314 A1-vignette

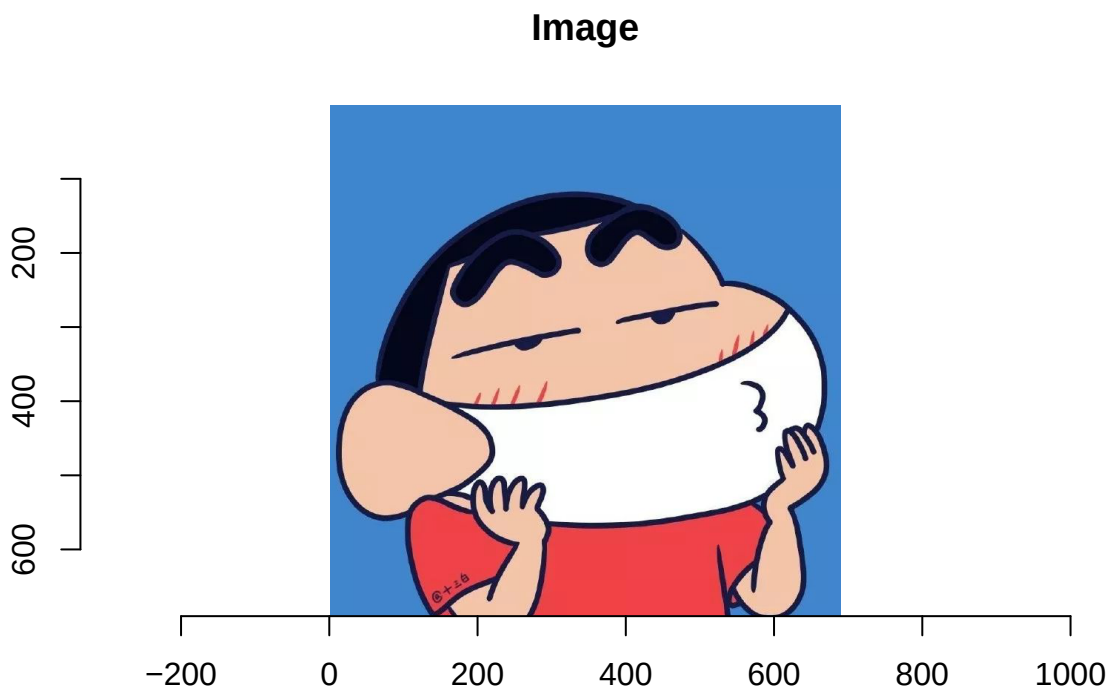
```
library(imager)
library(tidyverse)
library(tidymodels)
library(dplyr)
library(sp)
library(scales)
library(cowplot)
library(ggplot2)
#install.packages("devtools")
#library(devtools)
#devtools::install_github("sharlagelfand/dmc")
library(dmc)

source('functions.R')
```

The tasks for this project is to write several R functions that will take an image and create a cross-stitch pattern. I write the function in 'function.R', and I will test the function in following parts.

First, I load the image by function imager::load.image(). The following is the picture I use.

```
set.seed(8477)
image_file_name<-'Crayon.jpg'
im<-imager::load.image(image_file_name)
plot(im,main="Image")
```



The second step is to gain the tibble of information derived from the k_means. We use the function process_image() and store the result in 'gain_clusters'. Each row provide information for k-means clustering of each k.

k: k value from 2-8

totss: The total sum of squares.

betweenss: The between-cluster sum of squares, i.e. $totss - tot.withinss$.

tot.withinss: Total within-cluster sum of squares, i.e. sum(withinss).

totss, tot.withinss, betweenss, and iter (1 value) contain information about the full clustering

