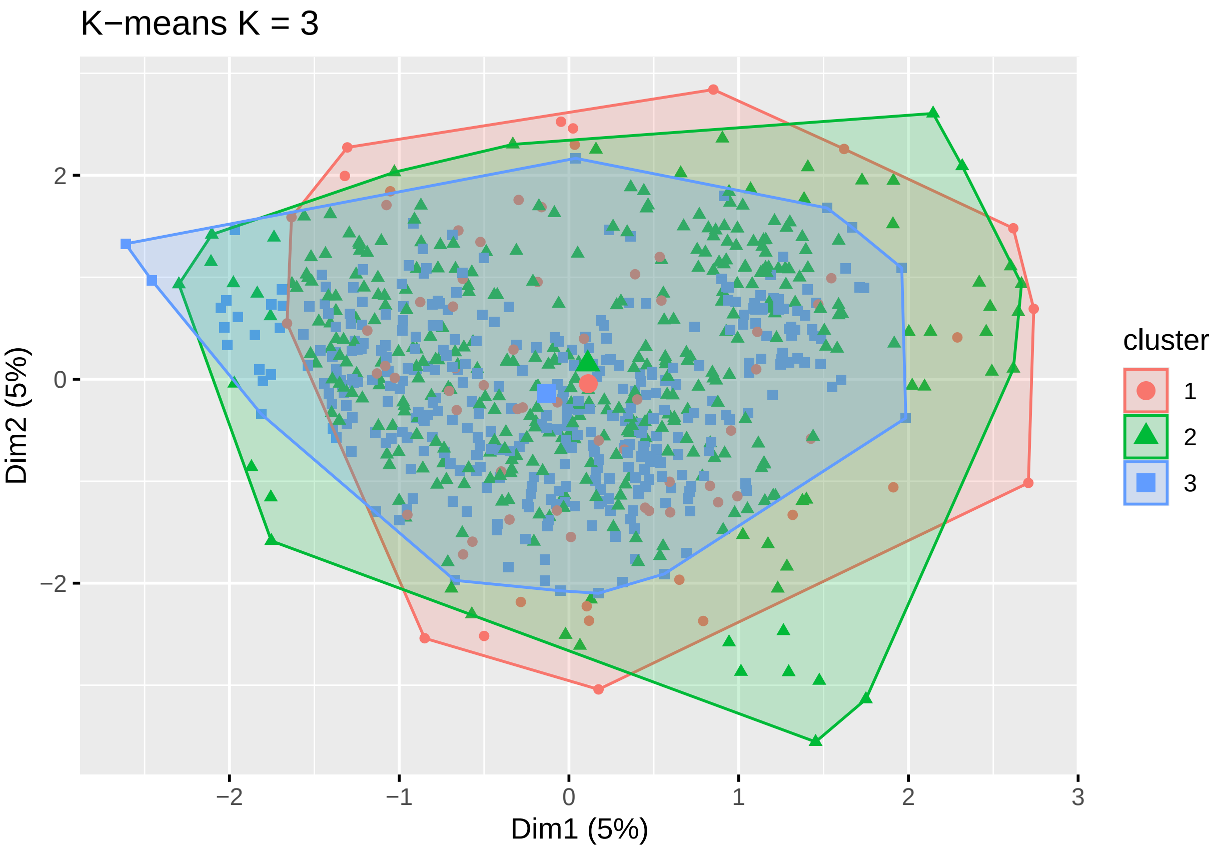
Visualize K-means results using two principal components

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



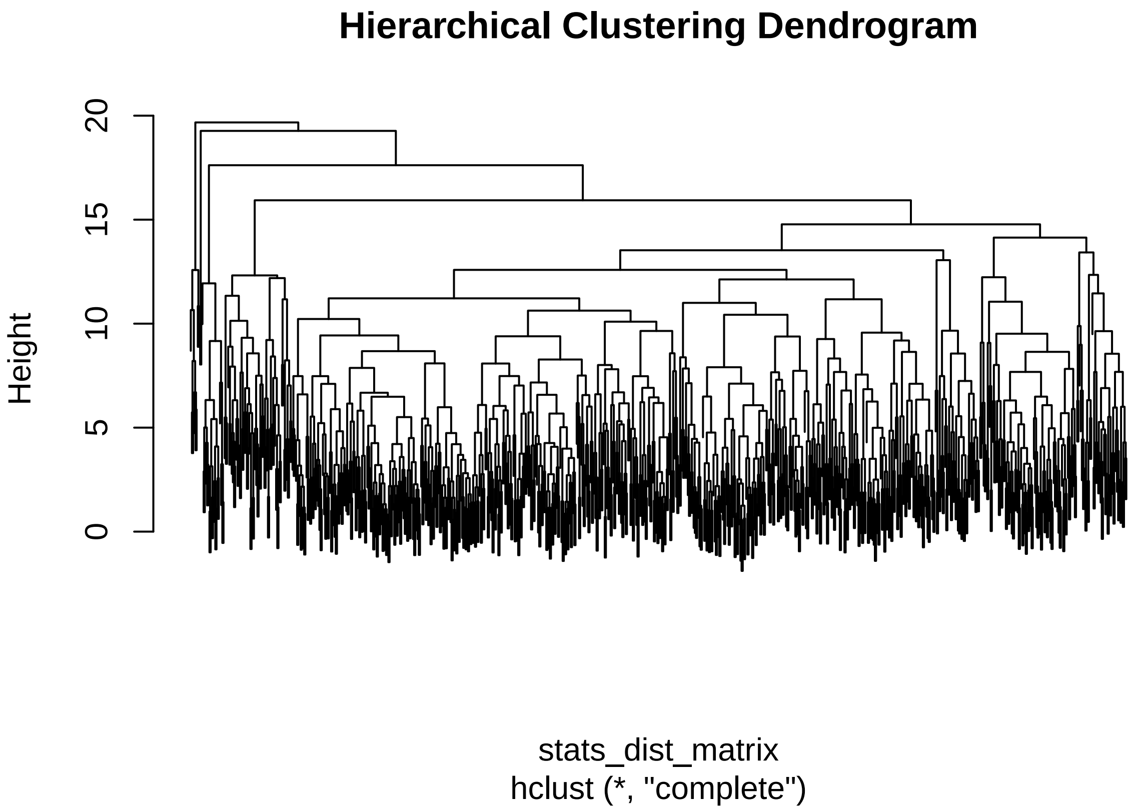
K means on optimal K = 3 (using elbow method)

```
kmeans_stats_optimal <- kmeans(pca_stats_df, centers = 3, nstart = 25)
fviz_cluster(kmeans_stats_optimal, data = pca_stats_df, geom = "point", main = "K-means K = 3")
```



Hierarchial clustering

```
stats_dist_matrix <- dist(pca_stats_df, method="euclidean")
stats_hclust_pca <- hclust(stats_dist_matrix, method = "complete")
plot(stats_hclust_pca, labels = FALSE, main = "Hierarchical Clustering Dendrogram")
```


Hierarchical Clustering Dendrogram
Height
0
5
10
15
20
stats_dist_matrix
hclust (*, "complete")