Creating a cross stitch

Haoying Shen

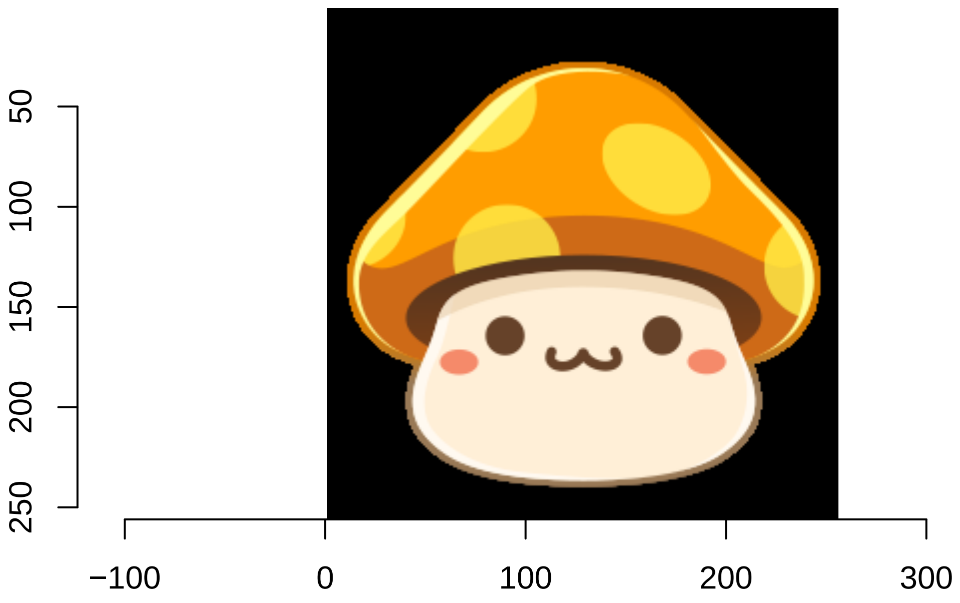
31/10/2020

This vignette will cover the functions used in creating a cross stitch template from a image.

##Creating Cluster_Info

The first function that will be covered is process_image, but before that we must save a image to a variable

```r
im <- imager::load.image("C:/Users/haoyi/Desktop/MS.png")
plot(im)
```



So this is the image that we want to turn into a cross stitch template.

```r
set.seed(7)
k_list = c(3, 5, 7, 9, 20)
cluster_info <- process_image(im, k_list)
```

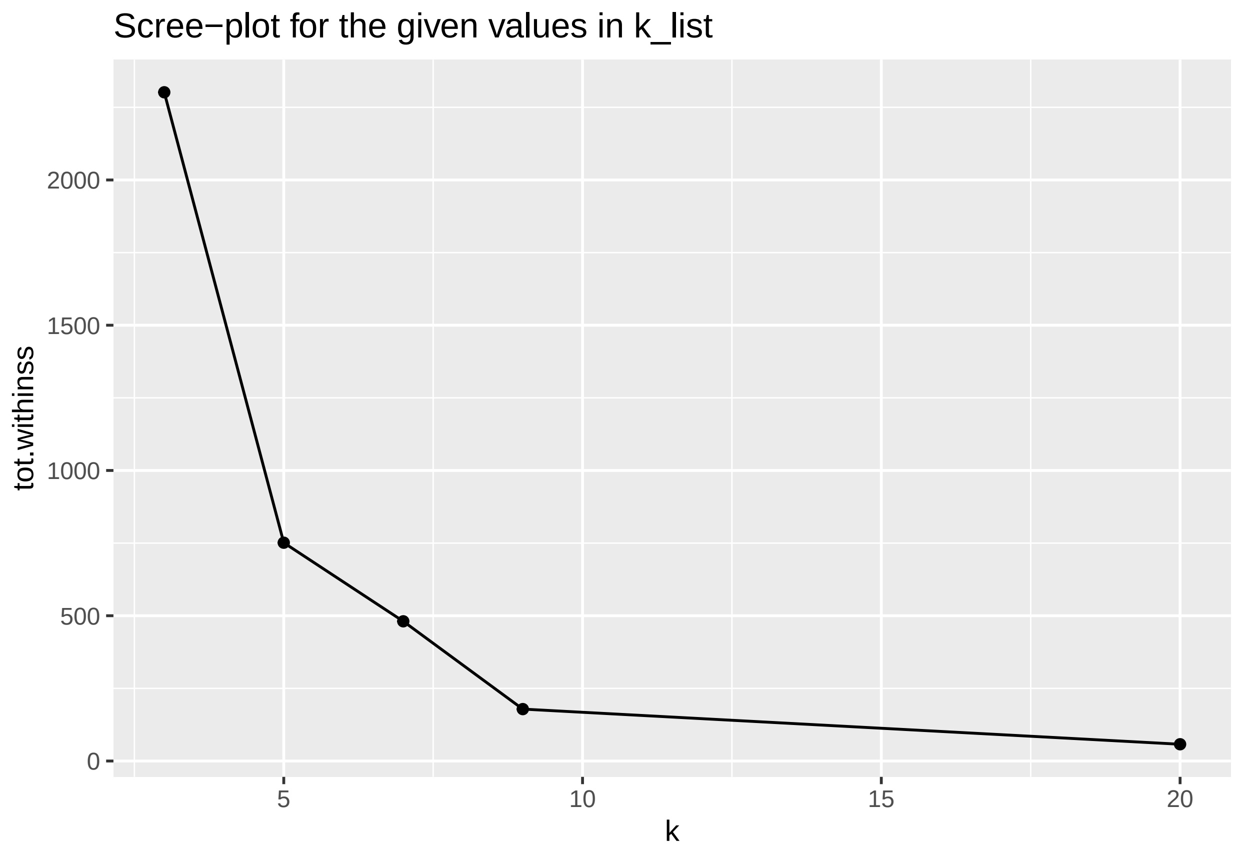
```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

process_image generates many different dataframes which include information such as the k-means clustering information for the image from each of the values in k_list, as well as the RGB values which will be used to create the overall color of the image. Additionally this also includes the respective DMC name, which is the matching tread color for the stitch.

##Creating a Scree_plot

since eventually we want to figureout the best clustering value, we should use a scree_plot, this plots the total within sum of squares against each of the k values in k_list.

