

Statistical Learning Workshop - Hierarchical

Abdollah Safari

26/02/2021

Load data

Use the `penguins` dataset that was wrangled previously in “Penguins.R.”

```
# Set random seed for reproducibility in processes like
# splitting the data
set.seed(1) # You can use any number here

# Import data and sample
df <- import("~/Downloads/Ex_Files_Data_Science_R/Exercise Files/data/penguins.rds")
%>%
  sample_n(100) # Reduce n for graphing

# Look at the first few rows of the prepared data frame
df
```

```
## # A tibble: 100 x 5
##   y          bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>          <dbl>          <dbl>          <int>          <int>
## 1 Chinstrap      49.8            17.3            198            3675
## 2 Gentoo         49.3            15.7            217            5850
## 3 Adelie        44.1             18             210            4000
## 4 Chinstrap      46.7            17.9            195            3300
## 5 Gentoo         47.2            13.7            214            4925
## 6 Gentoo         48.4            16.3            220            5400
## 7 Chinstrap      42.5            16.7            187            3350
## 8 Adelie        41.3            20.3            194            3550
## 9 Chinstrap      51.3            19.2            193            3650
## 10 Chinstrap     52.2            18.8            197            3450
## # ... with 90 more rows
```

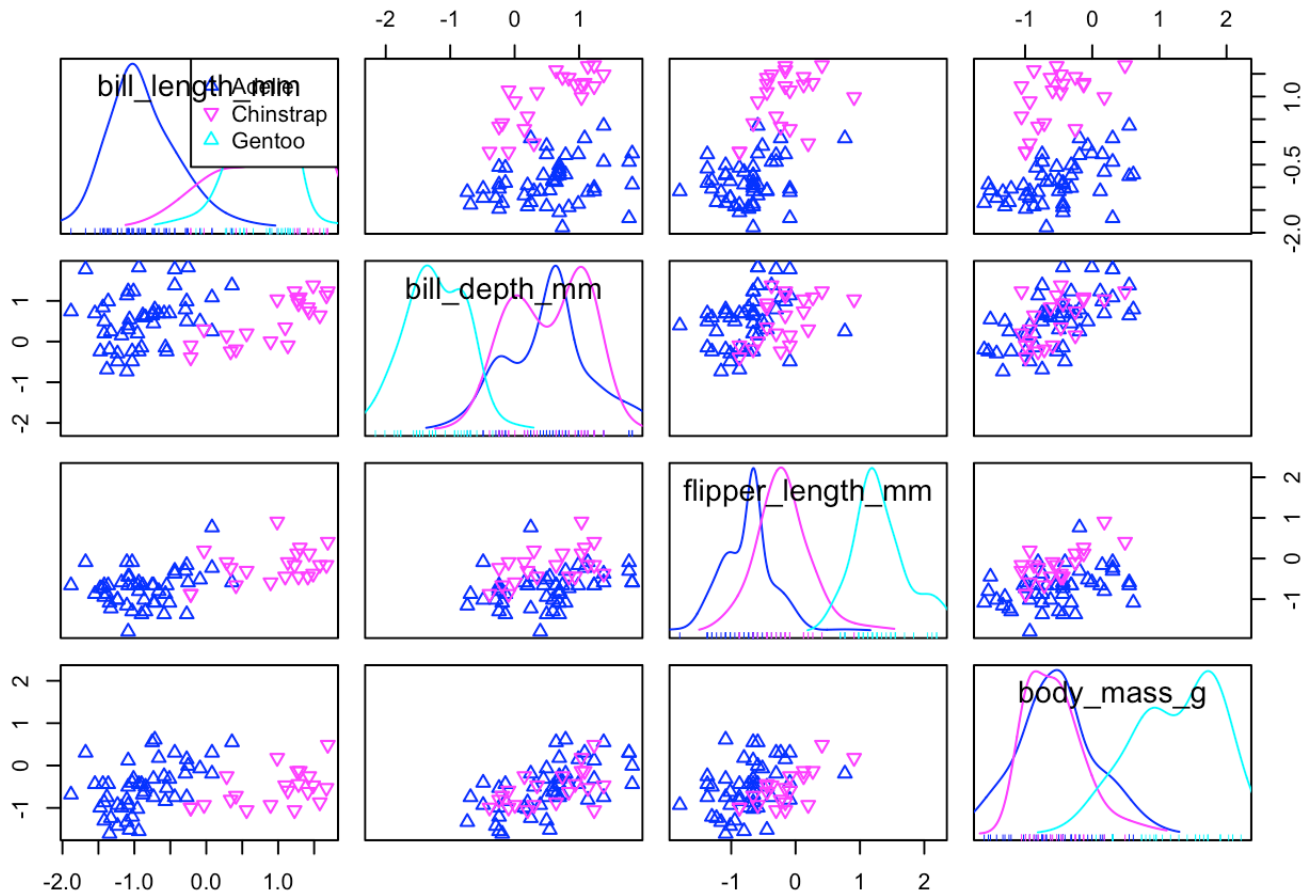
Prepare data

```
# Separate the class labels
y <- df %>%
  pull(y)      # Select just `y` as a vector

df %<>%
  select(-y) %>% # Select everything except `y`
  scale()        # Standardize variables
```

