

Making a cross-stitch pattern

Edmund Lo

November 3, 2020

This vignette covers how to make a cross-stitch pattern from an image using the functions in functions.R. The following functions: `process_image()`, `screed_plot`, `colour_strips`, and `make_pattern` will be used.

Loading an Image

The `load.image()` function allows you to load in an image of your choice, by inputting the file path of the image you have saved.

```
image <- load.image("image.png")
```

Clustering and Finding DMC Colours

The `process_image()` function is used by inputting the image and a `k_list`. `k_list` is a list of integers representing the number of clusters you may want. In the following example, 2, 4, 6, and 8 clusters are inputted. The function outputs a tibble of tibbles, where each row corresponds to the number of clusters given in `k_list` and each column gives a specific type of information. The first column is the number of clusters given in `k_list`. The second column gives the image dataframe containing x and y's and their respective R,G,B values. The third column is the output of the kmeans clustering. The last column gives information on the DMC colours, including the hex value, colour name and DMC colour number. For future use, we save the output of `process_image()` in `cluster_info`.

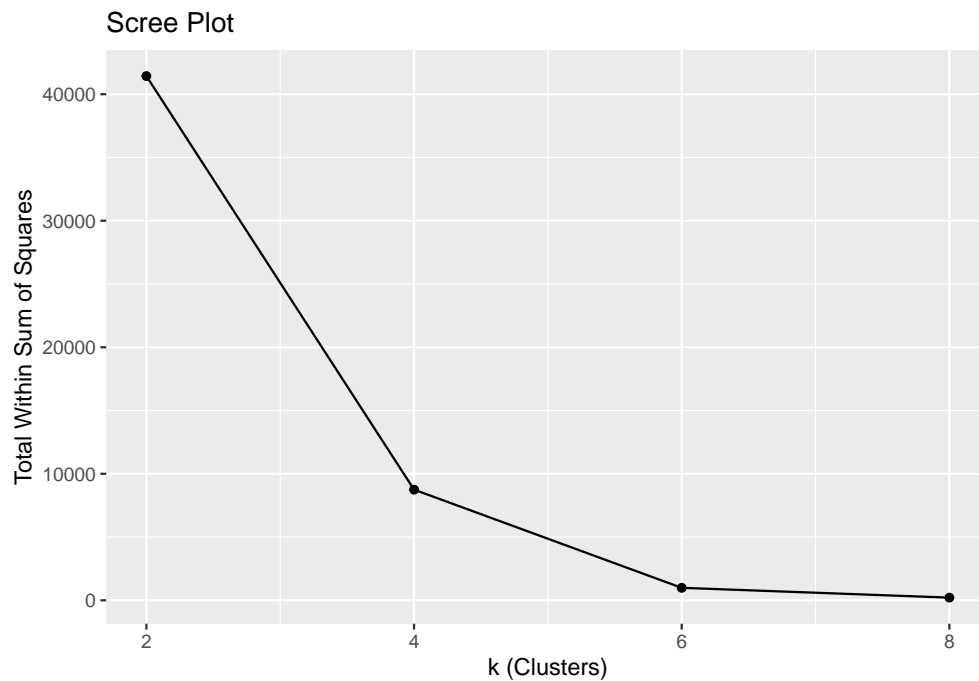
```
cluster_info <- process_image(image, k_list = c(2,4,6,8))
cluster_info
```

```
## # A tibble: 4 x 4
##   clusters im_dat          kclust   info
##   <dbl> <list>          <list> <list>
## 1      2 <df[,5] [262,144 x 5]> <kmeans> <df[,10] [2 x 10]>
## 2      4 <df[,5] [262,144 x 5]> <kmeans> <df[,10] [4 x 10]>
## 3      6 <df[,5] [262,144 x 5]> <kmeans> <df[,10] [6 x 10]>
## 4      8 <df[,5] [262,144 x 5]> <kmeans> <df[,10] [8 x 10]>
```

Scree Plot

The `scree_plot()` function is used by inputting `cluster_info` from above. Then this function produces and plots a scree plot. The plot shows the number of clusters vs the Total Within Sum of Squares. The scree plot can be used to determine the optimal number of clusters to use. To find the optimal cluster, look for the point at which there is a bend or elbow in the line. The bend indicates that any additional clusters provide little extra value. In the scree plot below, there is a clear bend or elbow at $k = 6$ or 6 clusters. We decide that 6 clusters is optimal for this image.