centers(nested column): A matrix of cluster centers. tidy_dat: contains coordinate,RGB information abour original data points. centers and tidy_dat contain information about each cluster

```
gain_clusters<-process_image(image_file_name,k=2:8)
gain_clusters
```

```
## # A tibble: 7 x 8
##       k kclust    totss tot.withinss betweenss  iter centres     tidy_dat
##   <int> <list>    <dbl>        <dbl>     <dbl> <int> <list>      <list>
## 1     2 <kmean~ 139964.       59752.    80212.     1 <tibble [2~ <tibble [476,1~
## 2     3 <kmean~ 139964.       29067.   110897.     3 <tibble [3~ <tibble [476,1~
## 3     4 <kmean~ 139964.        9042.   130922.     3 <tibble [4~ <tibble [476,1~
## 4     5 <kmean~ 139964.        2082.   137882.     3 <tibble [5~ <tibble [476,1~
## 5     6 <kmean~ 139964.        1510.   138454.     4 <tibble [6~ <tibble [476,1~
## 6     7 <kmean~ 139964.        1119.   138845.     3 <tibble [7~ <tibble [476,1~
```

```
## 7     8 <kmean~ 139964.         973.   138991.     5 <tibble [8~ <tibble [476,1~
```

The third step is to produce and plot a scree plot. We use the function scree_plot() and recieve one graph with two plots.

The within-cluster sum of squares used to measure the variability of the observations within each cluster

The left graph shows that the total within-cluster sum of squares is decreasing slowly after k=4. After k=5,the total within-cluster sum of squares is close to the results in k=5. The value at k=5 is good.

The right graph shows the change of ratio(=tot.withinss(k)/tot.withinss(k-1)).

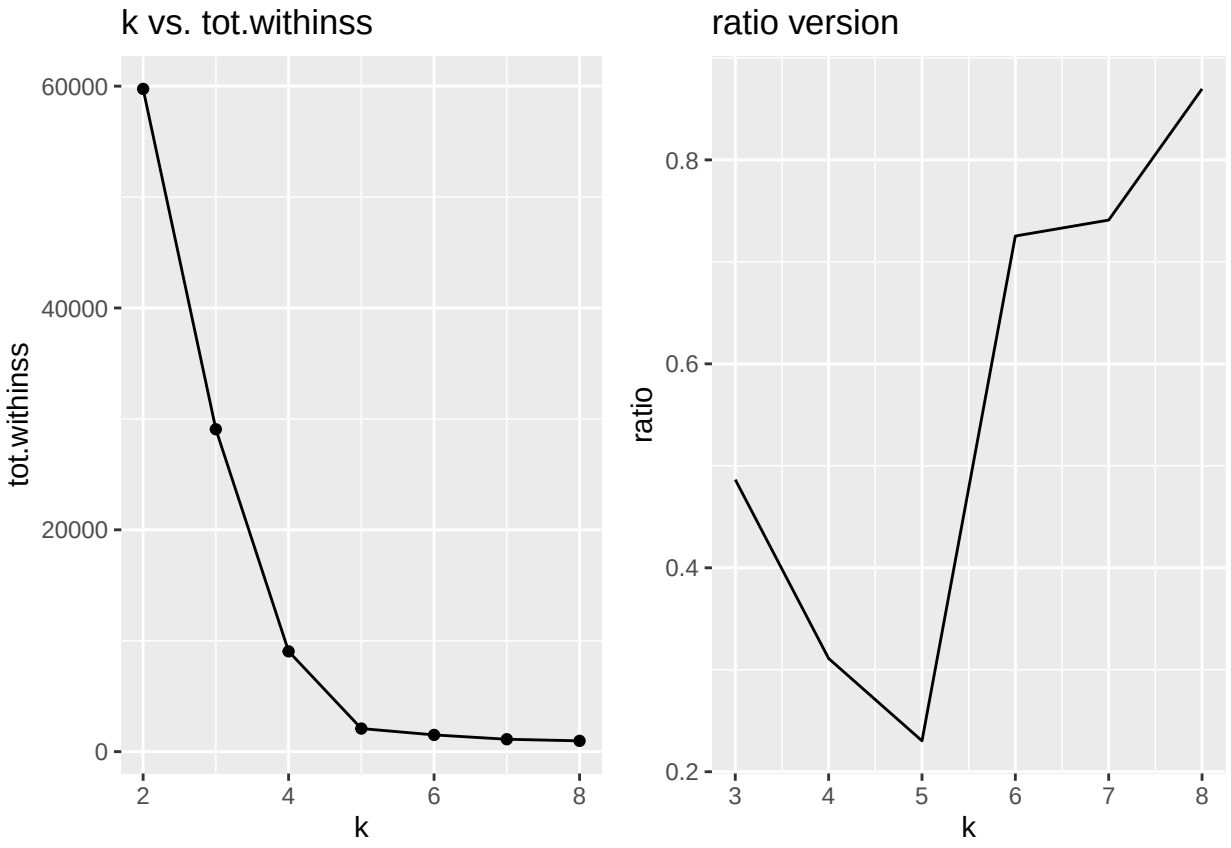
When k=7,ratio=tot.withinss(7)/tot.withinss(6)=0.74. The tot.withinss(k=7) is 72% of tot.withinss(k=6).