Explore data

```
# Make a scatter plot of some exploratory variables and
# color according to species (y)
scatterplotMatrix(
  ~ bill_length_mm +
    bill_depth_mm +
    flipper_length_mm +
    body_mass_g |
    y,
  data = df,
  regLine = FALSE,
  smooth = FALSE,
  pch = c(2, 6)
)
```



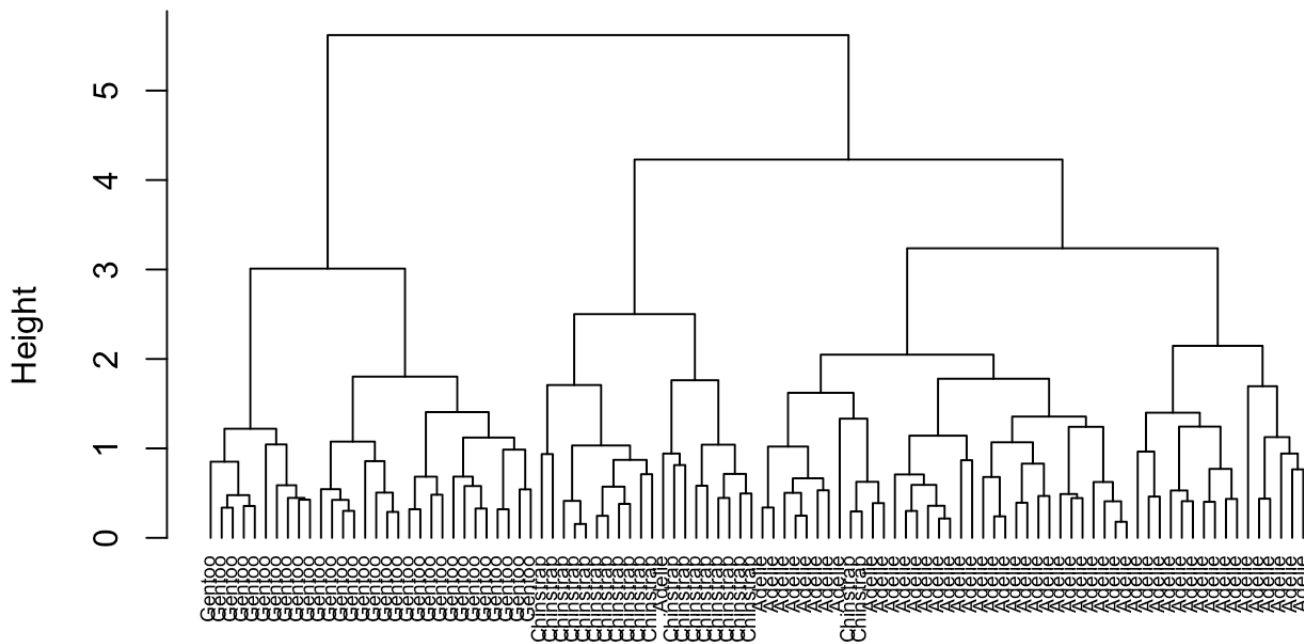
Initial Hierarchical clustering

```
# Calculate clusters
hc <- df %>%      # Get data
  dist %>%        # Compute distance/dissimilarity matrix
  hclust           # Compute hierarchical clusters

hc$labels <- y    # Set the class labels

# Plot the dendrogram
hc %>% plot(      # Generic X-Y plotting
  hang = -1,      # Line up names at bottom
  cex = 0.6       # Make font smaller
)
```

