Cross-Stitch an Image with K-means Clustering

Yingxuan Niu 1003887252

Learning Objective
Use K-means clustering in estimating the best number of clusters and produce colour strips to plot the cross-stitched pattern for an image.

Introduction

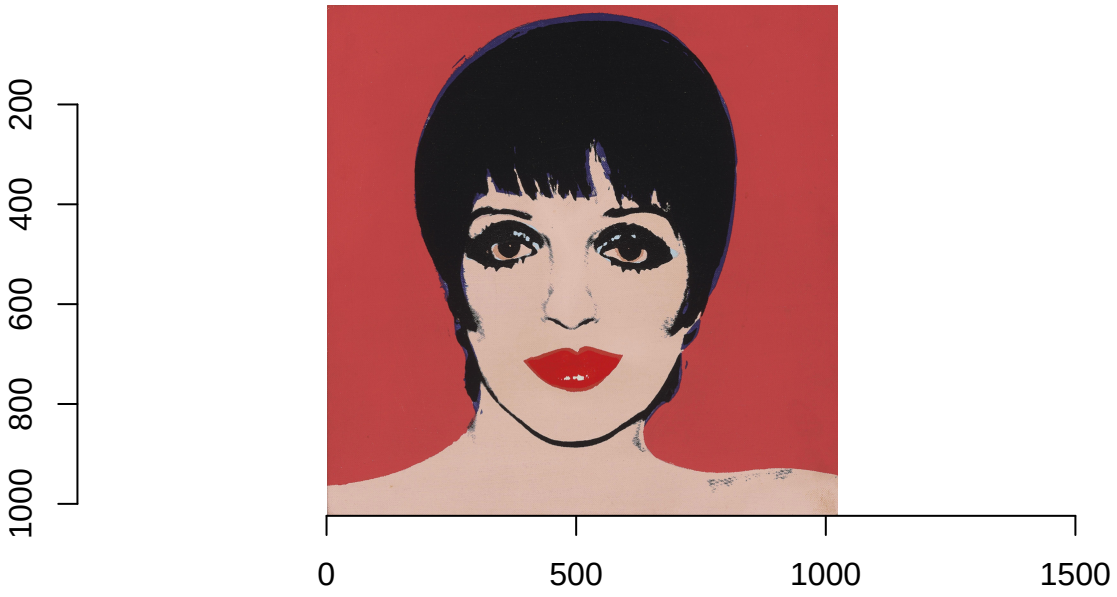
This article requires imager, tidyverse, tidymodels, sp, scales, cowplot and dmc package.
K-means clustering applies tidy data principles and allows users to use glance(), tidy() and augment() functions which are extremely helpful for statistical analysis.

```
source("functions.R")
set.seed(1200)
```

Clustering in R

The process_image function takes in the image file name and a list of number of cluster centres and outputs clustering information. Let's start by converting the desired image into a data frame for clustering. Then we will use the built in kmeans() function to conduct k-means clustering, which takes in the original image data frame. Along we produce glanced and tidied datasets for further use. Furthermore, with the use of for loops, we get the RGB values of cluster centres by using the rgb() function on R, G and B information. The DMC colour information is also acquired with the use of dmc() function on the RGB value. Both are attached to the tidied dataset to form a tibble. Finally, we attach the original image data frame for further use. The output is a list where the first element is a tibble of clustering information containing kclust, glanced and tidied datasets, and the second element is the original image data frame in order for further use. The output is stored in the variable cluster_info.

```
#substitute value and run process_image function
k_list = c(2:10)
image_file_name = "liza.jpg"
cluster_info <- process_image(image_file_name, k_list)
```



Plotting Scree Plot

We then produce the scree plot with the input as cluster_info outputted from the last function, where we utilize the glanced dataset and plot scree plots with k on the horizontal axis and ratio on the vertical axis. From the output ratio scree plot, it looks like k = 7 would be the best number of clusters that we can work with, since it explains around 90% of the data, and the ratio barely increases anymore after k = 7.

```
scree_plot(cluster_info[[1]])
```