When k=6,ratio=tot.withinss(6)/tot.withinss(5)=0.72. The tot.withinss(k=6) is 72% of tot.withinss(k=5).

When k=5,ratio=tot.withinss(5)/tot.withinss(4)=0.22. The tot.withinss(k=5) is 22% of tot.withinss(k=4).

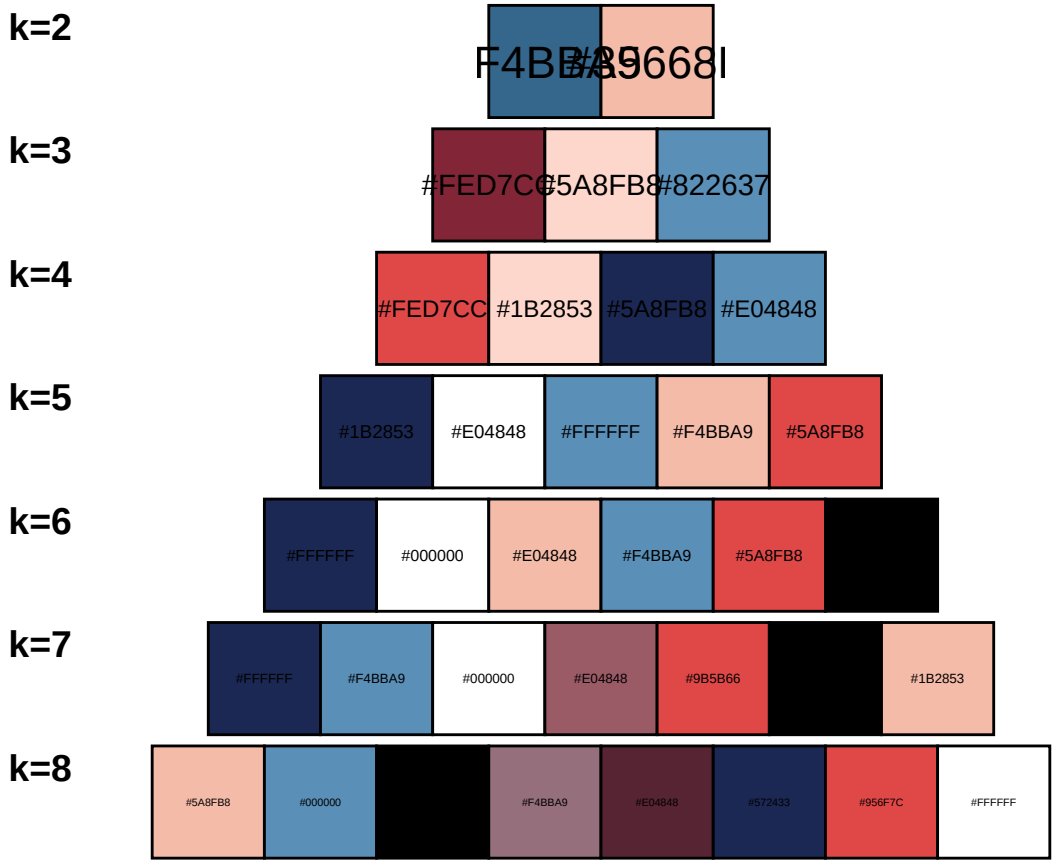
From this the number of clusters seems to be 6 or 7 (because increase trend becomes slow). And the tot.withinss is close when k=5,6 and 7. So we cannot have a certain conclusion that 6 is the right number of clusters. We will check it in following colour_strips.

```
scree_plot(gain_clusters)
```