Cluster Dendrogram

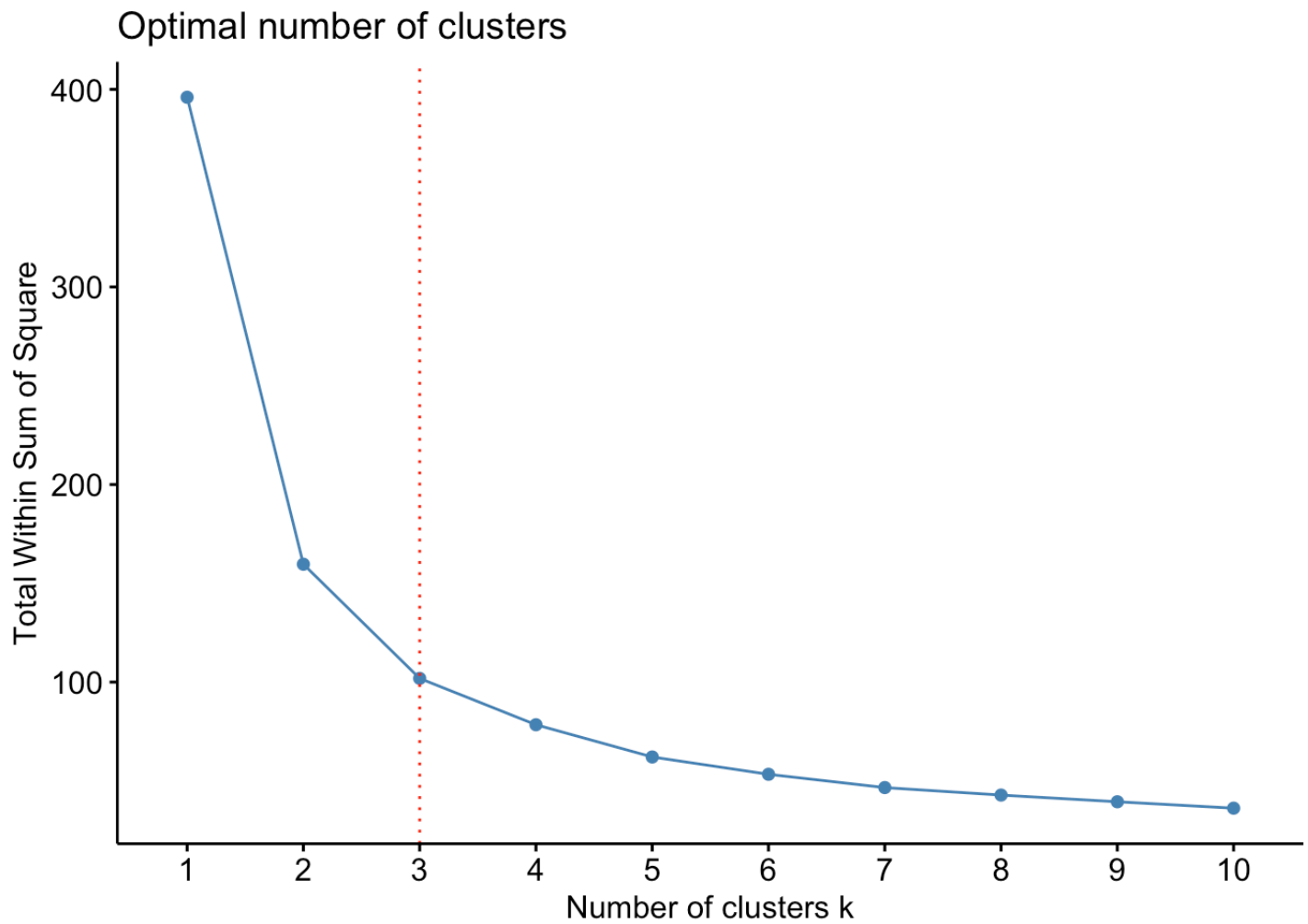


`hclust (*, "complete")`

Number of clusters (K)

