DS 5230 Homework 4 and Homework 5

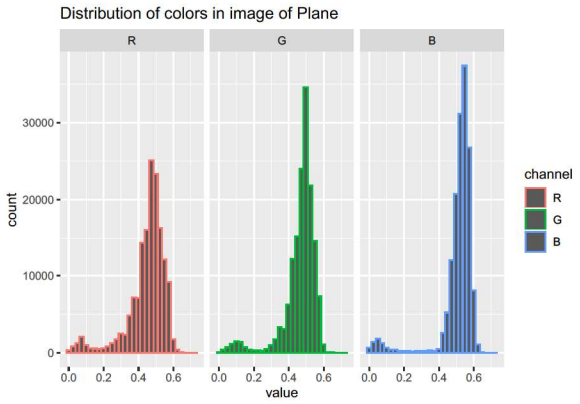
Siddarth Sathyanarayanan

4/7/2020

Homework 4

Data Preprocessing

```
img <- load.image("C:/Users/Siddarth S/Downloads/H4+H5/H4+H5/3096_colorPlane.jpg")

df <- as.data.frame(img)

df <- mutate(df, channel = factor(cc,
                                  labels = c('R',
                                             'G',
                                             'B')))

ggplot(df, aes(value, col = channel)) +
  geom_histogram(bins = 30) +
  facet_wrap(~ channel) +
  labs(title = "Distribution of colors in image of Plane")
```

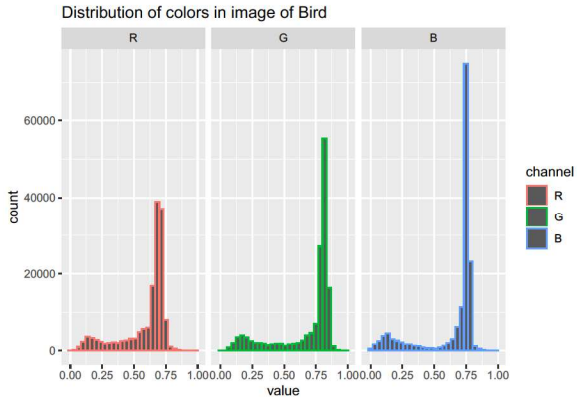




```
df_R <- df %>%
  filter(cc == 1) %>%
  dplyr::select(x, y, "R" = value)

df_G <- df %>%
  filter(cc == 2) %>%
  dplyr::select(x, y, "G" = value)

df_B <- df %>%
  filter(cc == 3) %>%
  dplyr::select(x, y, "B" = value)

matrix_colors <- df_R %>%
  inner_join(df_G) %>%
  inner_join(df_B)

normalized_matrix <-
  apply(matrix_colors, 2, function(x) {(x - min(x, na.rm = T))/(max(x, na.rm = T) - min(x, na.rm = T))})
```

```
img2 <- load.image("C:/Users/Siddarth S/Downloads/H4+H5/H4+H5/42049_colorBird.jpg")


df_2 <- as.data.frame(img2)
```



```
df_2 <- mutate(df_2, channel = factor(cc,
                                  labels = c('R',
                                             'G',
                                             'B')))

ggplot(df_2, aes(value, col = channel)) +
  geom_histogram(bins = 30) +
  facet_wrap(~ channel) +
  labs(title = "Distribution of colors in image of Bird")
```



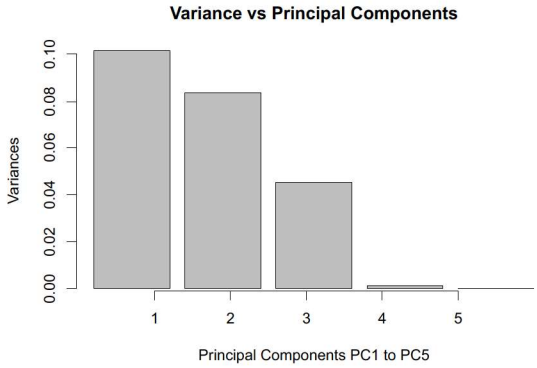

```
df_R_2 <- df_2 %>%
  filter(cc == 1) %>%
  dplyr::select(x, y, "R" = value)

df_G_2 <- df_2 %>%
  filter(cc == 2) %>%
  dplyr::select(x, y, "G" = value)

df_B_2 <- df_2 %>%
  filter(cc == 3) %>%
  dplyr::select(x, y, "B" = value)

matrix_colors_2 <- df_R_2 %>%
  inner_join(df_G_2) %>%
  inner_join(df_B_2)
```



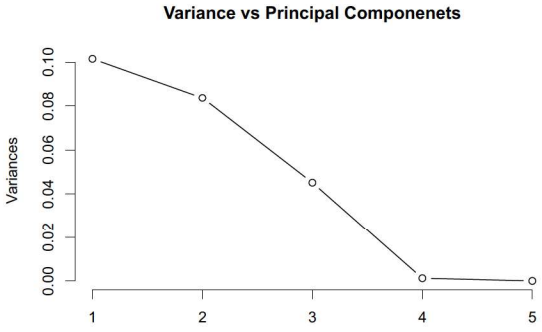
```
normalized_matrix_2 <-
  apply(matrix_colors_2, 2, function(x) {(x - min(x, na.rm = T))/(max(x, na.rm = T) - min(x, na.rm = T))})
```

PCA on Plane image

```
pc1 <- prcomp(normalized_matrix)

plot(pc1,
     xlab = "Principal Components PC1 to PC5",
     main="Variance vs Principal Components")
axis(1, at = c(1,2,3,4,5), labels = c(1,2,3,4,5))
```

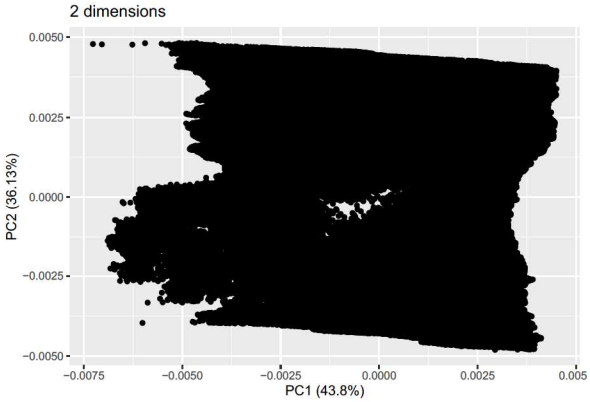



```
plot(pc1, type = "lines",
     main = "Variance vs Principal Componenets")
```



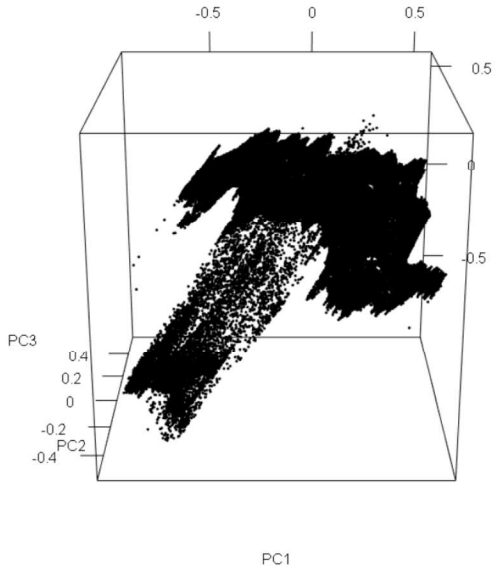
Here we see that the first 3 principal components account for most of the variation in the data. Hence the smallest reasonable value of D to present a good representation of the data is D = 3. D = 2 may also be used, however D = 3 gives a much better representation.

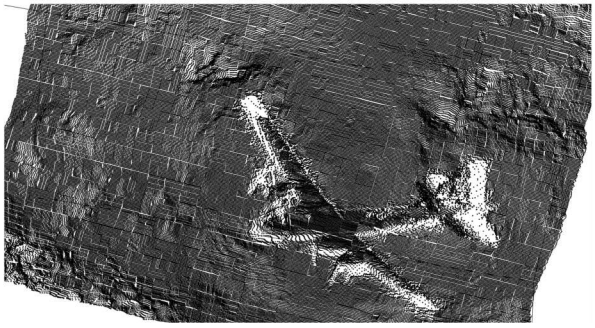
```
pca_df <- pc1$x[,1:3]

autoplot(pc1,
         main = "2 dimensions")
```


```
plot3d(pca_df[,1:3]) # 3D Plot
```

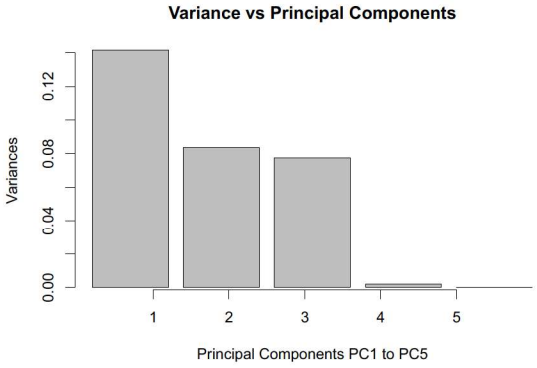
The above function 'plot3d' gives a 3D view of first 3 principal components. It is not visible in R markdown, hence screenshot are imported of the zoomed out 3D image and zoomed in 3D image.