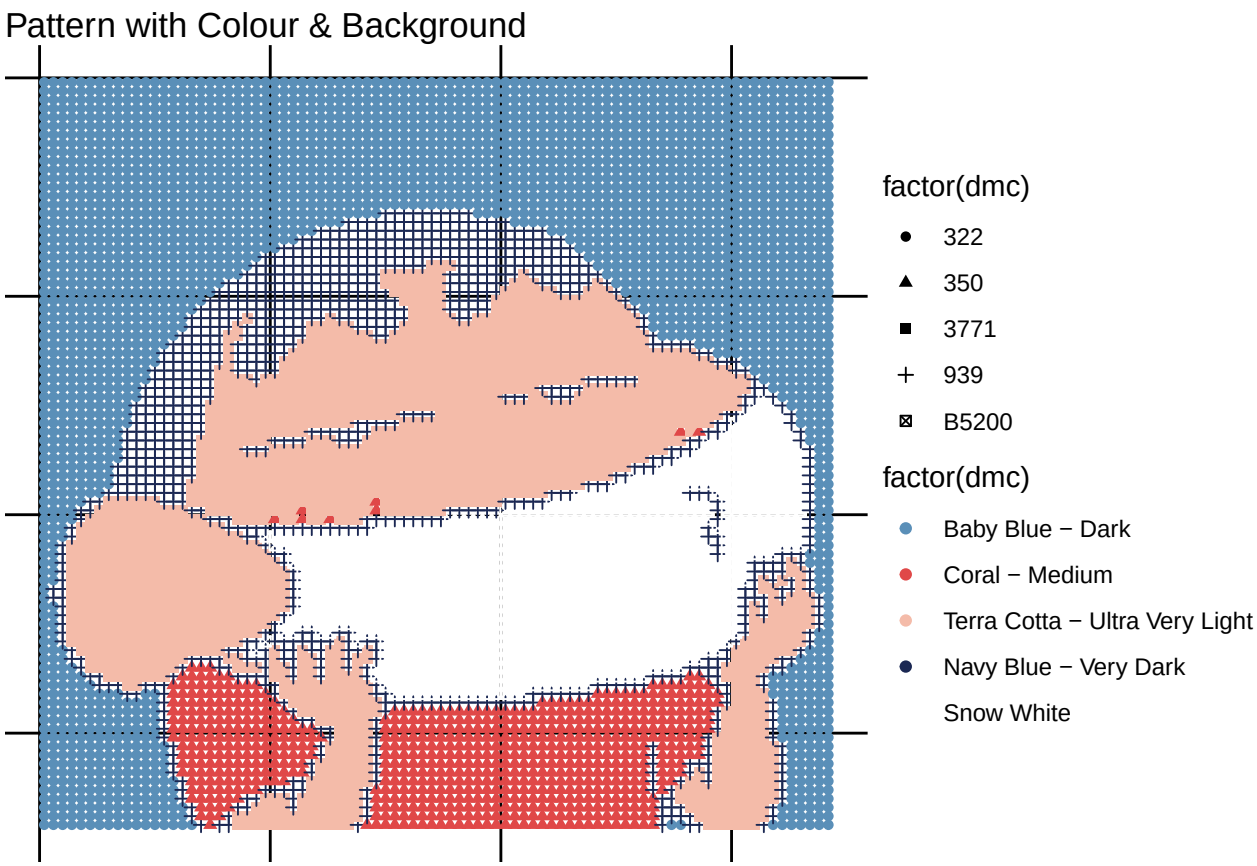
We use function colour_strips() to produce colour strips with the DMC colour. Visually,when k=6, color black and color dark blue is too similar (dark tones) . Let's see what happens if we choose k-5, 5 colors are clearly unlike each other. So K=5 is much better choice.

```
colour_strips(gain_clusters)
```



So now we have 5 clusters we need to put it into make_pattern() function, and produce a cross stitch pattern that can be followed. I graph 2 versions, one with color and one without color.

```
#Pattern with Colour & Background
make_pattern(cluster_info=gain_clusters, k=5, x_size=90,
             black_white =FALSE,background_colour=NULL)+
  ggtitle("Pattern with Colour & Background")+
  background_grid(color.major = "black",color.minor = "black")
```




```
#Pattern with Background & no color
make_pattern(cluster_info=gain_clusters, k=5, x_size=50,
             black_white =TRUE,background_colour=NULL)+
  ggtitle("Pattern with Colour & Background")+
  background_grid(color.major = "black",color.minor = "black")
```

Pattern with Colour & Background