

STA314

Zhixing Hong

11/1/2020

Firstly, we are going to load all the packages that needed for the following functions.

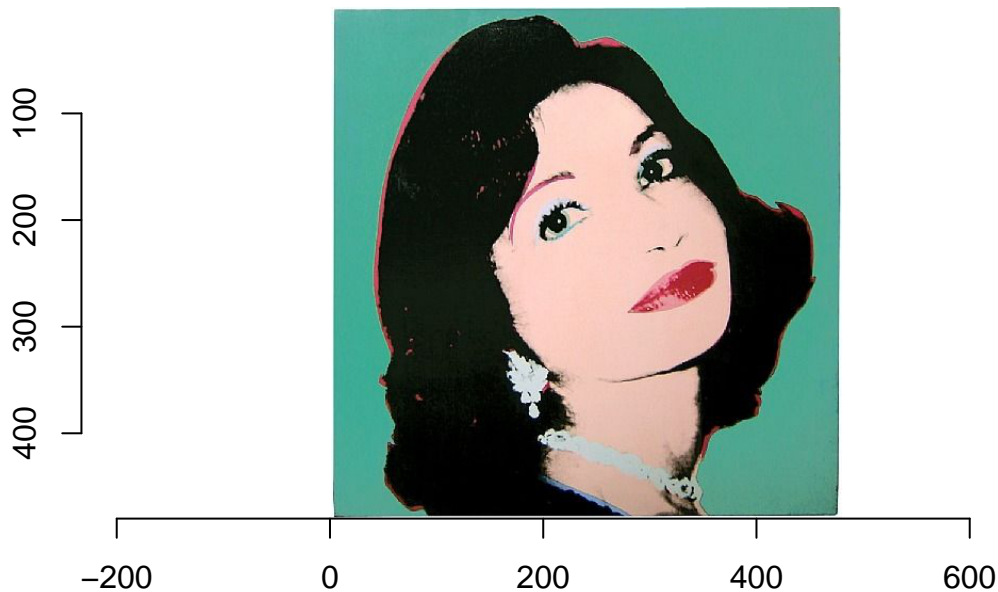
```
library(imager)
library(tidyverse)
library(tidymodels)
library(sp)
library(scales)
library(broom)
library(cowplot)
# devtools::install_github("sharlagelfand/dmc")
library(dmc)
```

Then we need to load all the functions written in `functions.R` into this file, to preform a example of the making cross-stitch with certain picture. We used `source()` here to load the four functions, `process_image()`, `scree_plot()`, `color_strips()` and `make_pattern()`.

The picture used here also made by Andy Warhol. The basic This picture have a relatively less color and the background of the image is also not very noisy. Therefore, this picture satisfy the basic requirement.

```
source("functions.R")

im <- load_image("similar_pic.jpg")
plot(im)
```



Here is an example of using `process_image()` to get the clustering information for the image showed above.

The clustering info now is stored in `info`.