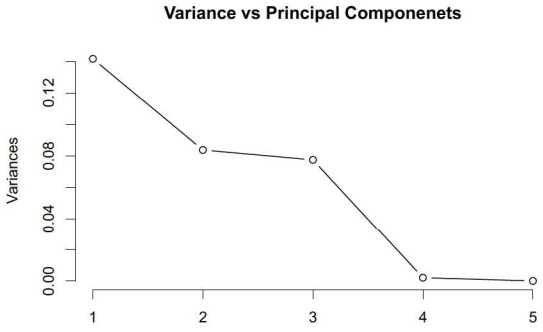


PCA on Bird image

```
pc_2 <- prcomp(normalized_matrix_2)

plot(pc_2,
     xlab = "Principal Components PC1 to PC5",
     main="Variance vs Principal Components")
axis(1, at = c(1,2,3,4,5), labels = c(1,2,3,4,5))
```

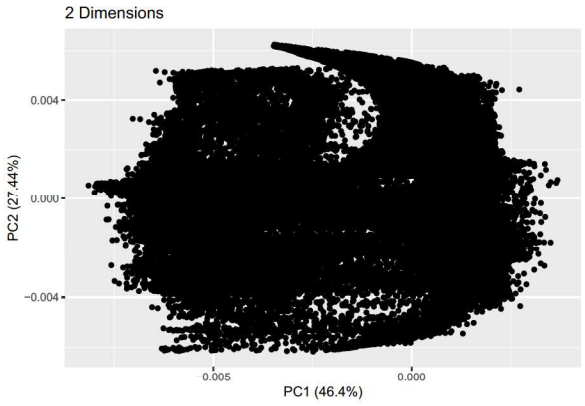



```
plot(pc_2, type = "lines",
     main = "Variance vs Principal Componenets")
```

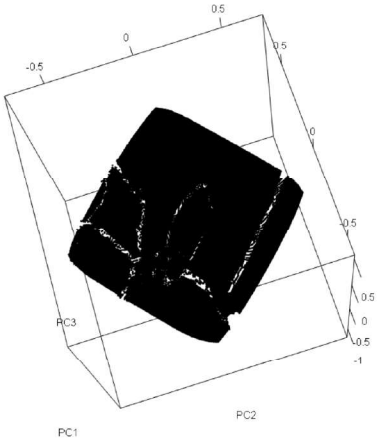
Here, similar to the plane image, we see that the first 3 principal components account for most of the variation in the data. Hence the smallest reasonable value of D to present a good representation of the data is D = 3. D = 2 may also be used, however D = 3 gives a much better representation.

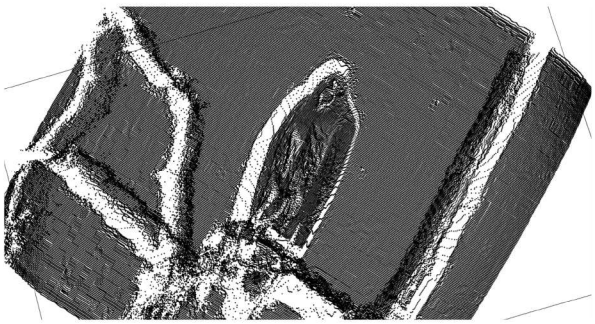
```
pca_df_2 <- pc_2$x[,1:3]

autoplot(pc_2,
         main = "2 Dimensions")
```


```
plot3d(pca_df_2[,1:3]) # 3D plot
```

The above function 'plot3d' gives a 3D view of first 3 principal components. It is not visible in R markdown, hence screenshot are imported of the zoomed out 3D image and zoomed in 3D image.





T-SNE Plane

```
plane_tsne <- normalized_matrix[sample(nrow(normalized_matrix), 1000),]
```

Perplexity = 30, Dimesnions = 2

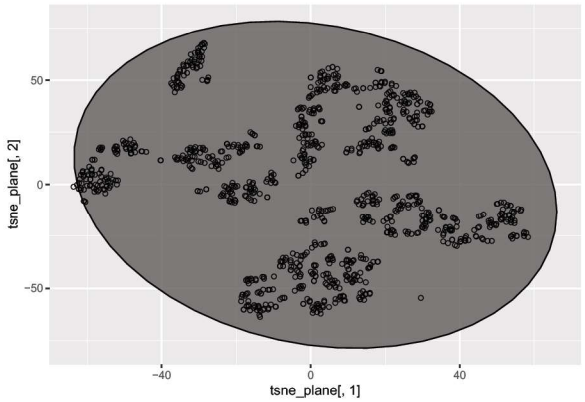
```
tsne_plane <- tsne(plane_tsne, k = 2, perplexity = 30, epoch = 500)
```

```
## sigma summary: Min. : 0.299517037910949 |1st Qu. : 0.403401427636214 |Median : 0.436405345288248 |Me
```

```
## Epoch: Iteration #500 error is: 0.310362064960431
```

```
## Epoch: Iteration #1000 error is: 0.303210849130399
```

```
ggplot(tsne_plane, aes(tsne_plane[,1], tsne_plane[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```



Perplexity = 5, Dimesnions = 2


```
tsne_plane_5 <- tsne(plane_tsne, k = 2, perplexity = 5)
```

```
## sigma summary: Min. : 0.114870344841091 |1st Qu. : 0.250262916125213 |Median : 0.291433427516082 |Me
```

```
## Epoch: Iteration #100 error is: 14.3719896268884
```

```
## Epoch: Iteration #200 error is: 1.08210174417629
```

```
## Epoch: Iteration #300 error is: 0.737158964772354
```

```
## Epoch: Iteration #400 error is: 0.574247126250254
```

```
## Epoch: Iteration #500 error is: 0.506512707949161
```

```
## Epoch: Iteration #600 error is: 0.468105445652011
```

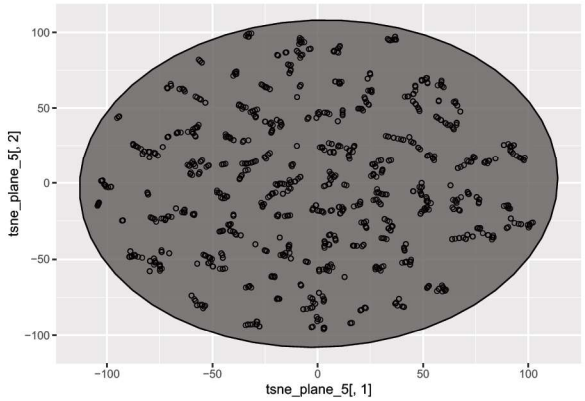
```
## Epoch: Iteration #700 error is: 0.443104387904279
```

```
## Epoch: Iteration #800 error is: 0.425276987107096
```

```
## Epoch: Iteration #900 error is: 0.411653548568094
```

```
## Epoch: Iteration #1000 error is: 0.400330660853039
```

```
ggplot(tsne_plane_5, aes(tsne_plane_5[,1], tsne_plane_5[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```



Perplexity = 50, Dimesnions = 2

```
tsne_plane_50 <- tsne(plane_tsne, k = 2, perplexity = 50)
```

```
## sigma summary: Min. : 0.344621859080952 |1st Qu. : 0.449566559203387 |Median : 0.478800852868747 |Me
```

```
## Epoch: Iteration #100 error is: 11.6728779266012
```

```
## Epoch: Iteration #200 error is: 0.347759462198227
```

```
## Epoch: Iteration #300 error is: 0.298772942105999
```

```
## Epoch: Iteration #400 error is: 0.290236020778999
```

```
## Epoch: Iteration #500 error is: 0.288547586343645
```

```
## Epoch: Iteration #600 error is: 0.287997546882067
```

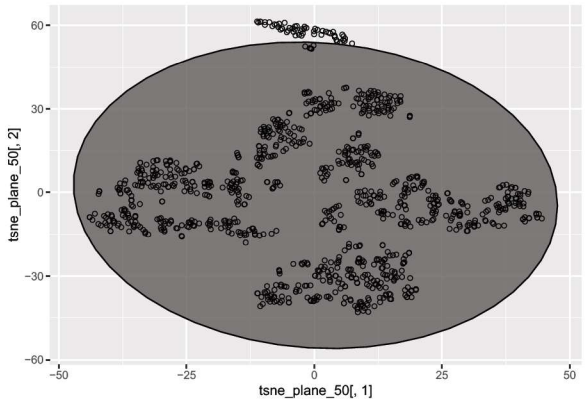
```
## Epoch: Iteration #700 error is: 0.287764067150249
```

```
## Epoch: Iteration #800 error is: 0.287636163337121
```

```
## Epoch: Iteration #900 error is: 0.287542549801056
```

```
## Epoch: Iteration #1000 error is: 0.287456957883659
```

```
ggplot(tsne_plane_50, aes(tsne_plane_50[,1], tsne_plane_50[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```

T-SNE Bird

```
bird_tsne <- normalized_matrix_2[sample(nrow(normalized_matrix), 1000),]
```

Perplexity = 30, Dimesnions = 2

```
tsne_bird <- tsne(bird_tsne, k = 2, perplexity = 30)
```

```
## sigma summary: Min. : 0.220461709177513 |1st Qu. : 0.376445362452227 |Median : 0.406803513841028 |Me
```

```
## Epoch: Iteration #100 error is: 12.2201054704474
```

```
## Epoch: Iteration #200 error is: 0.490166112157222
```

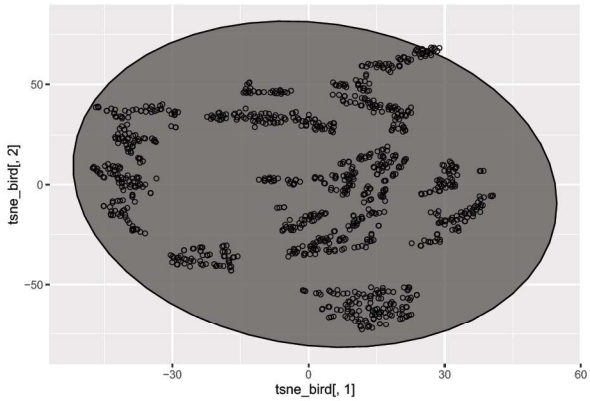
```
## Epoch: Iteration #300 error is: 0.405471499527768
```

```
## Epoch: Iteration #400 error is: 0.382500802761798
```

```
## Epoch: Iteration #500 error is: 0.374992020744206
```

```
## Epoch: Iteration #600 error is: 0.370802616757633

## Epoch: Iteration #700 error is: 0.367647323388406

## Epoch: Iteration #800 error is: 0.364544403234643

## Epoch: Iteration #900 error is: 0.360457658570239

## Epoch: Iteration #1000 error is: 0.353875772548557
```

```
ggplot(tsne_bird, aes(tsne_bird[,1], tsne_bird[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```