```
scree_plot(cluster_info)
```

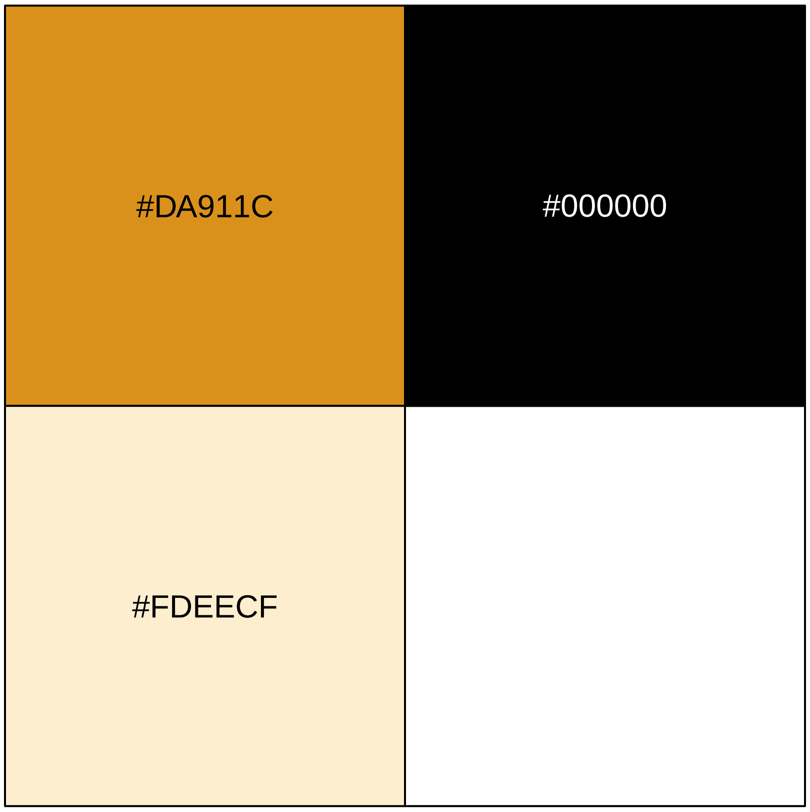


This is used to find the number of clusters that can explain most of the variance, or in our case create the most detailed picture. Screeplots almost exclusively decrease with more clusters. However this may be more harmful than beneficial, so we should go with the number that explains a good portion and does not rely on too many clusters. in this case the we see very little difference between 9 and 20 clusters, which shows that there is diminishing value. So 9 should be sufficient.

