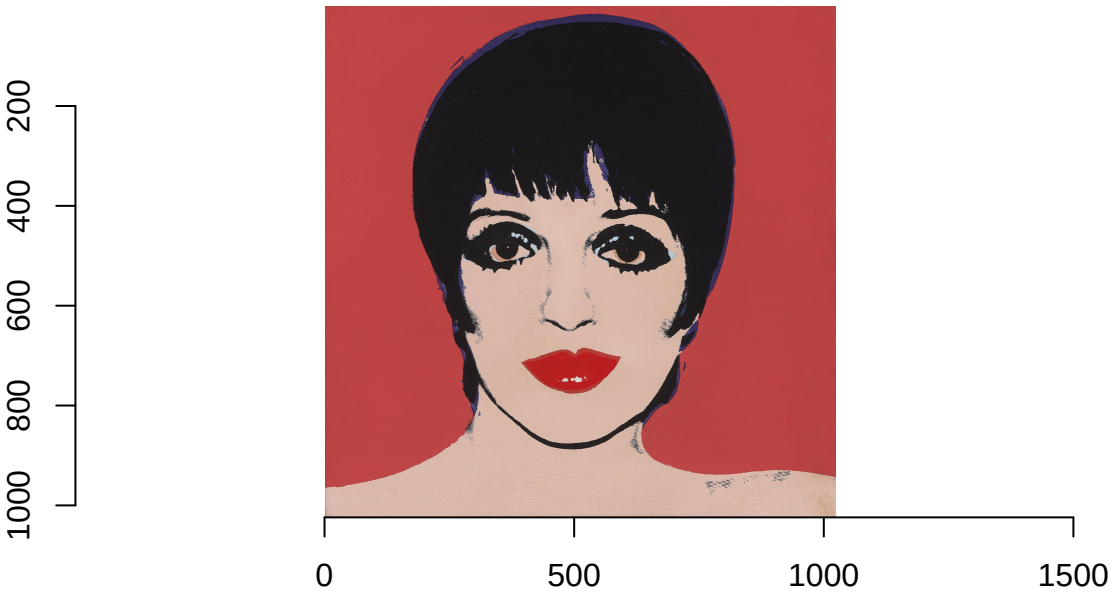
# Clustering

Jin Seo Jo

We will use Andy Warhol's iconic portrait: Liza Minnelli

```
library(imager)

im <- imager::load.image("liza_minnelli_andy_warhol_collection.jpg")
plot(im)
```




This picture is stored as a 'cimg' object, which is bascially a 4-dimensional array.
The first index is the horizontal pixel, the second is the vertical, the third is the opacity, and the fourth is the colour (R,G,B)

And we can turn this into something useful for clustering by using the `as.data.frame` method with the option wide = "c" (This option only works for a 'cimg' object.) We then rename the three colours to "R", "G", and "B" using 'rename' function from 'dplyr'.

```
library(dplyr)

tidy_data <- as.data.frame(im, wide = "c") %>%
  rename(R = c.1, G = c.2, B = c.3)

head(tidy_data, 5)
```

```
##   x y         R         G         B
## 1 1 1 0.7960784 0.5647059 0.5411765
## 2 2 1 0.5882353 0.3019608 0.2745098
```

```
## 3 3 1 0.6470588 0.2784314 0.2470588
## 4 4 1 0.6941176 0.2588235 0.2235294
## 5 5 1 0.7098039 0.2509804 0.2156863
```

Because 'class' has type `cimg` (type `class(im)` to confirm), when we call `as.data.frame` R finds the version of `as.data.frame` that works on on that type of object. In this case it finds the internal function `imager:::as.data.frame.cimg()`, which has the `wide = "c"` argument. (The three :s means that the function is internal to the package.)

We now have the data in the foremat required to the clustering. Explore various k-means clustering using the template laid out in the Learning K-Means with tidy data principles vignette.