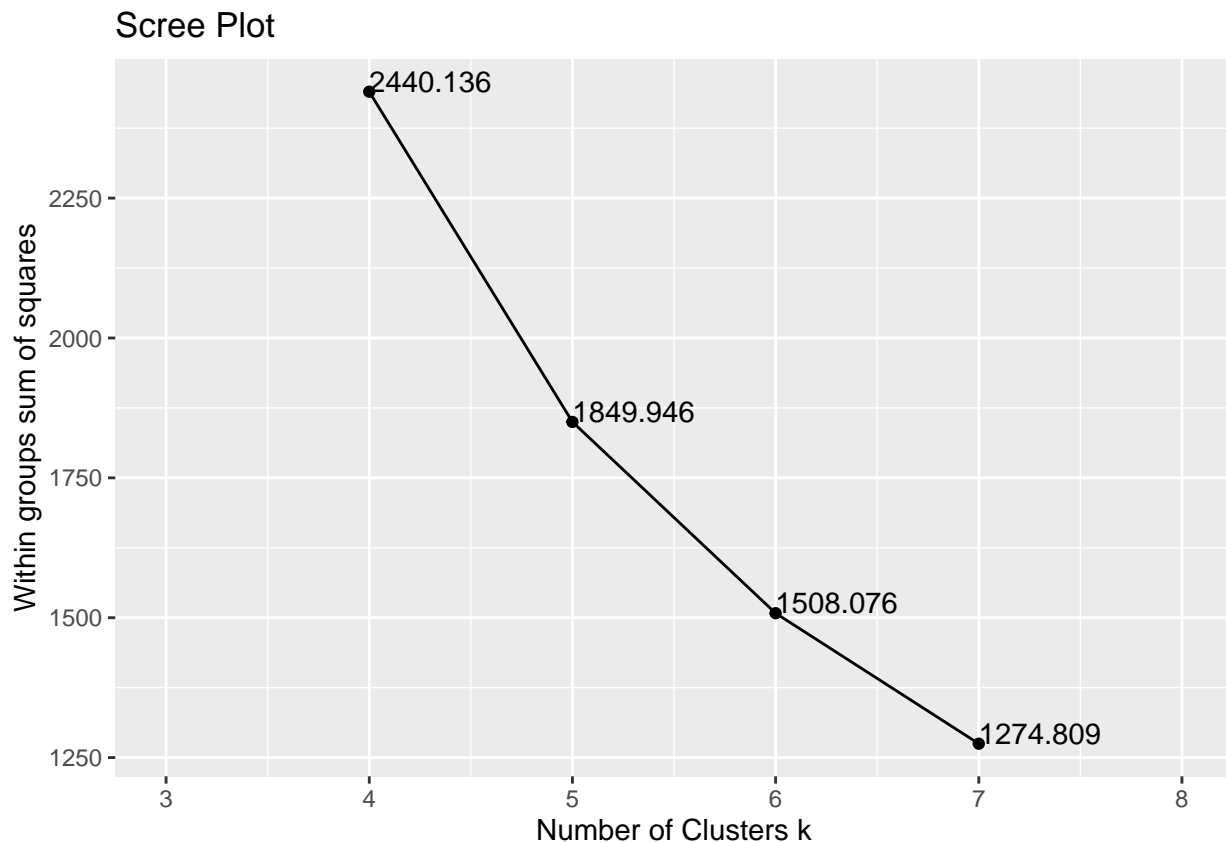
```
info <- process_image("similar_pic.jpg", 4:7)
info
```

```
## # A tibble: 4 x 8
##       k kclust  totss tot.withinss betweenss  iter cluster_center tidy_dat
##   <int> <list>   <dbl>      <dbl>      <dbl> <int> <list>          <list>
## 1     4 <kmean~ 74367.      2440.      71927.     3 <tibble [4 x 8~ <tibble [23~
## 2     5 <kmean~ 74367.      1850.      72517.     4 <tibble [5 x 8~ <tibble [23~
## 3     6 <kmean~ 74367.      1508.      72859.     3 <tibble [6 x 8~ <tibble [23~
## 4     7 <kmean~ 74367.      1275.      73092.     3 <tibble [7 x 8~ <tibble [23~
```

Scree Plot

After having all those numerical values are still not enough for us to decided how many clusters needed to be included. Therefore, scree plot is plotted below.

```
scree_plot(info)
```



The y-axis on the scree plot is the sum of squares of variance; while on the x-axis, it is the number of clusters choose. To enable users have quicker way to decided what exact number of clusters should be used, the value of sum of squares are also plotted on the graph.

From this scree plot, we noticed that when we change from 5 clusters to 6 clusters, the decline in the sum of squares is relatively large. Therefore, we might want use 6 clusters to fill the cross-stitch.