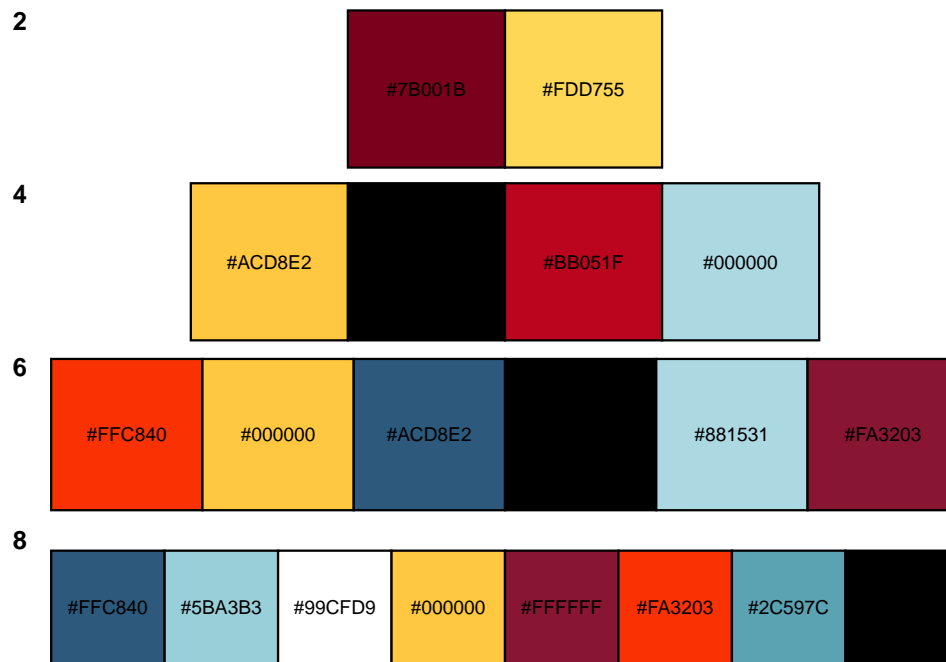
```
scree_plot(cluster_info)
```



Colour Strips

The `colour_strips` function is used by inputting `cluster_info` from above. This function produces colour strips for each `k` number of clusters. Each box of colour represents the colour of one of the clusters. For example, a cross-stitch with 6 clusters would have the 6 colours in the third colour strip. The colour strips can be used to verify our choice of the optimal number of clusters. For the 6 cluster colour strip, we can clearly see 6 distinct colours. If we choose 8 clusters, visually, we can see that there are two very similar blues. Therefore, we can determine that 6 clusters is best for our chosen image.