First things first, let's make the scree plot.

```
library(purrr)
library(tidymodels)

dat <- select(tidy_data, c(-x, -y))

kclusts <- tibble(k = c(2:10)) %>%
  mutate(
    kclust = map(k, ~kmeans(x = dat, centers = .x, nstart = 4)),
    glanced = map(kclust, glance),
  )

str(kclusts)
```

```
## tibble[,3] [9 x 3] (S3: tbl_df/tbl/data.frame)
##  $ k      : int [1:9] 2 3 4 5 6 7 8 9 10
##  $ kclust :List of 9
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 2 2 2 2 2 2 2 2 2 ...
##   .. ..$ centers     : num [1:2, 1:3] 0.117 0.775 0.109 0.425 0.125 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "1" "2"
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:2] 1645 69194
##   .. ..$ tot.withinss: num 70839
##   .. ..$ betweenss   : num 122420
##   .. ..$ size        : int [1:2] 268517 780059
##   .. ..$ iter        : int 1
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 1 2 2 2 2 2 2 2 2 ...
##   .. ..$ centers     : num [1:3, 1:3] 0.848 0.735 0.116 0.72 0.265 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:3] "1" "2" "3"
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:3] 848 1361 1522
##   .. ..$ tot.withinss: num 3732
##   .. ..$ betweenss   : num 189527
```