Color Strips

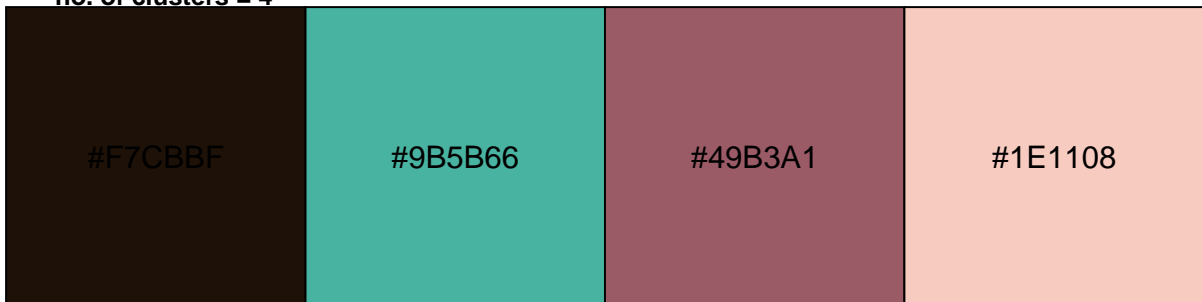
To make the decision more accurate, we can also view the color of each cluster. Since the color strips could also give some information whether we need to include more clusters or we can reduce some.

Here is the example for using `color_strips`. The input for this function is still the output from `process_image`.