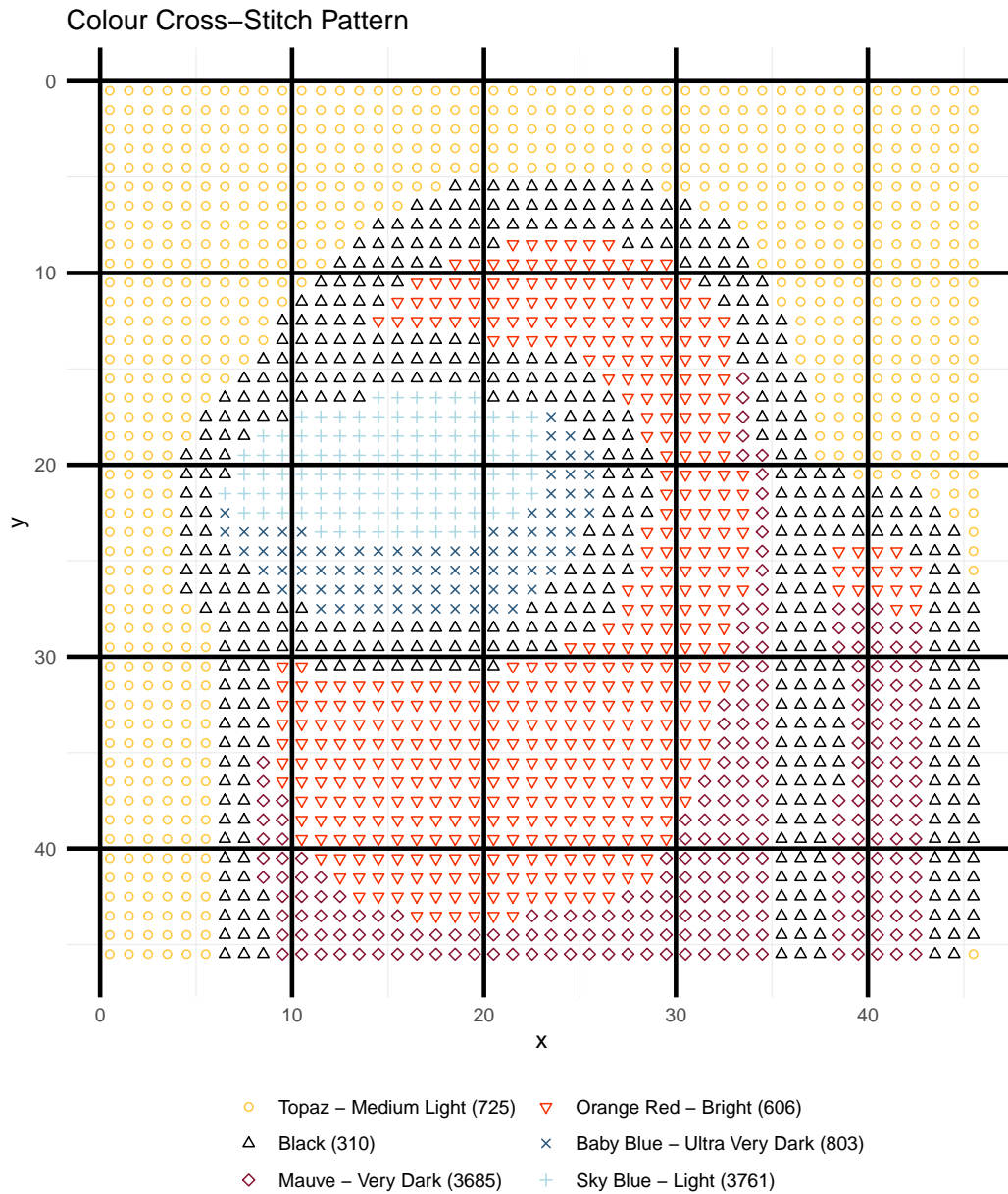
```
colour_strips(cluster_info)
```



Cross-Stitch Pattern

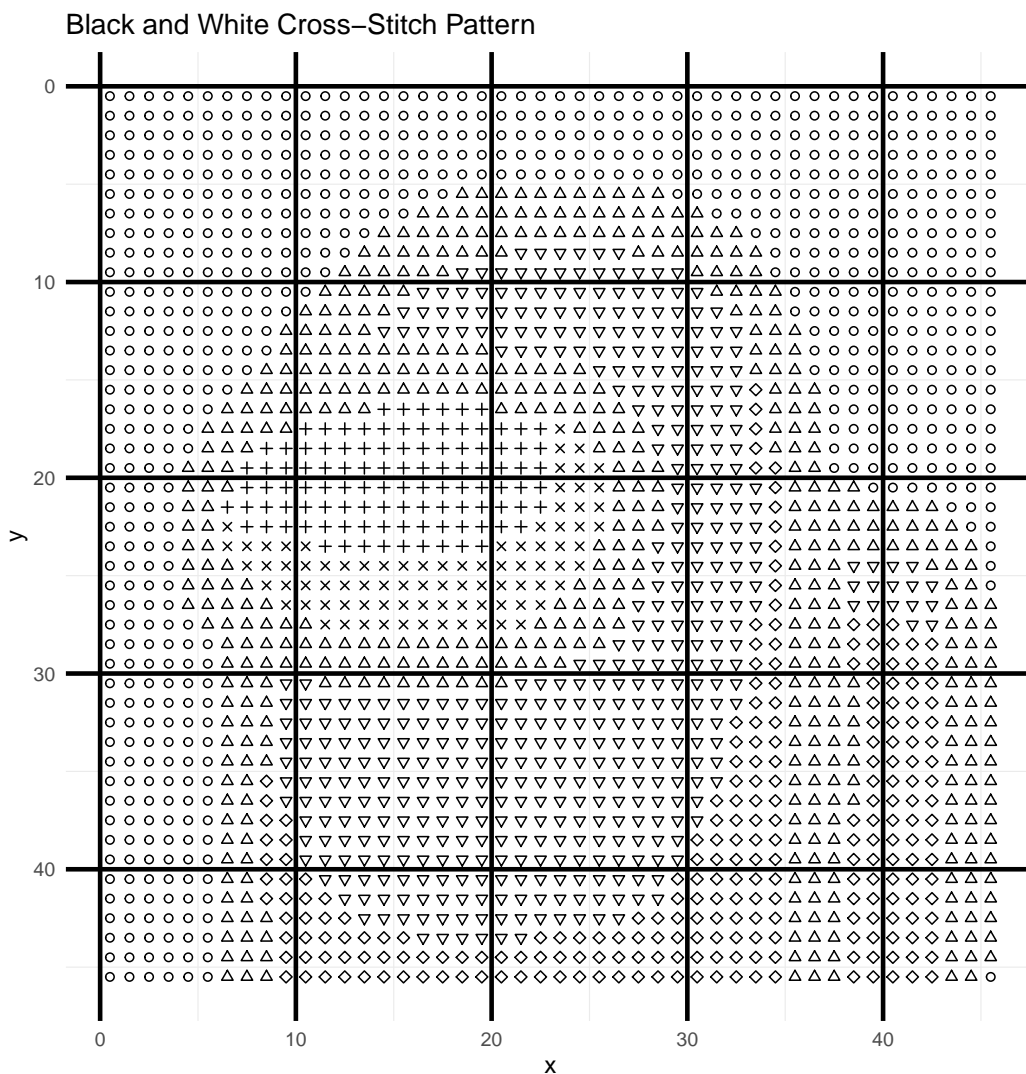
The `make_pattern()` function is used by inputting the `cluster_info` from above, the chosen number of clusters (`k`), the approximate total number of possible stitches in the horizontal direction (`x_size`), true or false for a black and white pattern, and the colour of the background, which should not be stitched in the pattern. By default, `black_white` is set to `FALSE` and `background_colour` is null, so the function produces a coloured cross-stitch pattern of the image with a different colour and shape for each of the clusters. The legend at the bottom gives the shape and DMC colour of each cluster. The cross-stitch below is a coloured pattern made using 6 clusters, as chosen before and with `x_size` equal to 50.

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



To get a black and white cross-stitch, set `black_white` equal to `TRUE`.

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



- | | |
|------------------------------|-------------------------------------|
| ○ Topaz – Medium Light (725) | ▽ Orange Red – Bright (606) |
| △ Black (310) | × Baby Blue – Ultra Very Dark (803) |
| ◇ Mauve – Very Dark (3685) | + Sky Blue – Light (3761) |

To get a cross-stitch without the background colour, we can set the value of `background_colour` to the hex value of the background colour. Looking at the coloured cross-stitch and its legend, we see that the background colour is Topaz - Medium Light (725). Then, we can check the 6 cluster colour strip from before to find the matching colour and its hex code. We see that the hex value of Topaz - Medium Light (725) is #FFC840 and we input this as the background colour in the `make_pattern()` function.

```
make_pattern(cluster_info, k = 6, x_size = 50, black_white = FALSE, background_colour = "#FFC840")
```