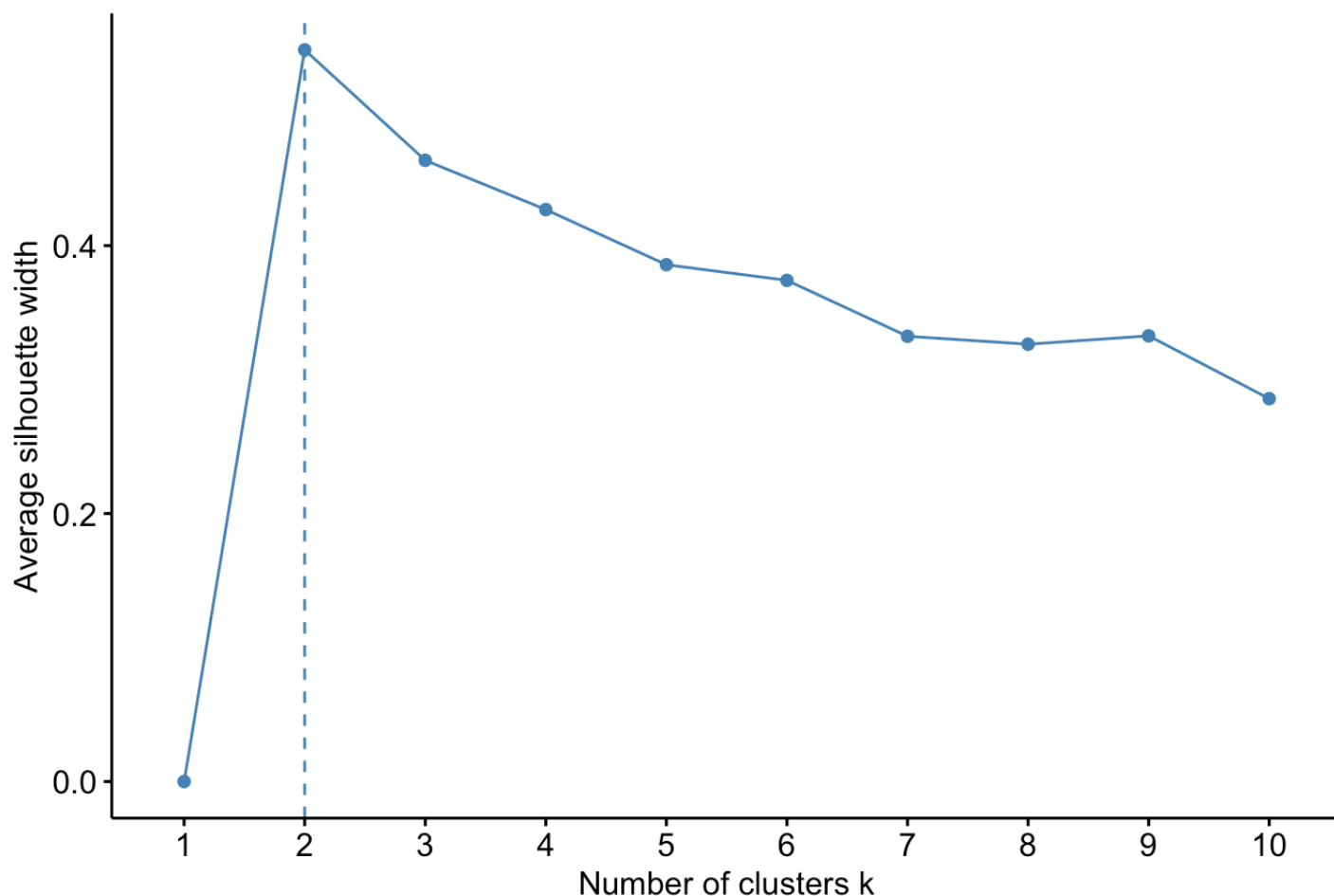
```
# OPTIMAL NUMBER OF CLUSTERS #####

# Elbow method
df %>%
  fviz_nbclust(                # From factoextra
    FUN = hcut,                 # Method for clustering
    method = "wss"              # "within cluster sums of squares"
  ) +
  geom_vline(                   # Reference line
    xintercept = 3,             # Draw line at X = 3
    color = "red",              # Color red
    linetype = "dotted"         # Use dotted line
  )                             # Look for "bend" in curve
```