Perplexity = 5, Dimesnions = 2

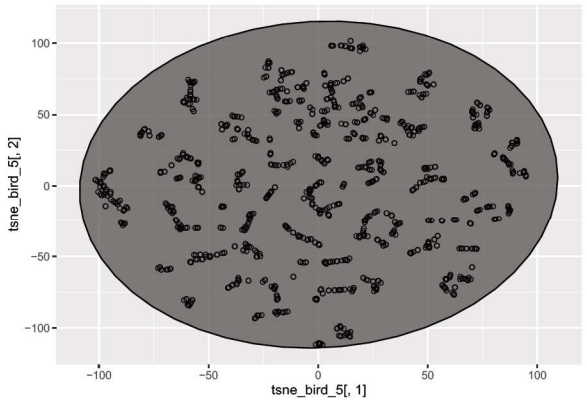
```
tsne_bird_5 <- tsne(bird_tsne, k = 2, perplexity = 5)
```

```
## sigma summary: Min. : 0.0804177495152935 |1st Qu. : 0.212604642434107 |Median : 0.257903576703926 |M

## Epoch: Iteration #100 error is: 14.4036585069944

## Epoch: Iteration #200 error is: 1.08178462856625
```



```
## Epoch: Iteration #300 error is: 0.747421557110705

## Epoch: Iteration #400 error is: 0.589121520578423

## Epoch: Iteration #500 error is: 0.522742958702706

## Epoch: Iteration #600 error is: 0.485656369257118

## Epoch: Iteration #700 error is: 0.46164358476976

## Epoch: Iteration #800 error is: 0.445052801550582

## Epoch: Iteration #900 error is: 0.432879924769027

## Epoch: Iteration #1000 error is: 0.42343662160762
```

```
ggplot(tsne_bird_5, aes(tsne_bird_5[,1], tsne_bird_5[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```

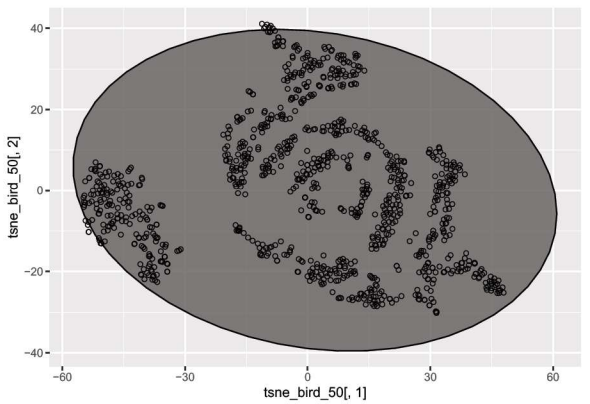


Perplexity = 50, Dimesnions = 2


```
tsne_bird_50 <- tsne(bird_tsne, k = 2, perplexity = 50)
```

```
## sigma summary: Min. : 0.270519415789701 |1st Qu. : 0.425379185398806 |Median : 0.45592657727621 |Mea

## Epoch: Iteration #100 error is: 11.5225587972156

## Epoch: Iteration #200 error is: 0.336677347847931

## Epoch: Iteration #300 error is: 0.292425439210568

## Epoch: Iteration #400 error is: 0.283039927569602

## Epoch: Iteration #500 error is: 0.280802115117103

## Epoch: Iteration #600 error is: 0.27994959509906

## Epoch: Iteration #700 error is: 0.279528046892077

## Epoch: Iteration #800 error is: 0.279280575236508

## Epoch: Iteration #900 error is: 0.279113940411927

## Epoch: Iteration #1000 error is: 0.278991022480306
```

```
ggplot(tsne_bird_50, aes(tsne_bird_50[,1], tsne_bird_50[,2])) +
  stat_ellipse(geom = "polygon", col = "black", alpha = 0.5) +
  geom_point(shape = 21, col = "black")
```

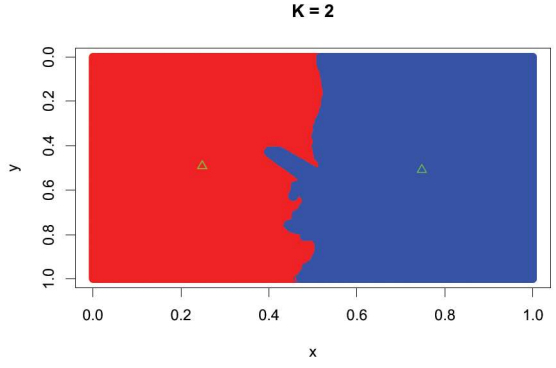


With increase in the perplexity value, it seems that the number of unique clusters is decreasing.

Homework 5

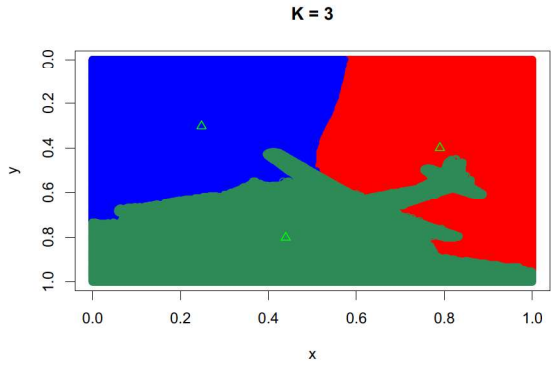
K-MEANS PLANE

```
kmeans_2 <- kmeans(normalized_matrix, centers = 2)

plot(normalized_matrix[kmeans_2$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix[,1]), max(normalized_matrix[,1])),
     ylim = rev(c(min(normalized_matrix[,2]), max(normalized_matrix[,2]))),
     main = "K = 2"
)
points(normalized_matrix[kmeans_2$cluster ==2, ],
       col = "blue")
points(kmeans_2$centers, pch=2, col = "green")
```




```
kmeans_3 <- kmeans(normalized_matrix, centers = 3)

plot(normalized_matrix[kmeans_3$cluster == 1, ],
     col = "red",
```

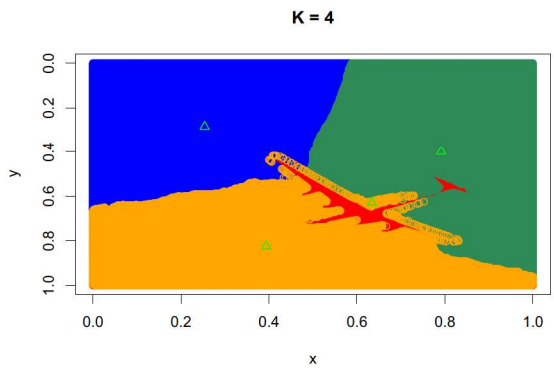
```
     xlim = c(min(normalized_matrix[,1]), max(normalized_matrix[,1])),
     ylim = rev(c(min(normalized_matrix[,2]), max(normalized_matrix[,2]))),
     main = "K = 3"
     )
points(normalized_matrix[kmeans_3$cluster ==2, ],
       col = "blue")
points(normalized_matrix[kmeans_3$cluster == 3, ],
       col = "seagreen")
points(kmeans_3$centers, pch=2, col = "green")
```




```
kmeans_4 <- kmeans(normalized_matrix, centers = 4)

#kmeans_4$centers

#kmeans_4$cluster

plot(normalized_matrix[kmeans_4$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix[,1]), max(normalized_matrix[,1])),
     ylim = rev(c(min(normalized_matrix[,2]), max(normalized_matrix[,2]))),
     main = "K = 4"
)
points(normalized_matrix[kmeans_4$cluster ==2, ],
       col = "blue")
points(normalized_matrix[kmeans_4$cluster == 3, ],
```



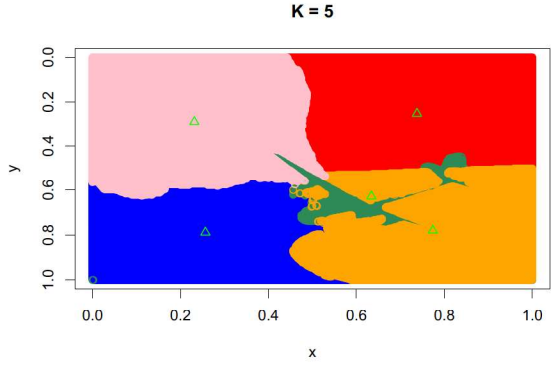
```
       col = "seagreen")
points(normalized_matrix[kmeans_4$cluster == 4, ],
       col = "orange")
points(kmeans_4$centers, pch=2, col = "green")
```




```
kmeans_5 <- kmeans(normalized_matrix, centers = 5)

#kmeans_5$centers

#kmeans_5$cluster

plot(normalized_matrix[kmeans_5$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix[,1]), max(normalized_matrix[,1])),
     ylim = rev(c(min(normalized_matrix[,2]), max(normalized_matrix[,2]))),
     main = "K = 5"
)
points(normalized_matrix[kmeans_5$cluster ==2, ],
       col = "blue")
points(normalized_matrix[kmeans_5$cluster == 3, ],
       col = "seagreen")
points(normalized_matrix[kmeans_5$cluster == 4, ],
       col = "orange")
points(normalized_matrix[kmeans_5$cluster == 5, ],
       col = "pink")
```



```
points(kmeans_5$centers, pch=2, col = "green")
```



Finding Silhouette Coefficient and CH Index

```
c <- normalized_matrix[sample(nrow(normalized_matrix), 10000),]
c <- as.matrix(c)

silhouette_score1 <- function(k1){
  km1 <- kmeans(c, centers = k1, nstart=25)
  ss1 <- silhouette(km1$cluster, dist(c))
  chi <- get_CH(c, km1$cluster, disMethod = "Euclidean")
  mean(ss1[, 3])
  print("K = ")
  print(k1)
  print("Silhouette COeff: ")
  print(mean(ss1[,3]))
  print("CH index: ")
  print(chi)
}
k1 <- 2:5
avg_sil1 <- sapply(k1, silhouette_score1)
```

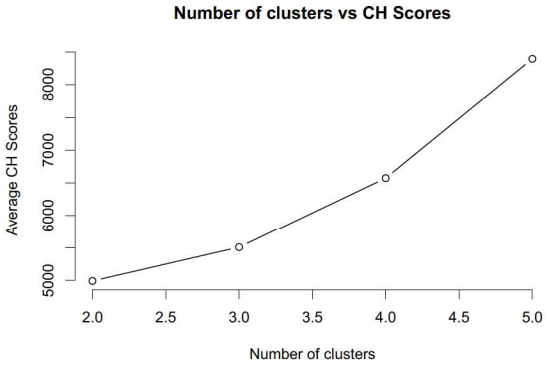
```
## [1] "K = "
```



```
## [1] 2
## [1] "Silhouette COeff: "
## [1] 0.338238
## [1] "CH index: "
## [1] 4992.299
## [1] "K = "
## [1] 3
## [1] "Silhouette COeff: "
## [1] 0.3442866
## [1] "CH index: "
## [1] 5508.816
## [1] "K = "
## [1] 4
## [1] "Silhouette COeff: "
## [1] 0.3870141
## [1] "CH index: "
## [1] 6573.29
## [1] "K = "
## [1] 5
## [1] "Silhouette COeff: "
## [1] 0.4125368
## [1] "CH index: "
## [1] 8392.92
```

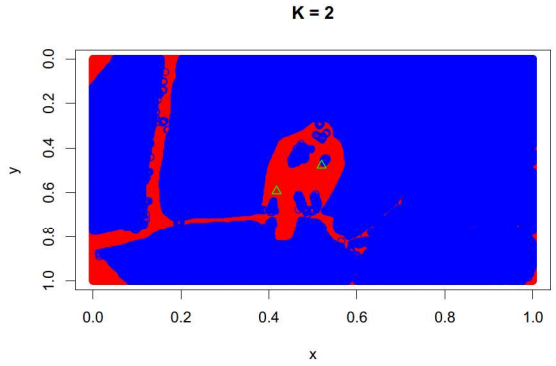
K = 5 has the highest silhouette Coefficient (0.4081913) and the highest CH index (8248.353). Hence we can conclude that K = 5 is the optimal number of clusters for this image. Also, from the visualization it is evident that K = 5 is the optimal number of clusters since the image is nicely defined.