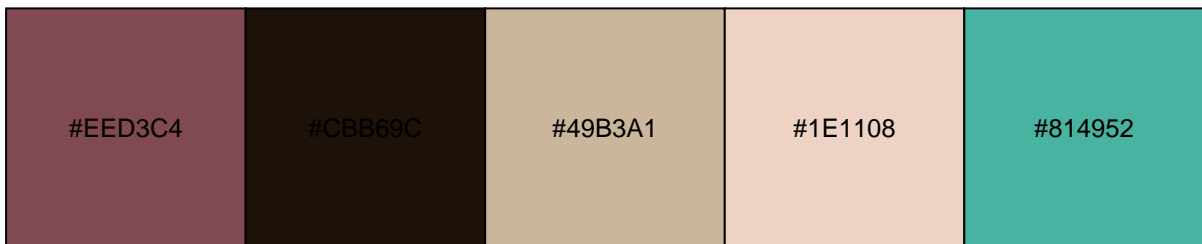
```
color_strips(info)
```

Colour of Each Cluster Centre In DMC

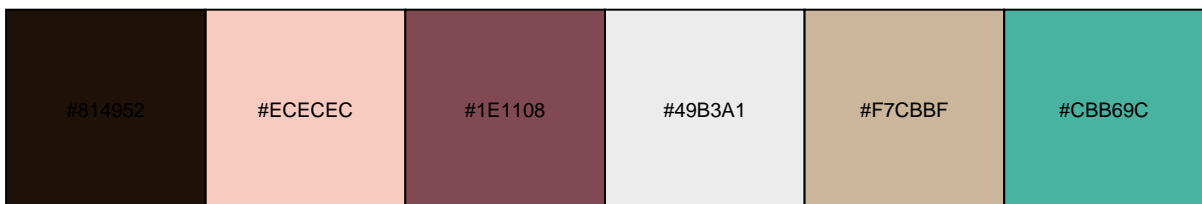
no. of clusters = 4



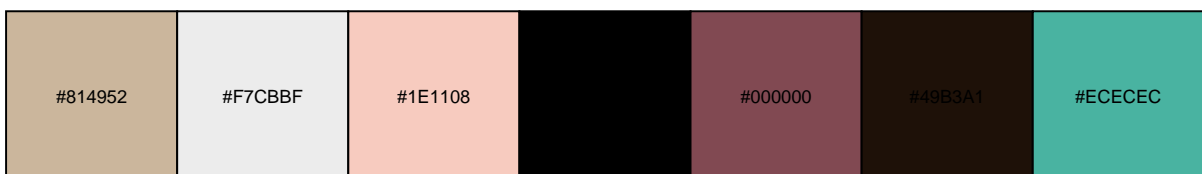
no. of clusters = 5



no. of clusters = 6



no. of clusters = 7



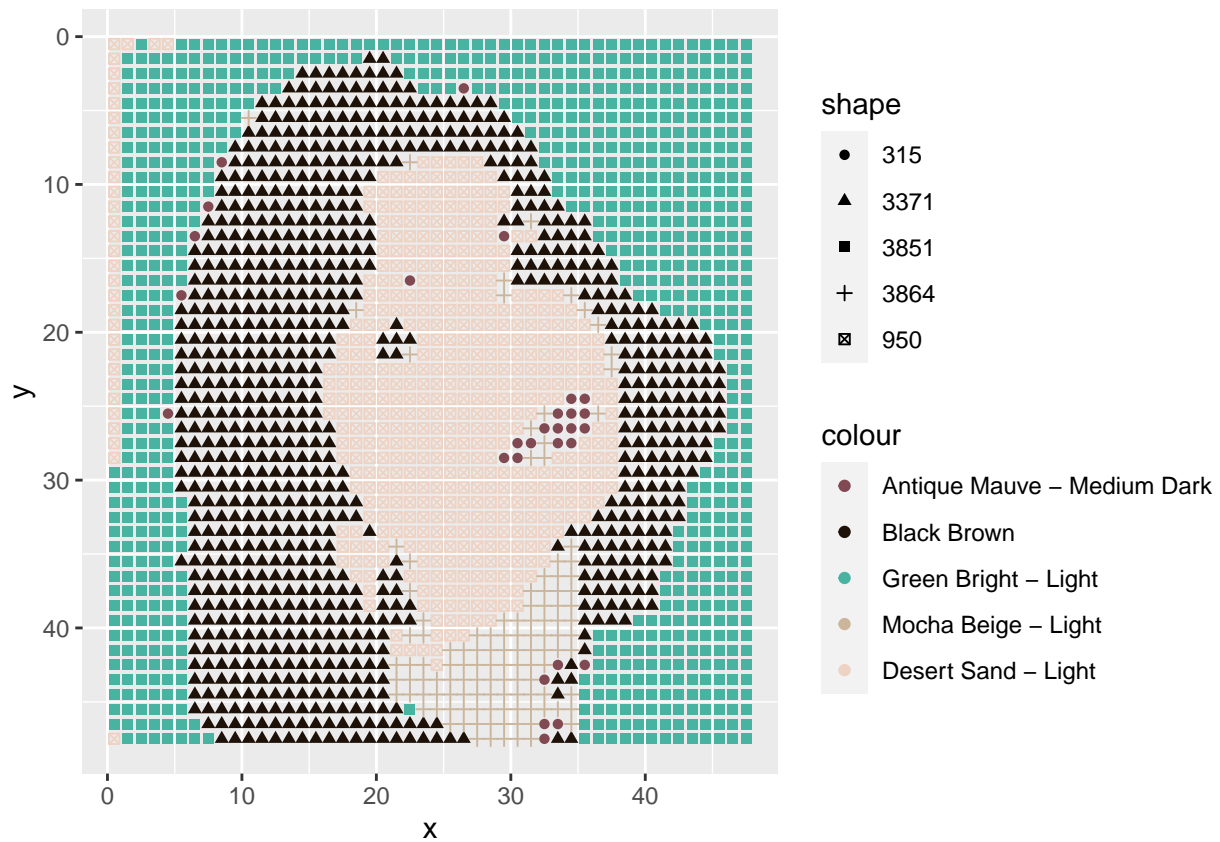
The color strips are plotted above. For each strip, the label are on the left upper corner. The color name(hex) is also labeled on each color box.

Make Cross Stitch Pattern

Having all the information above, we can finally produce the cross-stitch by `make_pattern`.