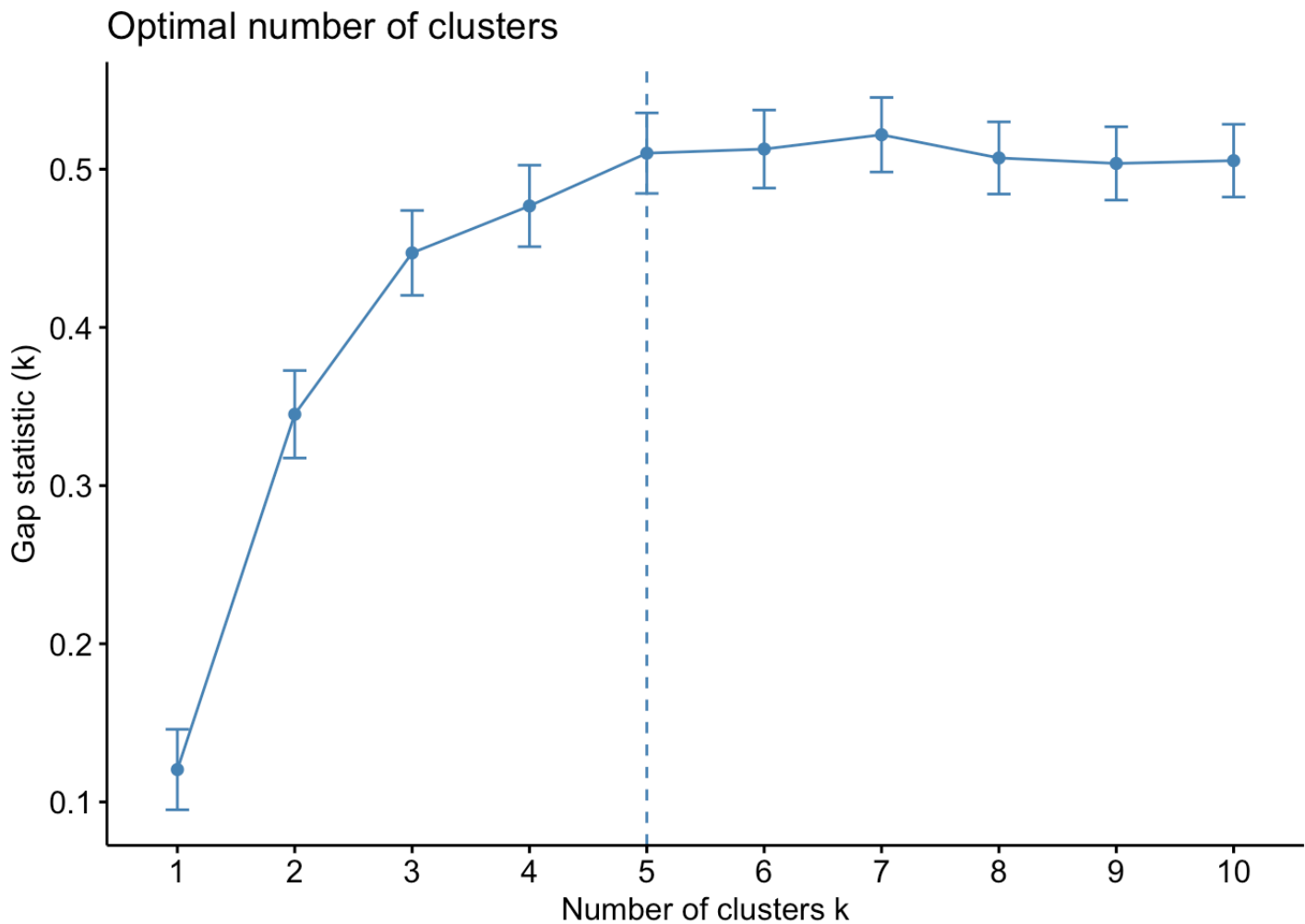


```
# Silhouette method
df %>%
  fviz_nbclust(
    FUN = hcut,
    method = "silhouette" # Look for maximum width
  )
```