```
plot(k1, type='b', avg_sil1, xlab='Number of clusters', ylab='Average CH Scores', frame=FALSE,
     main = "Number of clusters vs CH Scores")
```

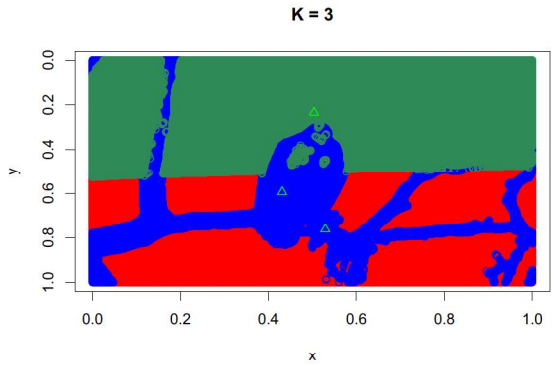


K-MEANS BIRD

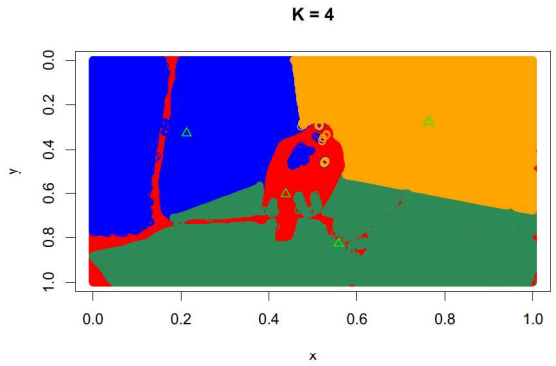
```r
kmeans_2b <- kmeans(normalized_matrix_2, centers = 2)

plot(normalized_matrix_2[kmeans_2b$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix_2[,1]), max(normalized_matrix_2[,1])),
     ylim = rev(c(min(normalized_matrix_2[,2]), max(normalized_matrix_2[,2]))),
     main = "K = 2"
)
points(normalized_matrix_2[kmeans_2b$cluster ==2, ],
       col = "blue")
points(kmeans_2b$centers, pch=2, col = "green")
```

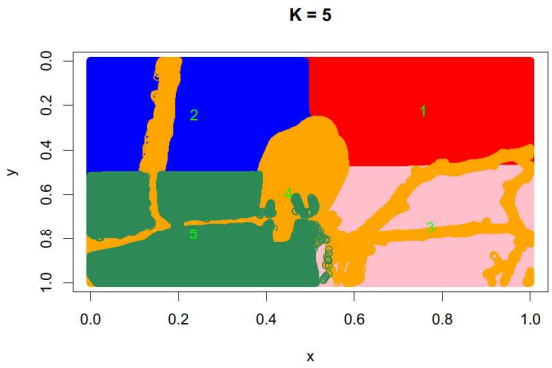

```r
kmeans_3b <- kmeans(normalized_matrix_2, centers = 3)

plot(normalized_matrix_2[kmeans_3b$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix_2[,1]), max(normalized_matrix_2[,1])),
     ylim = rev(c(min(normalized_matrix_2[,2]), max(normalized_matrix_2[,2]))),
     main = "K = 3"
)
points(normalized_matrix_2[kmeans_3b$cluster ==2, ],
       col = "blue")
points(normalized_matrix_2[kmeans_3b$cluster == 3, ],
       col = "seagreen")
points(kmeans_3b$centers, pch=2, col = "green")
```


```r
kmeans_4b <- kmeans(normalized_matrix_2, centers = 4)

#kmeans_4b$centers

#kmeans_4b$cluster

plot(normalized_matrix_2[kmeans_4b$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix_2[,1]), max(normalized_matrix_2[,1])),
     ylim = rev(c(min(normalized_matrix_2[,2]), max(normalized_matrix_2[,2]))),
     main = "K = 4"
)
points(normalized_matrix_2[kmeans_4b$cluster ==2, ],
       col = "blue")
points(normalized_matrix_2[kmeans_4b$cluster == 3, ],
       col = "seagreen")
points(normalized_matrix_2[kmeans_4b$cluster == 4, ],
       col = "orange")
points(kmeans_4b$centers, pch=2, col = "green")
```


```r
kmeans_5b <- kmeans(normalized_matrix_2, centers = 5)

#kmeans_5b$centers

#kmeans_5b$cluster

plot(normalized_matrix_2[kmeans_5b$cluster == 1, ],
     col = "red",
     xlim = c(min(normalized_matrix_2[,1]), max(normalized_matrix_2[,1])),
     ylim = rev(c(min(normalized_matrix_2[,2]), max(normalized_matrix_2[,2]))),
     main = "K = 5"
)
points(normalized_matrix_2[kmeans_5b$cluster ==2, ],
       col = "blue")
points(normalized_matrix_2[kmeans_5b$cluster == 3, ],
       col = "pink")
points(normalized_matrix_2[kmeans_5b$cluster == 4, ],
       col = "orange")
points(normalized_matrix_2[kmeans_5b$cluster == 5, ],
       col = "seagreen")
text(kmeans_5b$centers, labels = c('1','2','3','4','5'),
     col = "green")
```

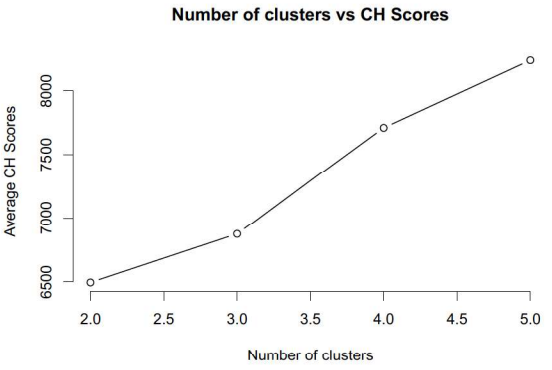


Finding Silhouette Coefficient

```
b <- normalized_matrix_2[sample(nrow(normalized_matrix_2), 10000),]
b <- as.matrix(b)

silhouette_score <- function(k){
  km <- kmeans(b, centers = k, nstart=25)
  ss <- silhouette(km$cluster, dist(b))
  chi <- get_CH(b, km$cluster, disMethod = "Euclidean")
  mean(ss[, 3])
  print("K = ")
  print(k)
  print("Silhouette COeff: ")
  print(mean(ss[,3]))
  print("CH index: ")
  print(chi)
}
k <- 2:5
avg_sil <- sapply(k, silhouette_score)
```

```
## [1] "K = "
## [1] 2
## [1] "Silhouette COeff: "
## [1] 0.4605161
## [1] "CH index: "
## [1] 6494.909
## [1] "K = "
## [1] 3
## [1] "Silhouette COeff: "
## [1] 0.3716012
## [1] "CH index: "
## [1] 6873.781

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 500000)

## [1] "K = "
## [1] 4
## [1] "Silhouette COeff: "
## [1] 0.3919332
## [1] "CH index: "
## [1] 7710.329
## [1] "K = "
## [1] 5
## [1] "Silhouette COeff: "
## [1] 0.4020038
## [1] "CH index: "
## [1] 8239.812
```

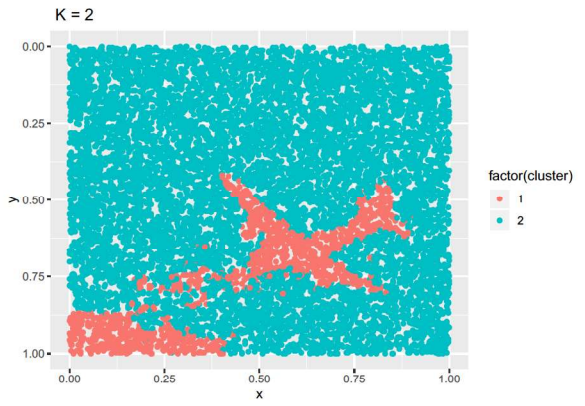
```
plot(k, type='b', avg_sil, xlab='Number of clusters', ylab='Average CH Scores', frame=FALSE,
     main = "Number of clusters vs CH Scores")
```

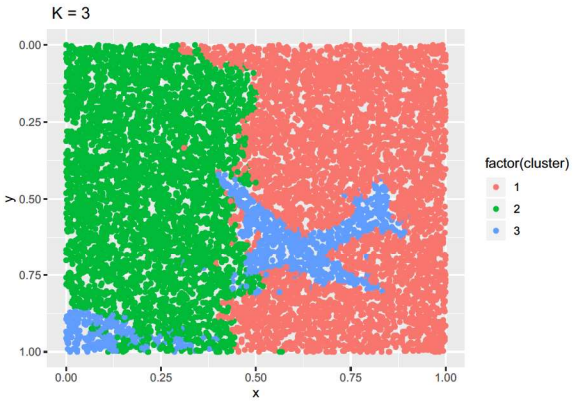
For K = 5, we have the highest CH scores and the Silhouette Coefficient is also reasonably high. The image also seems to be nicely defined, hecne we will select K=5 as the optimal number of clusters.

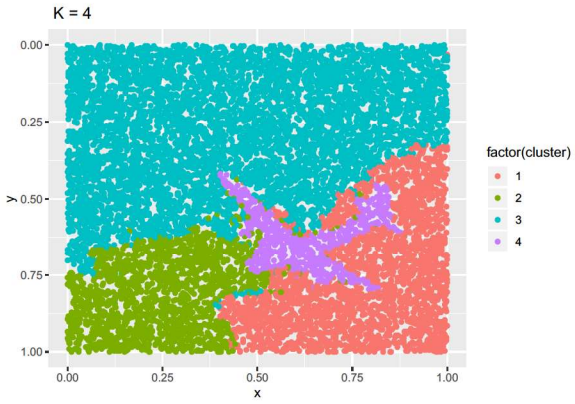
GMM Plane

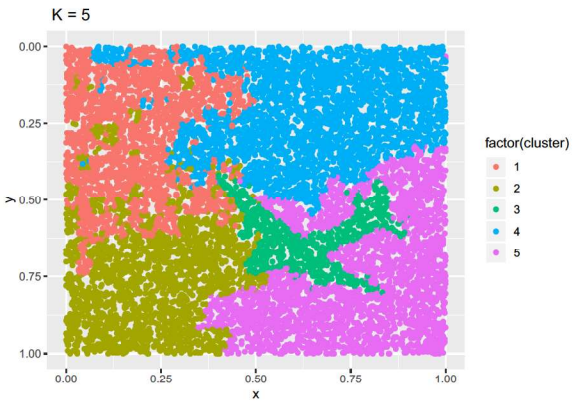
```
fit2 = Mclust(c, G=2)

