Vignette of A1: Cross stitch

Hao Shi

2020-11-03

Introduction

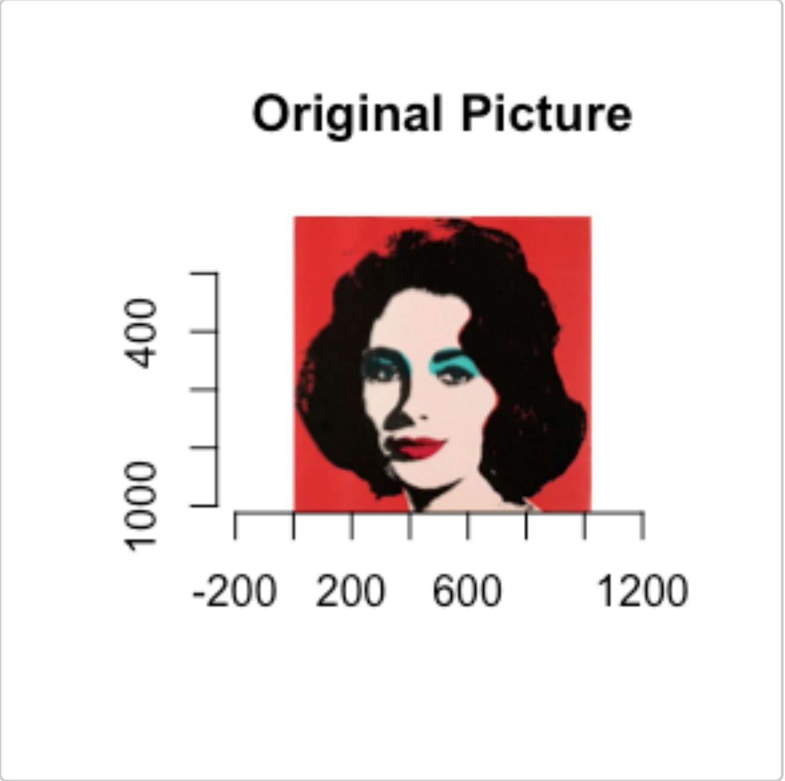
This vignette covers the description for the process of cross-stitch as well as briefings of the four functions which are required for the Assignment1: process_image(), scree_plot(), color_strips() and make_pattern().

The change_resolution() function used in the make_pattern() function is pre-written and already documented by the professor.

```
# load the libraries needed to run the assignment
library(imager)
library(tidyverse)
library(tidymodels)
library(sp)
library(scales)
library(cowplot)
#devtools::install_github("sharlagelfand/dmc", force=TRUE)
library(dmc)
```

Process_image( )

```
# load functions written in functions.R
source("functions.R")
image_file = "/Users/david/Desktop/STA314/STA314 A1/test.jpg"
k_list = c(2,3,4,5,6,7)
cluster_info = process_image(image_file, k_list)
```

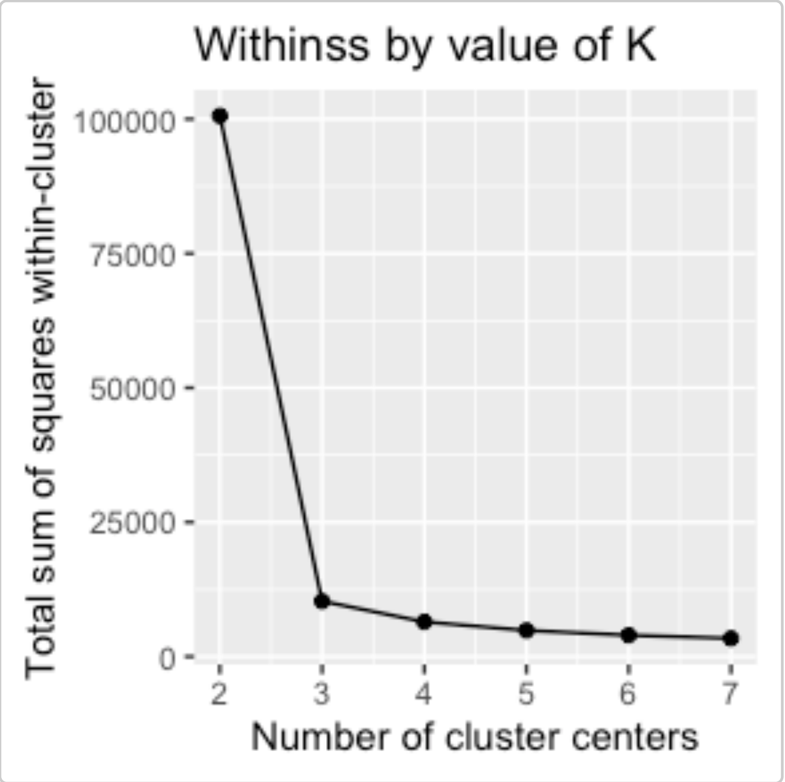


To start off, we setup the parameters which will be used as inputs for the process_image() function. we rename our picture as “test.jpg” and save it as image_file. For the other parameter k_list, which is the vector containing numeric values which represents the number of clusters we will choose from.

Since my observation of the original picture tell me the number of color in this picture is no more than 7, we use the vector of (2,3,4,5,6,7) to test on multiple values of number of clusters in order to choose our optimal number of clusters by drawing the screeplot using the scree_plot() function.