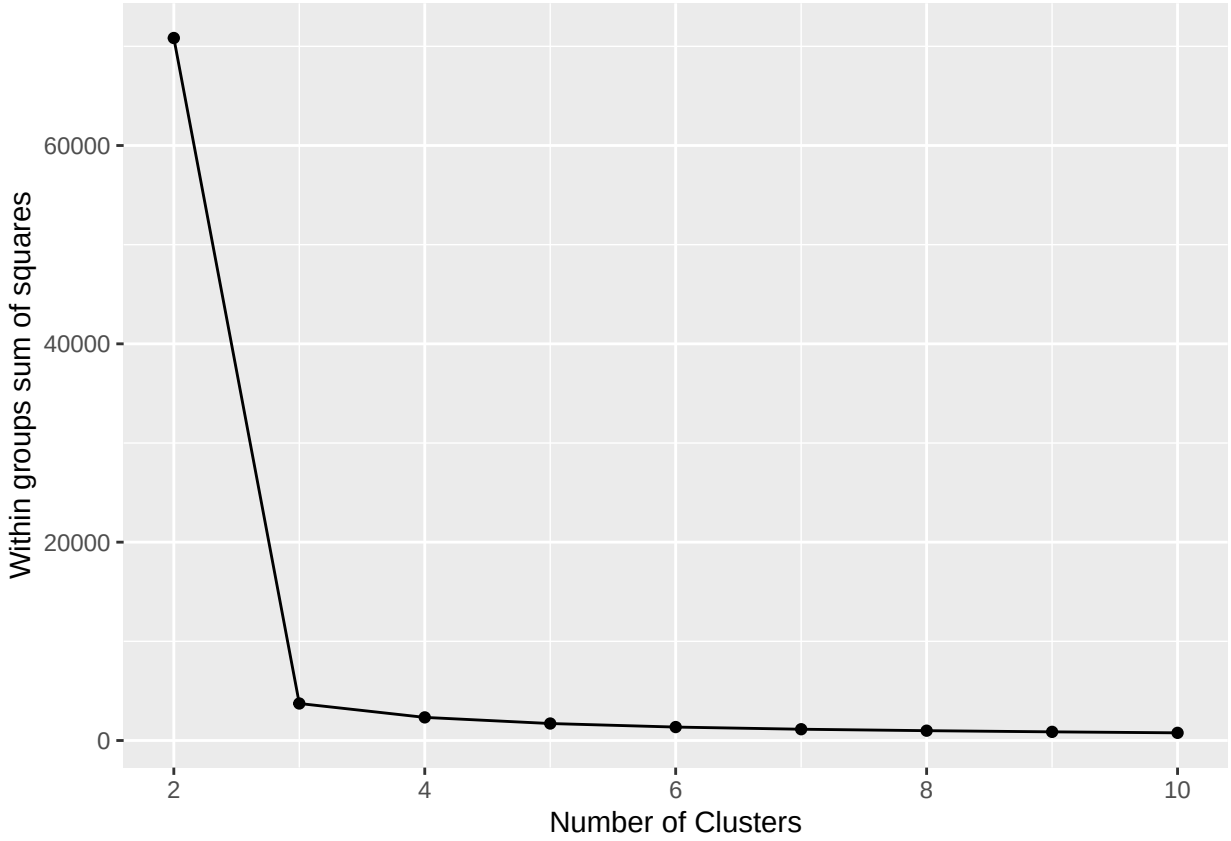
```
##   .. ..$ size        : int [1:3] 273940 506949 267687
##   .. ..$ iter        : int 3
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 3 2 2 2 2 2 2 2 2 2 ...
##   .. ..$ centers     : num [1:4, 1:3] 0.107 0.737 0.848 0.275 0.102 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:4] 212 816 846 457
##   .. ..$ tot.withinss: num 2331
##   .. ..$ betweenss   : num 190928
##   .. ..$ size        : int [1:4] 249609 502899 273935 22133
##   .. ..$ iter        : int 4
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 3 5 5 5 5 5 5 5 5 5 ...
##   .. ..$ centers     : num [1:5, 1:3] 0.107 0.254 0.851 0.593 0.738 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:5] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:5] 197 258 467 166 624
##   .. ..$ tot.withinss: num 1712
##   .. ..$ betweenss   : num 191547
##   .. ..$ size        : int [1:5] 249018 20617 269307 8381 501253
##   .. ..$ iter        : int 3
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 2 5 5 5 5 5 5 5 5 5 ...
##   .. ..$ centers     : num [1:6, 1:3] 0.711 0.851 0.593 0.107 0.738 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:6] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:6] 24.7 467.1 166 197.6 248 ...
##   .. ..$ tot.withinss: num 1355
##   .. ..$ betweenss   : num 191904
##   .. ..$ size        : int [1:6] 9765 269309 8381 249021 491565 20535
##   .. ..$ iter        : int 5
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 7 5 5 5 5 5 5 5 5 5 ...
##   .. ..$ centers     : num [1:7, 1:3] 0.711 0.251 0.107 0.567 0.738 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:7] 24.9 234.1 195.9 142.2 248.7 ...
```

```
##   .. ..$ tot.withinss: num 1138
##   .. ..$ betweenss   : num 192121
##   .. ..$ size        : int [1:7] 9767 20310 248942 7555 491573 186232 84197
##   .. ..$ iter        : int 4
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 1 3 3 3 3 3 3 3 3 3 ...
##   .. ..$ centers     : num [1:8, 1:3] 0.645 0.107 0.738 0.414 0.711 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:8] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:8] 81.2 187.2 237.8 73.2 25.5 ...
##   .. ..$ tot.withinss: num 991
##   .. ..$ betweenss   : num 192268
##   .. ..$ size        : int [1:8] 6837 248498 491372 5088 9780 167572 18286 101143
##   .. ..$ iter        : int 5
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 6 1 1 1 1 1 1 1 1 1 ...