Optimal number of clusters



```
# Gap method: This compares the total intracluster variation
# for values of k with their expected values from null
# distributions (from Monte Carlo simulations)
df %>%
  clusGap(          # Function from `cluster`
    FUN = hcut,     # Method for clustering
    K.max = 10,     # Maximum number of clusters
    B = 100         # n Monte Carlo/bootstrap samples
  ) %>%
  fviz_gap_stat()  # Look for highest value
```

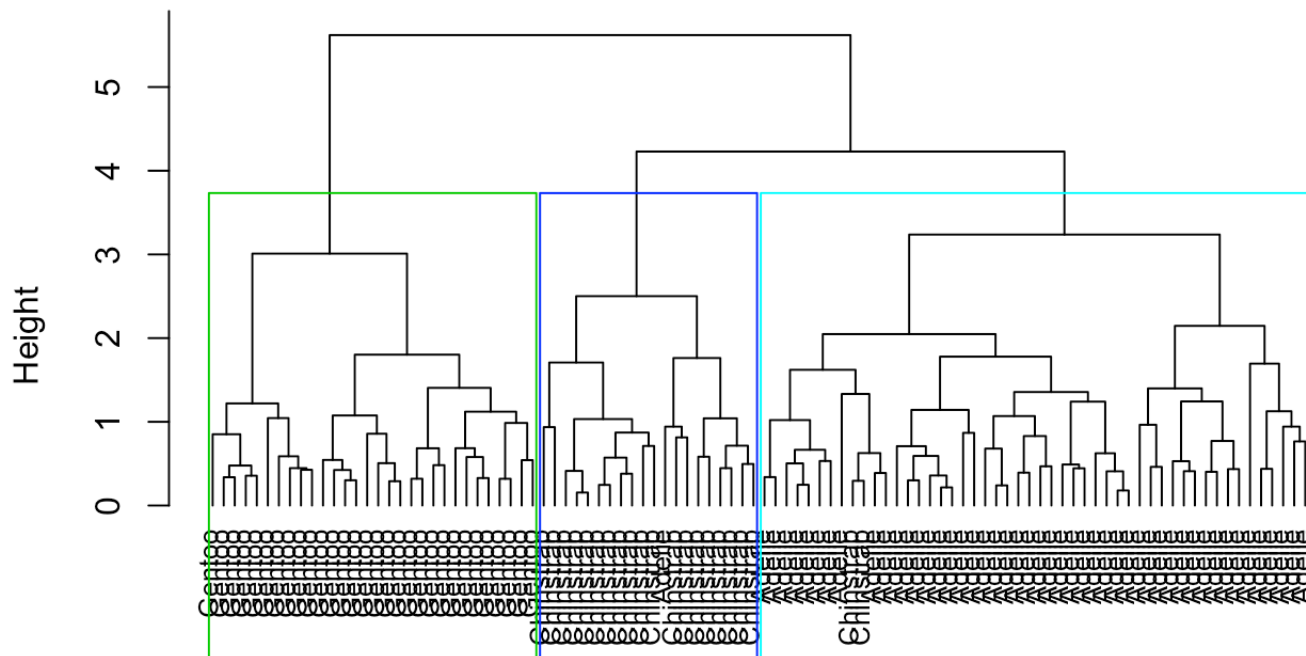


Final Hierarchical clustering

```
# Plot dendrogram (again)
hc %>% plot(          # Generic X-Y plotting
  hang = -1,          # Line up names at bottom
  cex = 0.8           # Make font smaller
)

# Draw boxes around clusters
hc %>% rect.hclust(   # Add rectangles to clusters
  k = 3,              # Draw three rectangles
  border = 3:5        # Use colors 3 through 5
)
```

Cluster Dendrogram




```
# Fit the hierarchical clustering groups to data
y_hc = hc %>% cutree(3)

# Visualize the clusters in 2-D space; label points
# according to species and color points according to
# assigned cluster
fviz_cluster(
  list(
    cluster = y_hc,
    data = df
  ),
  geom = c("point")
) +
geom_text(
  vjust = 1.5, # Label points
  aes(
    color = as.factor(y_hc),
    label = hc$labels
  )
)
```

Cluster plot