Scree_plot( )

```
scree_plot(cluster_info[[2]])
```

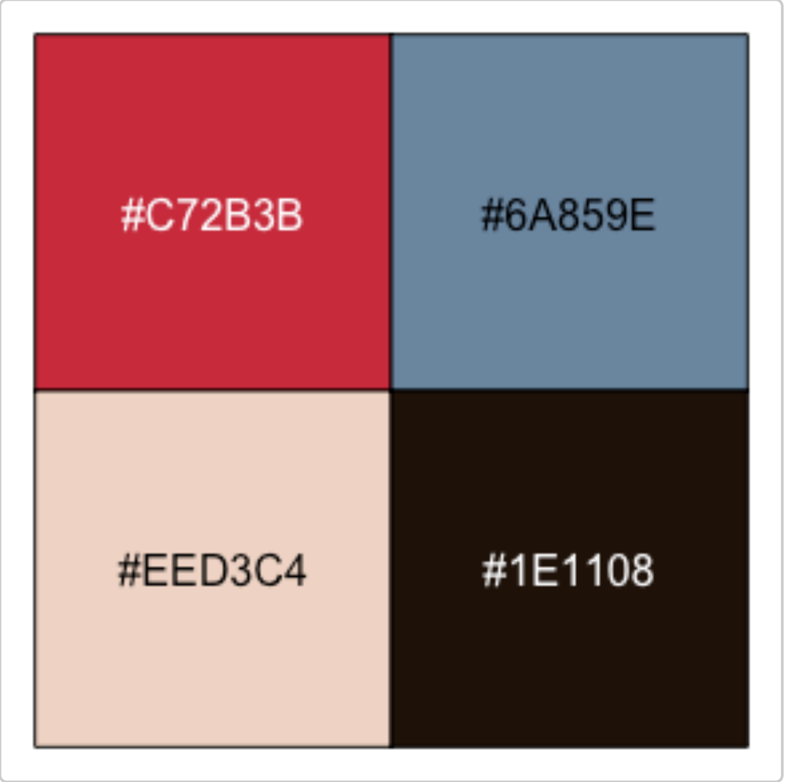


Produced by the scree_plot() function, the Withinss by value of K plot illustrates the change in total sum of squares within the clusters as the number of cluster centers increases. The variance decreases as the number of clusters increases, also notice a “elbow” around k = 3.

We can also find additional information from the Proportion by num of K plot (not shown in the knitted pdf) which has a bend at k=4, the proportion of variance explained goes up by a bigger amount with the increment of 1 in the number of clusters compared to other values with the increment of 1 in the number of clusters. Hence, we chose the number of k to be 4. In other words, the color_strips() function will give four DMC thread colors.

Color_strips( )

```
k=4
a_cluster_info = color_strips(cluster_info[[2]][k-1,])
```



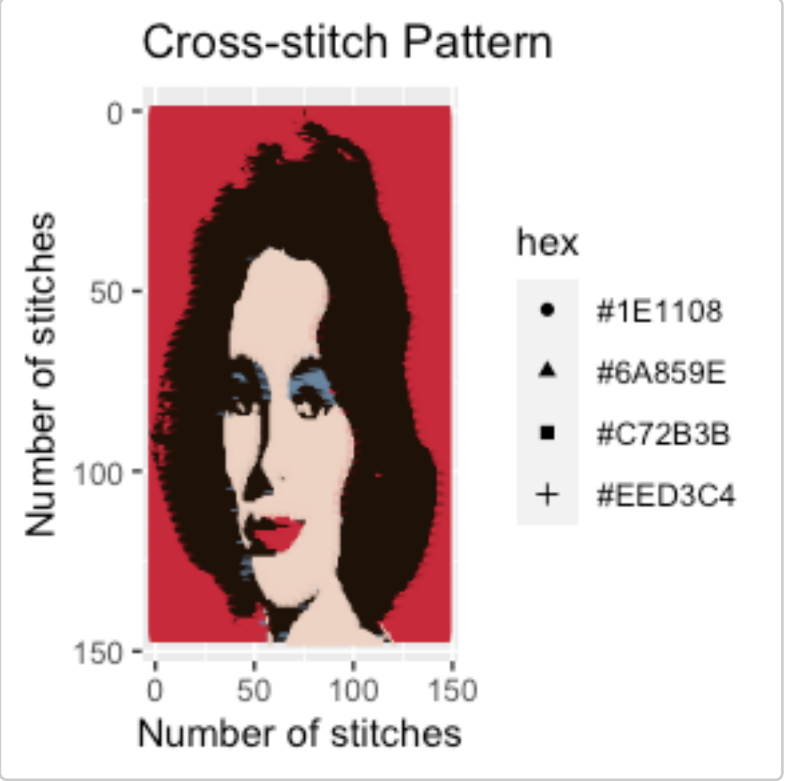
The color_strips() function gives 4 colors since we chose k=4:

1. #1E1108 This color has RGB value of (30, 17, 8), which is this black-ish orange that colors the woman’s hair, eyebrow and shadow of nose.
2. #6A859E This color has RGB value of (106, 133, 158), which is this dark blue that colors the woman’s eye shadow.
3. #C72B3B This color has RGB value of (78, 17, 23), which is this strong red that colors the background, the woman’s lips.
4. #EED3C4 This color has RGB value of (93, 83, 77), which is this Light grayish orange that colors the woman’s skin.

These hex codes of the colors given by the kclust results will be used to help make our cross-stitch pattern in the make_pattern() function.

Make_pattern( )

```
make_pattern(a_cluster_info, k = 4, x_size = 150, black_white = FALSE, background_colour = NULL)
```



Lastly, the make_pattern() function produces a cross-stitch pattern plot which resembles our original picture by using the four colors given by the color_strips() function. The parameter x_size controls how many stitches we want on each axis.

Personally I prefer the number of stitches on each axis to be 150 as it gives a detailed cross-stitch pattern while taking a moderate amount of time to compile. In addition, we set black_white = FALSE to avoid producing black-white plot and we set background_color = NULL since we want only the colors from the colors_strips() functions to appear in our final cross-stitch plot.