##   .. ..$ centers     : num [1:9, 1:3] 0.738 0.145 0.83 0.415 0.237 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:9] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:9] 237.6 50 90.5 70.8 121.1 ...
##   .. ..$ tot.withinss: num 872
##   .. ..$ betweenss   : num 192387
##   .. ..$ size        : int [1:9] 491367 40656 82972 4987 17027 6690 9783 185921 209173
##   .. ..$ iter        : int 5
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##   ..$ :List of 9
##   .. ..$ cluster     : int [1:1048576] 1 9 9 9 9 9 9 9 9 9 ...
##   .. ..$ centers     : num [1:10, 1:3] 0.642 0.1 0.237 0.83 0.414 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:10] "1" "2" "3" "4" ...
##   .. .. .. ..$ : chr [1:3] "R" "G" "B"
##   .. ..$ totss       : num 193259
##   .. ..$ withinss    : num [1:10] 77.6 40 123 90.1 66.3 ...
##   .. ..$ tot.withinss: num 773
##   .. ..$ betweenss   : num 192486
##   .. ..$ size        : int [1:10] 6668 209173 17081 82950 4889 40654 185921 9482 181838 309920
##   .. ..$ iter        : int 8
##   .. ..$ ifault      : int 0
##   .. ..- attr(*, "class")= chr "kmeans"
##  $ glanced:List of 9
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 70839
##   .. ..$ betweenss   : num 122420
##   .. ..$ iter        : int 1
```

```
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 3732
##   .. ..$ betweenss   : num 189527
##   .. ..$ iter        : int 3
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 2331
##   .. ..$ betweenss   : num 190928
##   .. ..$ iter        : int 4
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 1712
##   .. ..$ betweenss   : num 191547
##   .. ..$ iter        : int 3
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 1355
##   .. ..$ betweenss   : num 191904
##   .. ..$ iter        : int 5
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 1138
##   .. ..$ betweenss   : num 192121
##   .. ..$ iter        : int 4
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 991
##   .. ..$ betweenss   : num 192268
##   .. ..$ iter        : int 5
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 872
##   .. ..$ betweenss   : num 192387
##   .. ..$ iter        : int 5
##   ..$ : tibble[,4] [1 x 4] (S3: tbl_df/tbl/data.frame)
##   .. ..$ totss       : num 193259
##   .. ..$ tot.withinss: num 773
##   .. ..$ betweenss   : num 192486
##   .. ..$ iter        : int 8
```

```
clusterings <- kclusts %>%
  unnest(cols = c(glanced))

ggplot(clusterings, aes(k, tot.withinss)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of Clusters", y = "Within groups sum of squares")
```