##Creating colorstrips

A good way to visualize what colours would be used for your given choice of k is to use the color_strips function. This will show you all of the colors that can be used to create the cross stitch for each k in k_list.

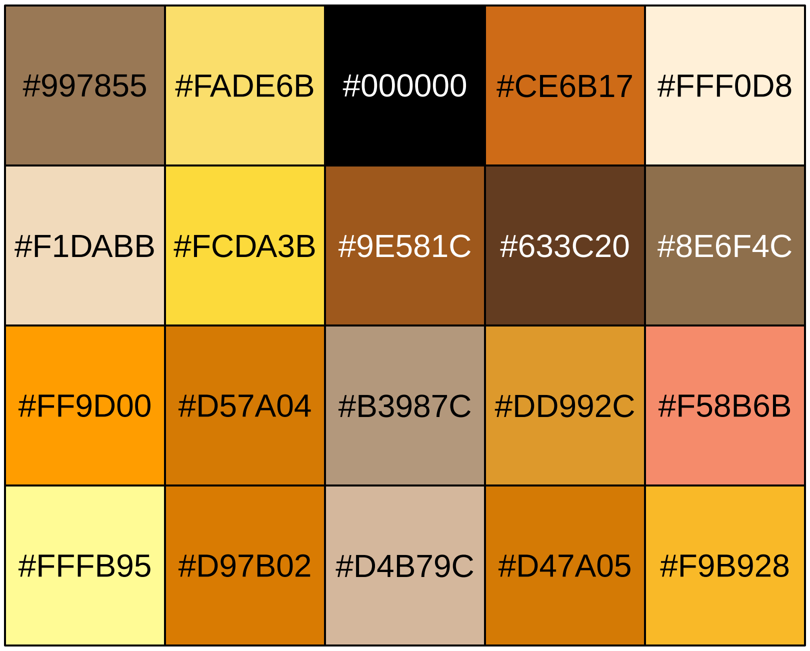
```
colour_strips(cluster_info)
```



#714B2D	#EB8A08	#000000
#FDEED0	#F9D243	

#B98461	#FCD93D	#EC8A08
#FEEED0	#000000	#653D21
#BC761D		

#000000	#B88461	#FCD83C
#BC761D	#D06E13	#FFEFD8
#FF9D01	#F7E9A5	#643C21

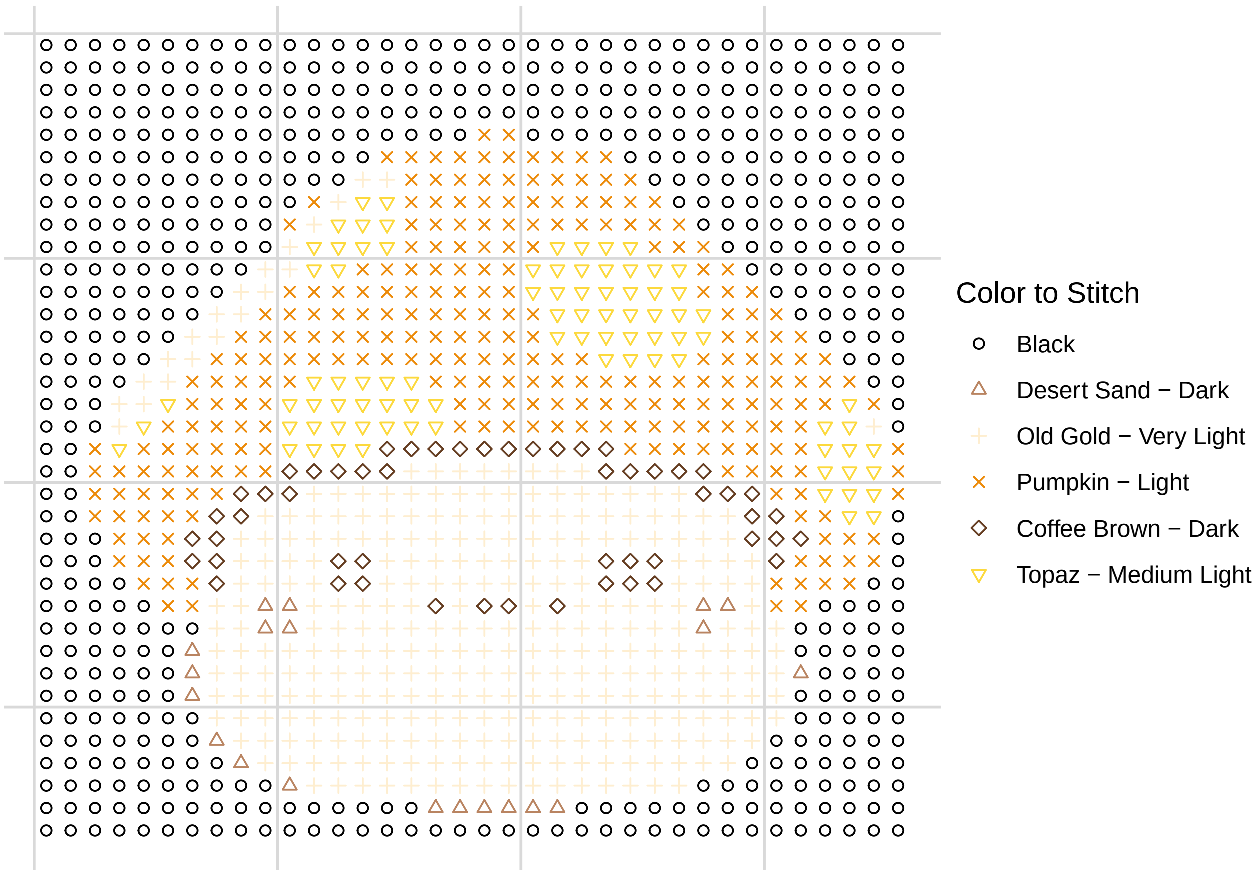