gmm_plane_2 <- as.data.frame(c)

gmm_plane_2 <- mutate(gmm_plane_2, cluster = fit2$classification)

ggplot(gmm_plane_2, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() +
  labs(title = " K = 2")
```


```
fit3 = Mclust(c, G=3)

gmm_plane_3 <- as.data.frame(c)

gmm_plane_3 <- mutate(gmm_plane_3, cluster = fit3$classification)

ggplot(gmm_plane_3, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse()+
  labs(title = " K = 3")
```







Silhouette Coefficient and CH index

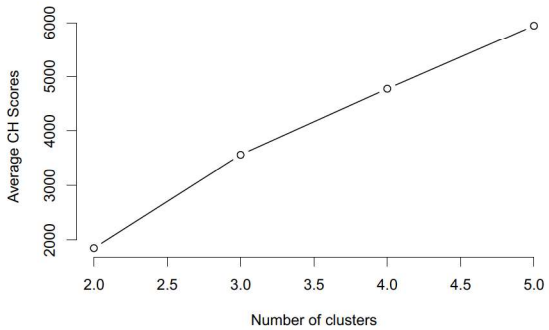
```
silhouette_score1 <- function(k1){
  km1 <- Mclust(c, G=k1)
  ss1 <- silhouette(km1$classification, dist(c))
  chi <- get_CH(c, km1$classification, disMethod = "Euclidean")
  mean(ss1[, 3])
  print("K = ")
  print(k1)
  print("Sillhouette COeff: ")
  print(mean(ss1[,3]))
  print("CH index: ")
  print(chi)
}
k1 <- 2:5
avg_sil_gmm_plane <- sapply(k1, silhouette_score1)
```

```
## [1] "K = "
## [1] 2
## [1] "Sillhouette COeff: "
## [1] 0.373821
## [1] "CH index: "
## [1] 1850.645
## [1] "K = "
## [1] 3
## [1] "Sillhouette COeff: "
```



```
## [1] 0.3040115
## [1] "CH index: "
## [1] 3562.707
## [1] "K = "
## [1] 4
## [1] "Sillhouette COeff: "
## [1] 0.3353574
## [1] "CH index: "
## [1] 4775.569
## [1] "K = "
## [1] 5
## [1] "Sillhouette COeff: "
## [1] 0.3600235
## [1] "CH index: "
## [1] 5946.586
```

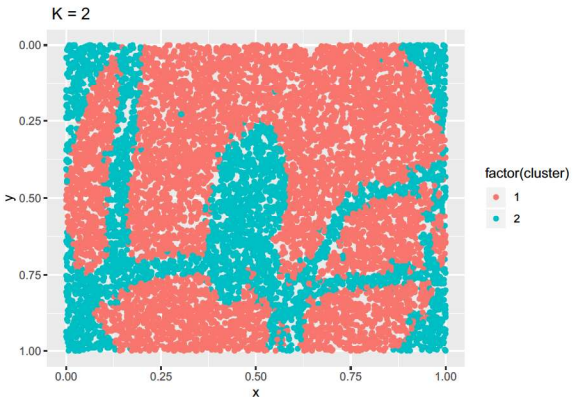
```
plot(k1, type='b', avg_sil_gmm_plane, xlab='Number of clusters', ylab='Average CH Scores', frame=FALSE)
```



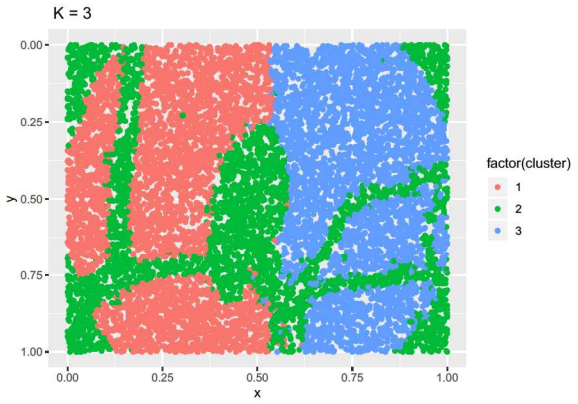
We see that the optimal number of clusters for GMM is K = 5. The Silhouette coefficient and CH index are highest for K = 5. The visualization also seems to be well defined.

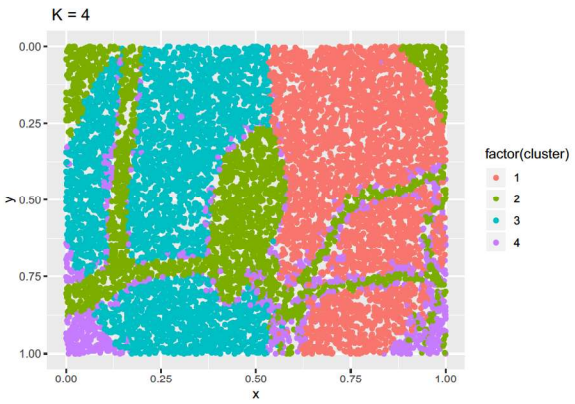
GMM Bird

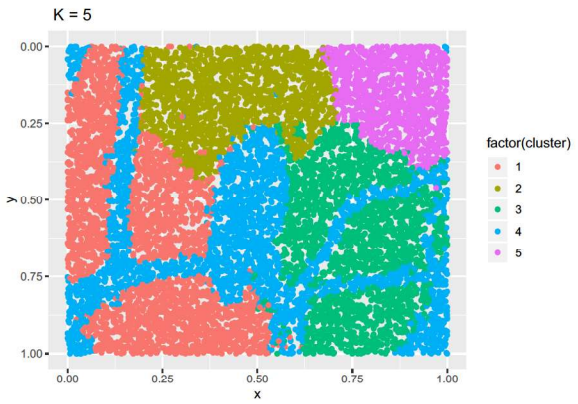
```
fit2b = Mclust(b, G=2)

gmm_bird_2 <- as.data.frame(b)

gmm_bird_2 <- mutate(gmm_bird_2, cluster = fit2b$classification)

ggplot(gmm_bird_2, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() +
  labs(title = " K = 2")
```




```
fit3b = Mclust(b, G=3)

gmm_bird_3 <- as.data.frame(b)

gmm_bird_3 <- mutate(gmm_bird_3, cluster = fit3b$classification)

ggplot(gmm_bird_3, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() +
  labs(title = " K = 3")
```







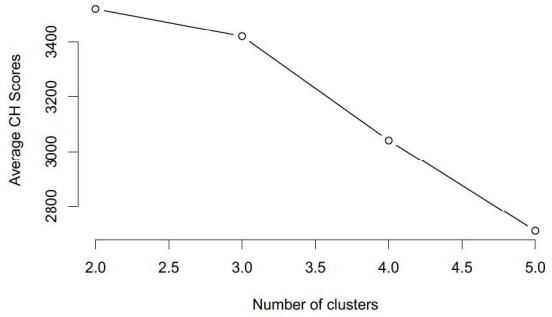
Silhouette Coefficient and CH index

```
silhouette_score1 <- function(k1){
  km1 <- Mclust(b, G=k1)
  ss1 <- silhouette(km1$classification, dist(b))
  chi <- get_CH(b, km1$classification, disMethod = "Euclidean")
  mean(ss1[, 3])
  print("K = ")
  print(k1)
  print("Sillhouette COeff: ")
  print(mean(ss1[,3]))
  print("CH index: ")
  print(chi)
}
k1 <- 2:5
avg_sil_gmm_plane <- sapply(k1, silhouette_score1)
```

```
## [1] "K = "
## [1] 2
## [1] "Sillhouette COeff: "
## [1] 0.3079633
## [1] "CH index: "
## [1] 3519.671
## [1] "K = "
## [1] 3
## [1] "Sillhouette COeff: "
```

```
## [1] 0.2481392
## [1] "CH index: "
## [1] 3419.84
## [1] "K = "
## [1] 4
## [1] "Sillhouette COeff: "
## [1] 0.2006313
## [1] "CH index: "
## [1] 3041.88
## [1] "K = "
## [1] 5
## [1] "Sillhouette COeff: "
## [1] 0.1908573
## [1] "CH index: "
## [1] 2709.01
```