The basic usage is just put the *cluster information* getting from `process_image`, the number of clusters the user chose, and the desired size for the cross-stitch. Here, we used `info` and set the number of cluster equals to 5 and the size of cross-stitch equals to 50.

```
make_pattern(info, 5, 50)
```

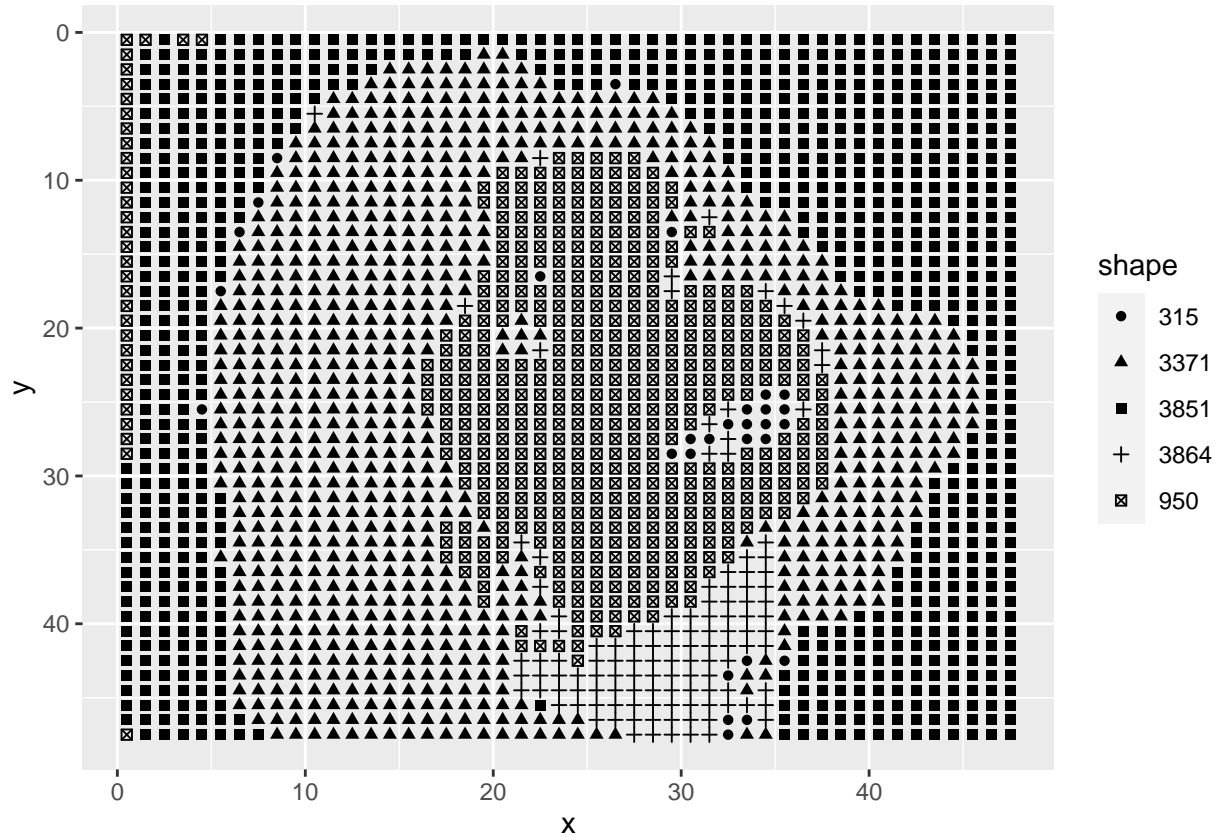


More options with Cross Stitch Pattern

What is also worth to mentioning is that, `make_pattern` have some other options could be use.

The first option is the color option(`black_white`). This option gives a choice for users whether they want to have a colored cross stitch or just black and white. The default value for this is `FALSE`, which means a colorful pattern will be plotted. The other option is to set `black_white = TRUE`. Here is an example for that.

```
make_pattern(info, 5, 50, black_white = TRUE)
```



The second option is setting the `background_color`. After setting input value of background color to be some hex value, the new cross-stitch is shown below. Compared to the cross-stitch, we noticed that the background points are now removed, leaving room for the background color. This also informs the user that they don't need to make those black space.

```
make_pattern(info, 5, 50, background_colour = "#49B3A1")
```