This further backs up the diminishing value of having more clusters, as we can see in the 20 cluster colorstrip there are many colors that are hard to distinguish from eachother. so choosing a lower number would get rid of that inefficiency by grouping those colors together.

##Creating the cross stitch

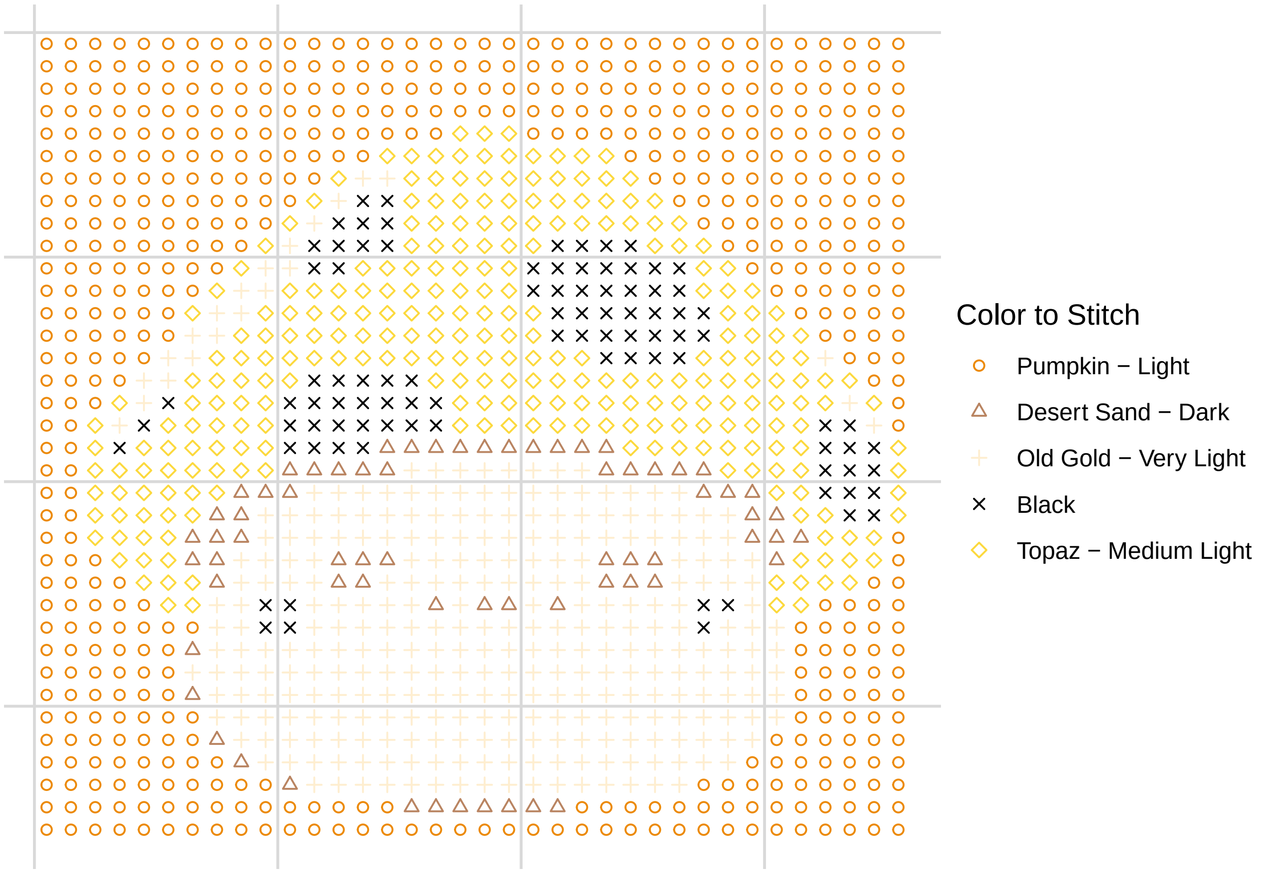
Finally we get to the cross stitching part, here we have many different options. the first input is cluster_info, the second is your chosen k-value, keep in mind this must be the same as a value within k_list. For the first one I will choose k = 7