```
plot(k1, type='b', avg_sil_gmm_plane, xlab='Number of clusters', ylab='Average CH Scores', frame=FALSE)
```

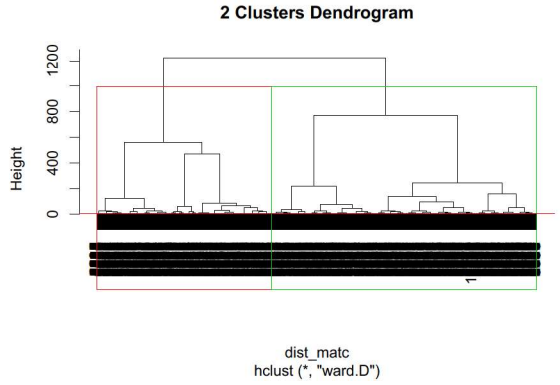


We see that the optimal number of clusters for GMM for the bird image is K = 2. The Silhouette coefficient and CH index are highest for K = 2. The visualization also seems to be well defined for 2 clusters.

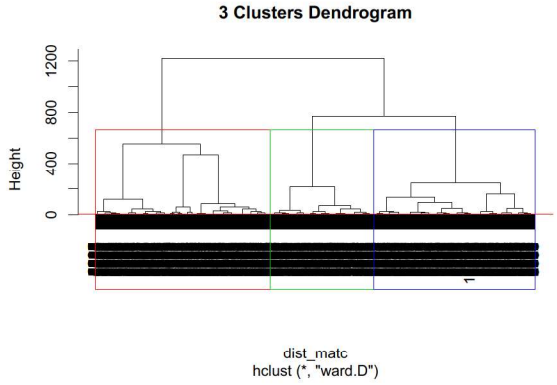
Hierarchical Plane


```
dist_matc <- dist(c, method = 'euclidean')
hclust_ward <- hclust(dist_matc, method = 'ward')
```

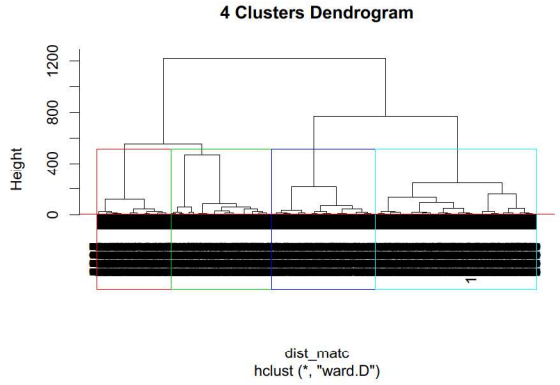
```
cut_ward2 <- cutree(hclust_ward, k=2)
plot(hclust_ward,
     main = "2 Clusters Dendrogram")
rect.hclust(hclust_ward, k = 2, border = 2:6)
abline(h=2, col = 'red')
```



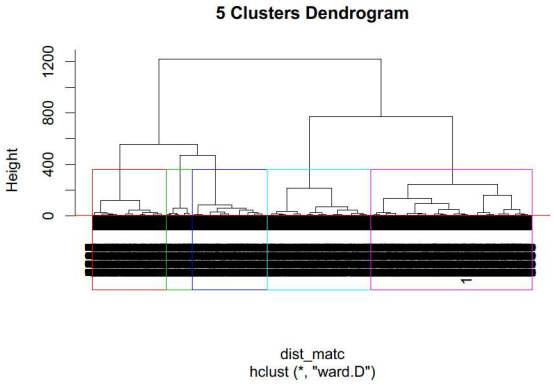

```
cut_ward3 <- cutree(hclust_ward, k=3)
plot(hclust_ward,
     main = "3 Clusters Dendrogram")
rect.hclust(hclust_ward, k = 3, border = 2:6)
abline(h=3, col = 'red')
```



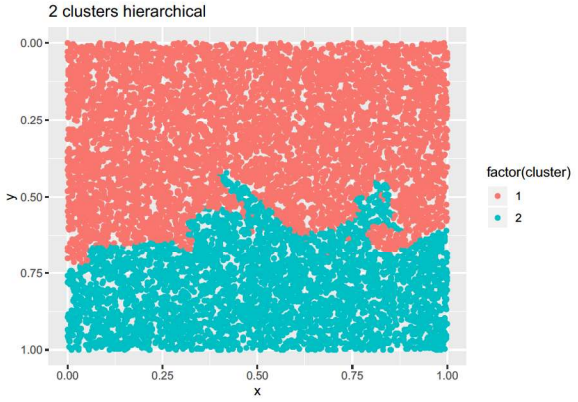

```
cut_ward4 <- cutree(hclust_ward, k=4)
plot(hclust_ward,
     main = "4 Clusters Dendrogram")
rect.hclust(hclust_ward, k = 4, border = 2:6)
abline(h=4, col = 'red')
```



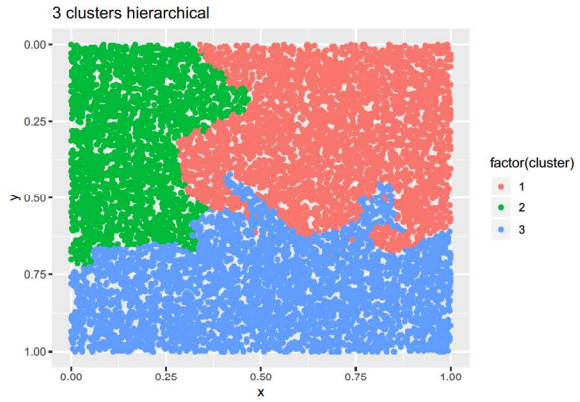

```
cut_ward5 <- cutree(hclust_ward, k=5)
plot(hclust_ward,
     main = "5 Clusters Dendrogram")
rect.hclust(hclust_ward, k = 5, border = 2:6)
abline(h=5, col = 'red')
```



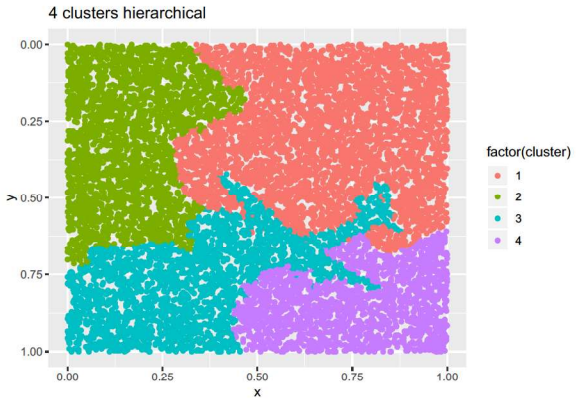

```
c2 <- as.data.frame(c)
c2 <- mutate(c2, cluster = cut_ward2)

ggplot(c2, aes(x = x, y = y, color = factor(cluster))) +
  geom_point() + labs(title = "2 clusters hierarchical")+ scale_y_reverse()
```

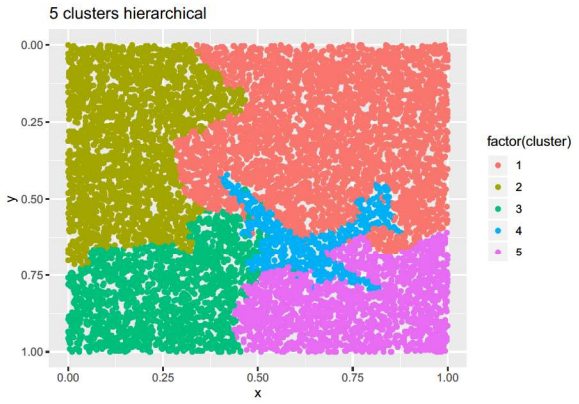

```
c3 <- as.data.frame(c)
c3 <- mutate(c3, cluster = cut_ward3)

ggplot(c3, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ labs(title = "3 clusters hierarchical")+ scale_y_reverse()
```


```
c4 <- as.data.frame(c)
c4 <- mutate(c4, cluster = cut_ward4)

ggplot(c4, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ labs(title = "4 clusters hierarchical")+ scale_y_reverse()
```


```
c5 <- as.data.frame(c)
c5 <- mutate(c5, cluster = cut_ward5)

ggplot(c5, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ labs(title = "5 clusters hierarchical")+ scale_y_reverse()
```



Silhouette Coefficients and CH Indices

```
print("Clusters = 2 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 2 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(c2$cluster, dist(c))[,3]))
```

```
## [1] 0.3365496
```

```
print(get_CH(c, c2$cluster, disMethod = "Euclidean"))
```

```
## [1] 4871.74
```

```
print("Clusters = 3 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 3 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(c3$cluster, dist(c))[,3]))
```

```
## [1] 0.3123385
```



```
print(get_CH(c, c3$cluster, disMethod = "Euclidean"))
```

```
## [1] 4833.924
```

```
print("Clusters = 4 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 4 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(c4$cluster, dist(c))[,3]))
```

```
## [1] 0.3378521
```

```
print(get_CH(c, c4$cluster, disMethod = "Euclidean"))
```

```
## [1] 5358.462
```

```
print("Clusters = 5 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 5 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(c5$cluster, dist(c))[,3]))
```

```
## [1] 0.3772365
```

```
print(get_CH(c, c5$cluster, disMethod = "Euclidean"))
```

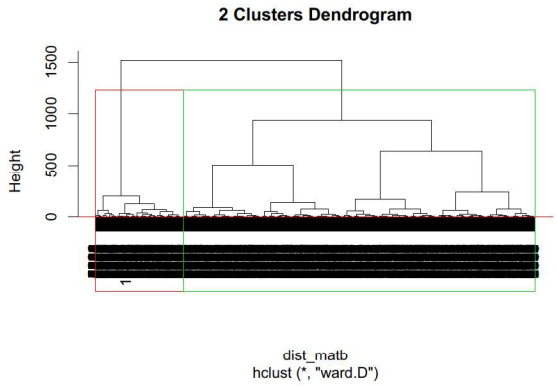
```
## [1] 7044.127
```

We see that the optimal number of clusters is 5 since it gives the largest Silhouette Coefficient and CH Index. Also, the visualization looks well defined.

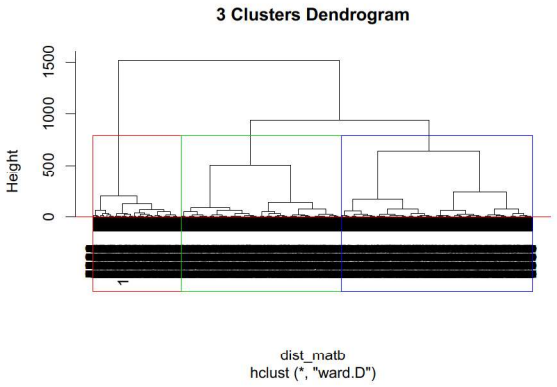
Hierarchical Clustering Bird