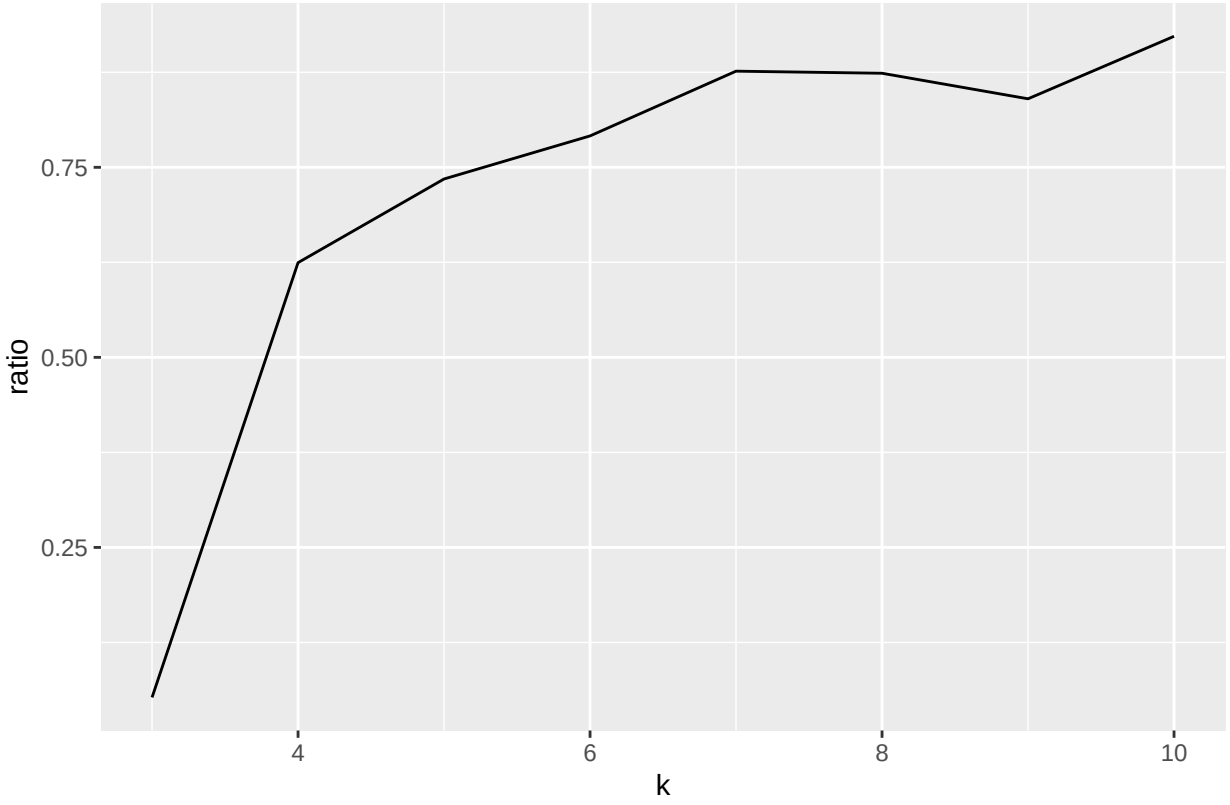
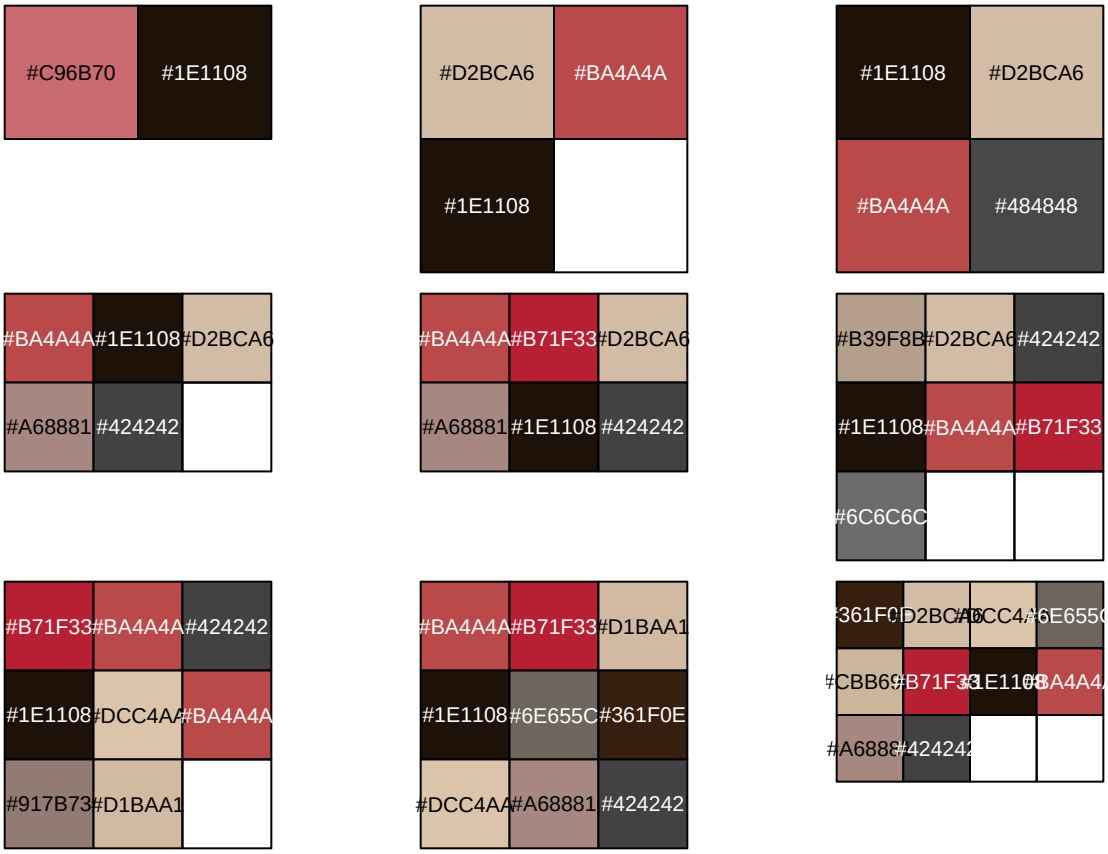


Figure 1. Scree plot

Plotting Colour Strips

We then plot the closest DMC colour to the cluster centre colour for every number of cluster centres in the k_list, so that we can see what each colour strip looks like for different k's.
In this case, when k>7, there appears to be a some repetitive colours; whereas when k<7, the colour strip is not representative enough of the colours appeared in the image. Thus we choose k=7 to be the number of clusters.