```r
make_pattern(cluster_info, 7, 40, FALSE)
```



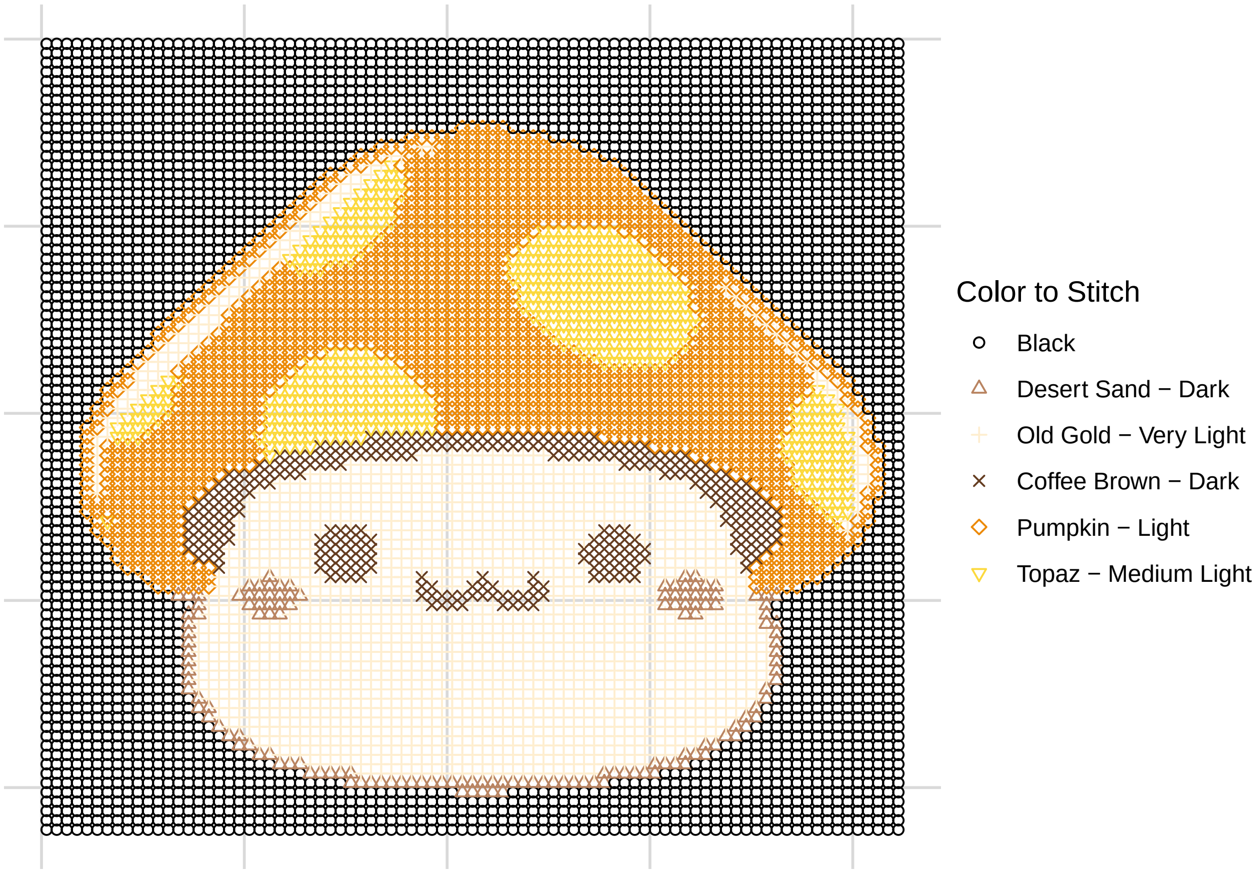
This next one uses 5 clusters instead, as you can see there are less colour choices