```
dist_matb <- dist(b, method = 'euclidean')
hclust_wardb <- hclust(dist_matb, method = 'ward')
```

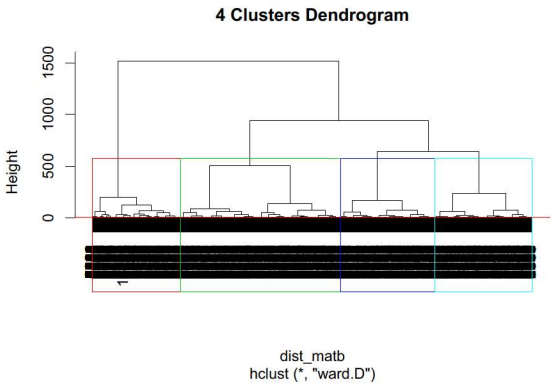
```
cut_ward2b <- cutree(hclust_wardb, k=2)
plot(hclust_wardb,
     main = "2 Clusters Dendrogram")
rect.hclust(hclust_wardb, k = 2, border = 2:6)
abline(h=2, col = 'red')
```

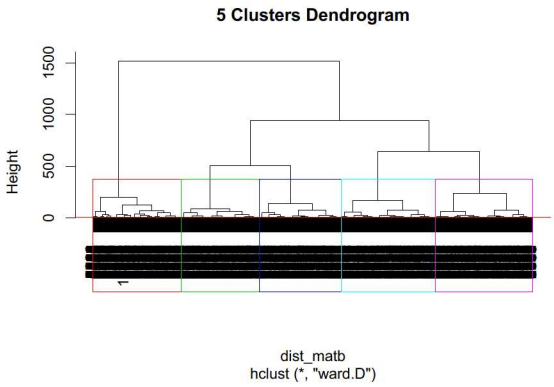

```
cut_ward3b <- cutree(hclust_wardb, k=3)
plot(hclust_wardb,
     main = "3 Clusters Dendrogram")
rect.hclust(hclust_wardb, k = 3, border = 2:6)
abline(h=3, col = 'red')
```

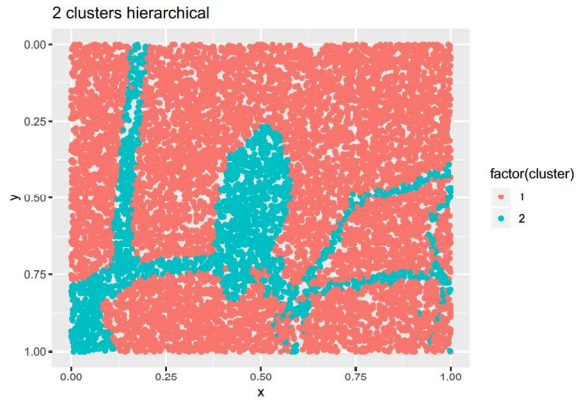

```
cut_ward4b <- cutree(hclust_wardb, k=4)
plot(hclust_wardb,
     main = "4 Clusters Dendrogram")
rect.hclust(hclust_wardb, k = 4, border = 2:6)
abline(h=4, col = 'red')
```



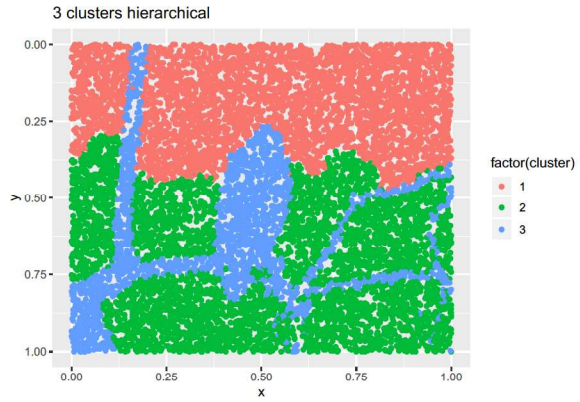

```
cut_ward5b <- cutree(hclust_wardb, k=5)
plot(hclust_wardb,
     main = "5 Clusters Dendrogram")
rect.hclust(hclust_wardb, k = 5, border = 2:6)
abline(h=5, col = 'red')
```

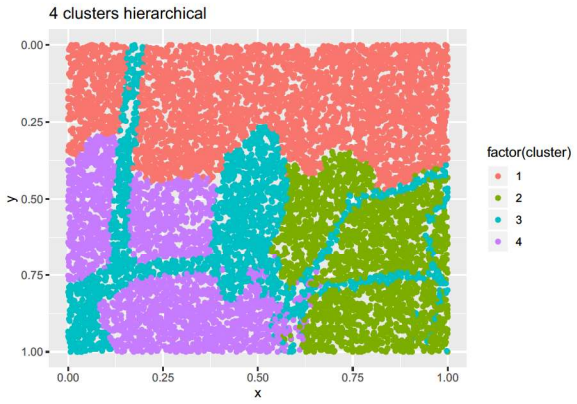

```
b2 <- as.data.frame(b)
b2 <- mutate(b2, cluster = cut_ward2b)

ggplot(b2, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() + labs(title = "2 clusters hierarchical")
```

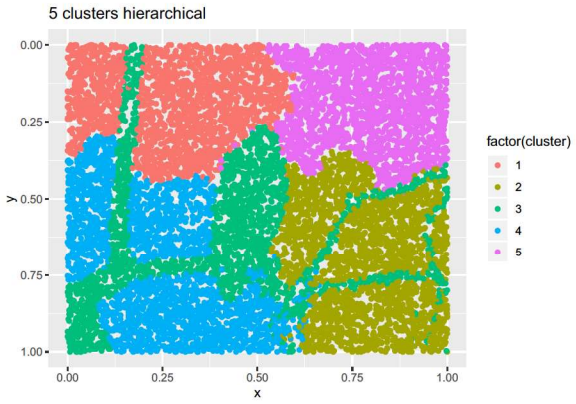

```
b3 <- as.data.frame(b)
b3 <- mutate(b3, cluster = cut_ward3b)

ggplot(b3, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() + labs(title = "3 clusters hierarchical")
```


```
b4 <- as.data.frame(b)
b4 <- mutate(b4, cluster = cut_ward4b)

ggplot(b4, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() + labs(title = "4 clusters hierarchical")
```




```
b5 <- as.data.frame(b)
b5 <- mutate(b5, cluster = cut_ward5b)

ggplot(b5, aes(x = x, y = y, color = factor(cluster))) +
  geom_point()+ scale_y_reverse() + labs(title = "5 clusters hierarchical")
```

Silhouette Coefficients and CH Indices

```
print("Clusters = 2 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 2 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(b2$cluster, dist(b))[,3]))
```

```
## [1] 0.4485325
```

```
print(get_CH(b, b2$cluster, disMethod = "Euclidean"))
```

```
## [1] 6250.304
```

```
print("Clusters = 3 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 3 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(b3$cluster, dist(b))[,3]))
```

```
## [1] 0.339672
```



```
print(get_CH(b, b3$cluster, disMethod = "Euclidean"))
```

```
## [1] 6230.889
```

```
print("Clusters = 4 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 4 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(b4$cluster, dist(b))[,3]))
```

```
## [1] 0.341644
```

```
print(get_CH(b, b4$cluster, disMethod = "Euclidean"))
```

```
## [1] 6329.484
```

```
print("Clusters = 5 , Silhouette Coeff, CH Index")
```

```
## [1] "Clusters = 5 , Silhouette Coeff, CH Index"
```

```
print(mean(silhouette(b5$cluster, dist(b))[,3]))
```

```
## [1] 0.3597828
```

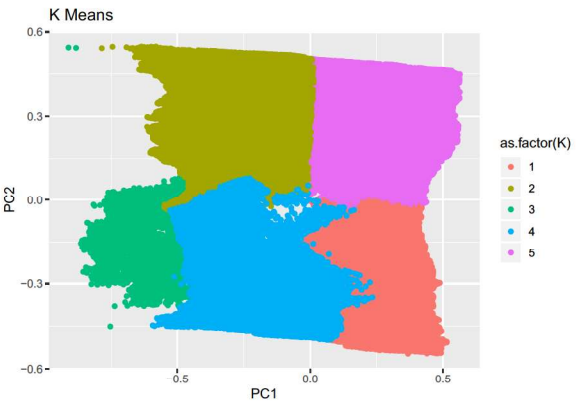
```
print(get_CH(b, b5$cluster, disMethod = "Euclidean"))
```

```
## [1] 6748.169
```

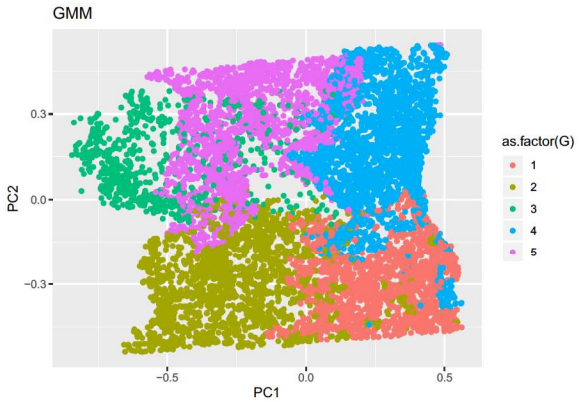
The Silhouette coefficient and CH Index are largest for 2 clusters in hierarchical clustering and also the image is nicely defined. Hence C = 2 is the optimal number of clusters for hierarchical clustering.

K Means, GMM, Hierarchical on PCA result Plane

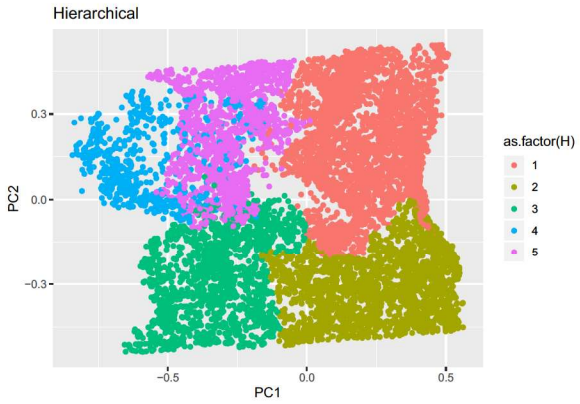
```
pc11 <- as.data.frame(pc1$x)
pc11 <- mutate(pc11, "K" = kmeans_5$cluster)

ggplot(pc11, aes(x = PC1, y = PC2, color = as.factor(K)))+ labs(title = "K Means") + geom_point()
```