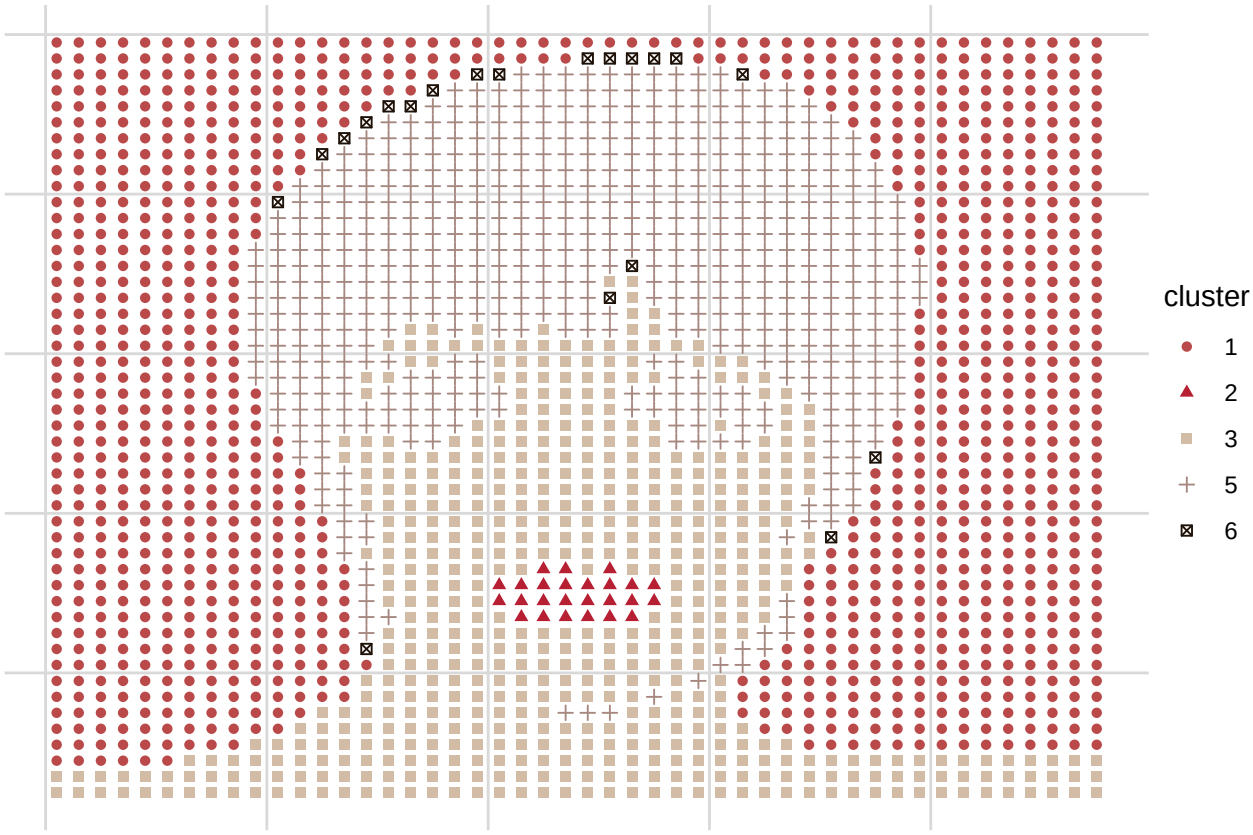
```
colour_strips(cluster_info[[1]])
```



Making Cross-Stitch Pattern

To make a cross-stitch pattern by inputting cluster_info, the number of clusters we chose, x_size and colours, first we want to augment the original data frame with the cluster centre and RGB information using the augment() function. In this way, each data point is assigned to a cluster. Then we produce a lower resolution image by summarizing the non-coordinate columns in the data frame with their most common value in the large grid cell, therefore outputs a data frame with the same column names as the original frame, but with fewer entries that corresponds to the reduced resolution image.
After reducing the resolution, we plot the cross stitch pattern with ggplot. By setting colour = cluster and shape = cluster, ggplot differentiates each pixel by its cluster ID and thus each pixel has the colour of its cluster centre.

```
make_pattern(cluster_info, k=6, x_size= 50, black_white=FALSE, background_colour=NULL)
```



Cross Stitch Image