```r
make_pattern(cluster_info, 5, 40, FALSE)
```



we can also change the resolution of the final picture by changing x_value, the previous were done using 40, lets try with 100

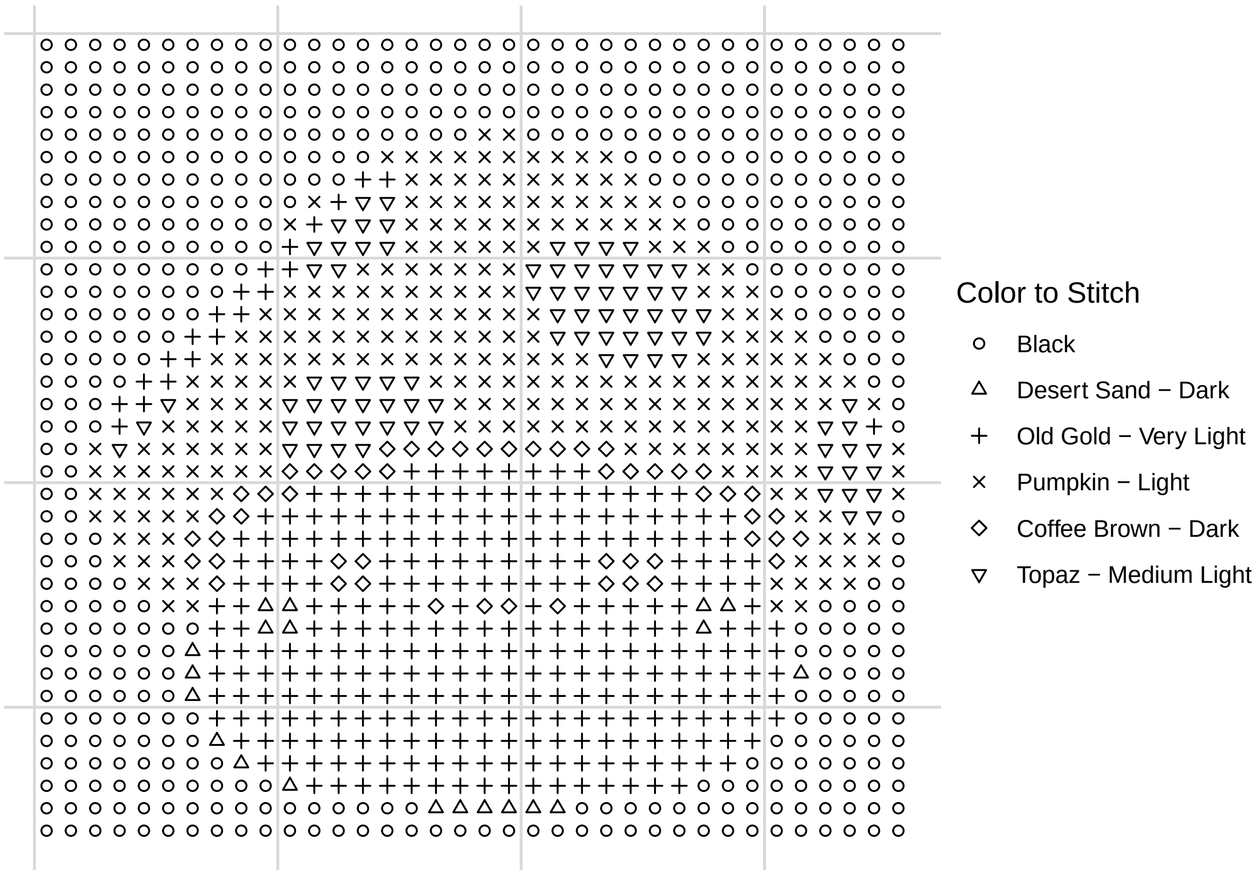
```r
make_pattern(cluster_info, 7, 100, FALSE)
```



In this case the picture is far more detailed.

Lastly, the picture can be cross stitched in black and white when the input for black_white is turned to TRUE, and the symbol will be the only differentiation between the stitches.

```
make_pattern(cluster_info, 7, 40, TRUE)
```


Color to Stitch
○ Black
△ Desert Sand − Dark
+ Old Gold − Very Light
× Pumpkin − Light
◇ Coffee Brown − Dark
▽ Topaz − Medium Light