```
pcone <- prcomp(c)

pc11_b <- as.data.frame(pcone$x)
pc11_b <- mutate(pc11_b, "G" = gmm_plane_5$cluster, "H" = c5$cluster)

ggplot(pc11_b, aes(x = PC1, y = PC2, color = as.factor(G))) + labs(title = "GMM") + geom_point()
```

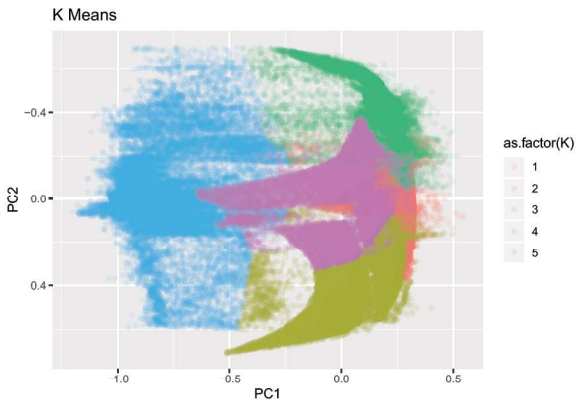



```
ggplot(pc11_b, aes(x = PC1, y = PC2, color = as.factor(H))) + labs(title = "Hierarchical") + geom_point
```

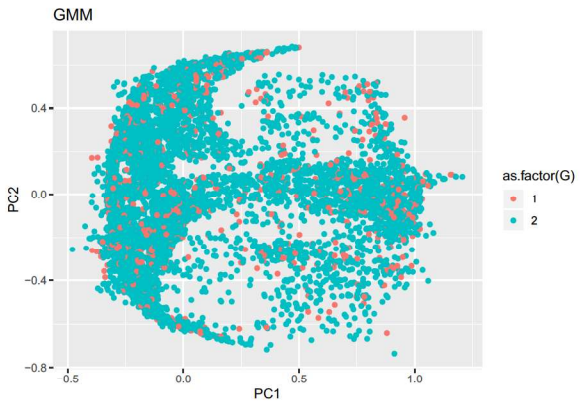



K Means, GMM, Hierarchical on PCA result Bird

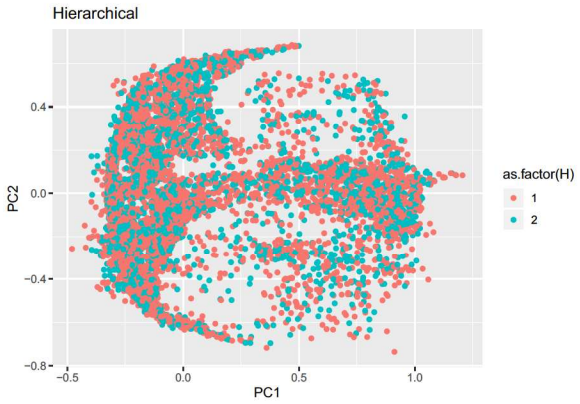
```
pc11b <- as.data.frame(pc_2$x)
pc11b <- mutate(pc11b, "K" = kmeans_5b$cluster)

ggplot(pc11b, aes(x = PC1, y = PC2, color = as.factor(K)))+ labs(title = "K Means") +
  geom_point(alpha = 0.1)+ scale_y_reverse()
```


```
pconeb <- prcomp(b)

pc11_bb <- as.data.frame(pconeb$x)
pc11_bb <- mutate(pc11_bb, "G" = gmm_plane_2$cluster, "H" = c2$cluster)

ggplot(pc11_bb, aes(x = PC1, y = PC2, color = as.factor(G))) + labs(title = "GMM") + geom_point()
```

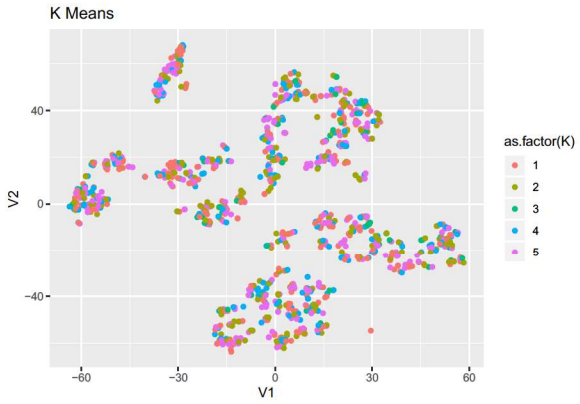




```
ggplot(pc11_bb, aes(x = PC1, y = PC2, color = as.factor(H))) + labs(title = "Hierarchical") + geom_poin
```

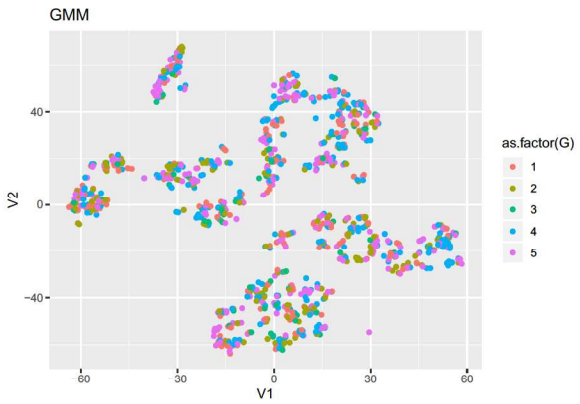


K Means, GMM, Hierarchical on TSNE result Plane

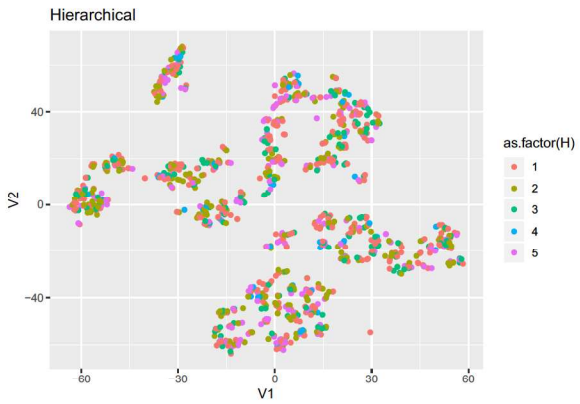
```
#plane_tsne <- normalized_matrix[sample(nrow(normalized_matrix), 1000),]

k5 <- as.data.frame(kmeans_5$cluster)
k5 <- k5[sample(nrow(k5), 1000),]

tsne_plane <- as.data.frame(tsne_plane)
tsne_plane <- mutate(tsne_plane, "K" = k5)

g5 <- as.data.frame(gmm_plane_5$cluster)
g5 <- g5[sample(nrow(g5), 1000),]

tsne_plane <- mutate(tsne_plane, "G" = g5)

h5 <- as.data.frame(c5$cluster)
h5 <- h5[sample(nrow(h5), 1000),]

tsne_plane <- mutate(tsne_plane, "H" = h5)

ggplot(tsne_plane, aes(x = V1, y = V2, color = as.factor(K)))+ labs(title = "K Means") + geom_point()
```


```
ggplot(tsne_plane, aes(x = V1, y = V2, color = as.factor(G))) + labs(title = "GMM") + geom_point()
```

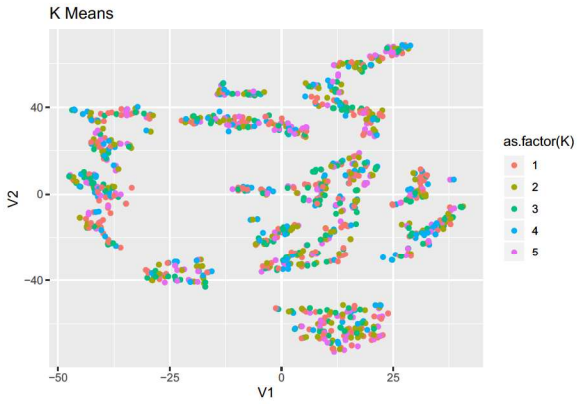




```
ggplot(tsne_plane, aes(x = V1, y = V2, color = as.factor(H))) + labs(title = "Hierarchical") + geom_poi
```

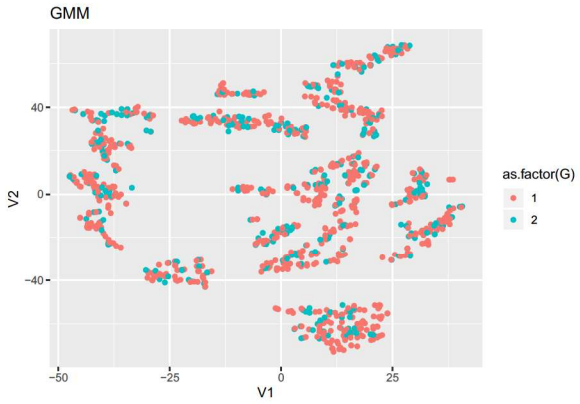



K Means, GMM, Hierarchical on TSNE result Bird

```
#plane_tsne <- normalized_matrix[sample(nrow(normalized_matrix), 1000),]

k5b <- as.data.frame(kmeans_5b$cluster)
k5b <- k5b[sample(nrow(k5b), 1000),]

tsne_bird <- as.data.frame(tsne_bird)
tsne_bird <- mutate(tsne_bird, "K" = k5b)

g5b <- as.data.frame(gmm_bird_2$cluster)
g5b <- g5b[sample(nrow(g5b), 1000),]

tsne_bird <- mutate(tsne_bird, "G" = g5b)

h5b <- as.data.frame(b2$cluster)
h5b <- h5b[sample(nrow(h5b), 1000),]

tsne_bird <- mutate(tsne_bird, "H" = h5b)

ggplot(tsne_bird, aes(x = V1, y = V2, color = as.factor(K)))+ labs(title = "K Means") + geom_point()
```




```
ggplot(tsne_bird, aes(x = V1, y = V2, color = as.factor(G))) + labs(title = "GMM") + geom_point()
```


```
ggplot(tsne_bird, aes(x = V1, y = V2, color = as.factor(H))) + labs(title = "Hierarchical") + geom_poin
```