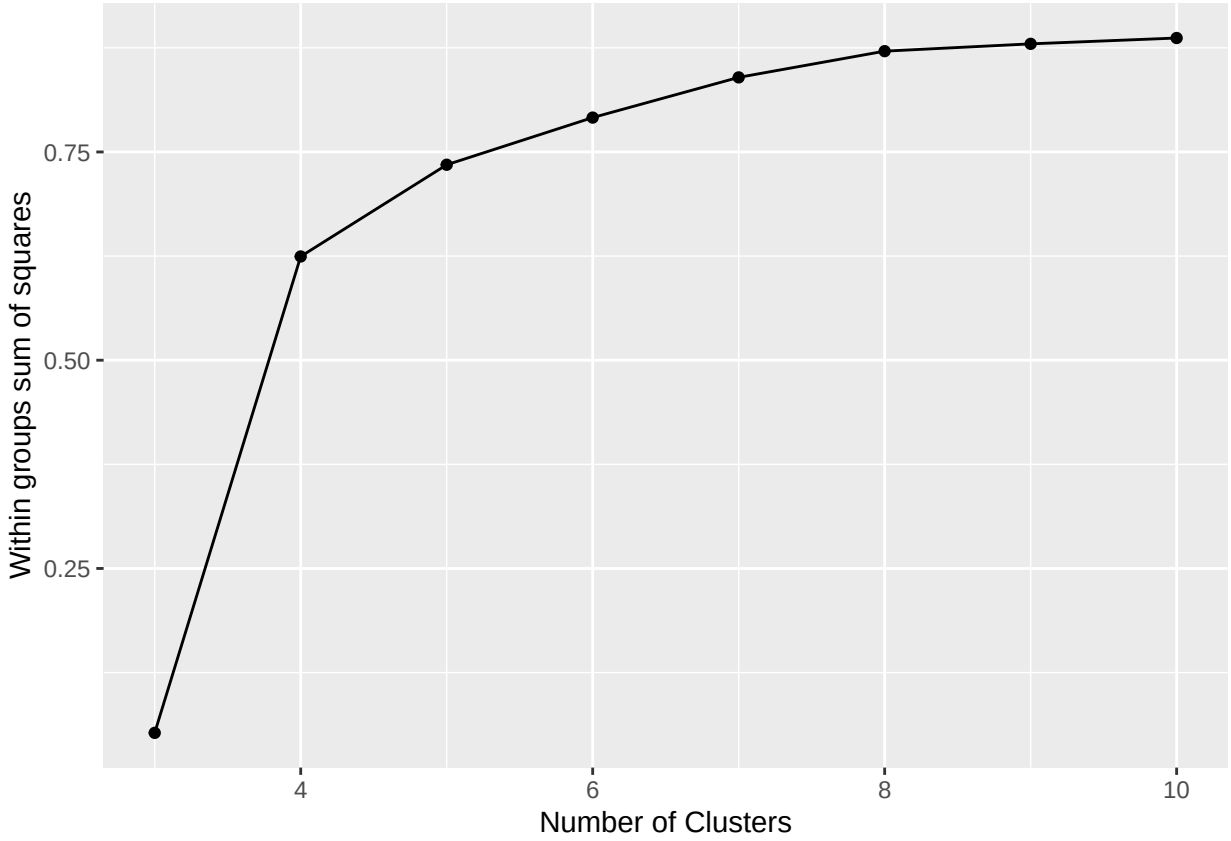

Maybe 6 is the right number of clusters, but it's hard to tell. Hence, we will try the ratio version.

```
nclust = length(clusterings$k)
ratio = rep(NA, nclust-1)

for (kk in 2:nclust) {
  ratio[kk-1] = clusterings$tot.withinss[kk]/clusterings$tot.withinss[kk-1]
}

plot_data <- data.frame(k = clusterings$k[2:nclust], ratio)

ggplot(plot_data, aes(x = k, y = ratio)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of Clusters", y = "Within groups sum of squares")
```




From this the number of clusters seems to be six. So let's use that going forward.

First, let's re-do the clustering and save the centres.

```
k <- 7
kclust <- kmeans(select(tidy_data, -x, -y), centers = k, nstart = 20)
centres <- tidy(kclust)
```

We can also add a column to the tidied centres to add the colour in a way that we can use for plots. The `rgb` function will do this and display the colour as a hex string.

```
centres <- centres %>%
  mutate(col = rgb(R, G, B))

centres
```

```
## # A tibble: 7 x 7
##       R     G     B   size withinss cluster col
##   <dbl> <dbl> <dbl>  <int>    <dbl> <fct>   <chr>
## 1 0.107 0.102 0.110 248942    196.  1       #1B1A1C
## 2 0.711 0.130 0.135   9767     24.9 2       #B52122
## 3 0.567 0.500 0.492   7555    142.  3       #90807E
## 4 0.738 0.266 0.270 491573    249.  4       #BC4445
## 5 0.861 0.733 0.680 186232    158.  5       #DCBBAD
## 6 0.827 0.698 0.641  84197    134.  6       #D3B2A3
## 7 0.251 0.203 0.315  20310    234.  7       #403450
```

It's probably worth seeing what the colours are. In this case, we will use `show_col` from `scales`.

```
library(scales)
show_col(centres$col)
```


Visually, we can see that two of these colours are skin tones. Let's see what happens if we choose 6 colours.

```
kclust6 <- kmeans(select(tidy_data, -x, -y), centers = 6, nstart = 20)

centres6 <- tidy(kclust6)

centres6 <- centres6 %>%
  mutate(col = rgb(R, G, B))

show_col(centres6$col)
```


It's slightly different but probably better. This is one of those cases where the scree plot can be misleading and using visualizations can help.

So now we have six clusters we need to put the do the cluster centre replacement. To do this, we first need to augment the initial data with the clusters. We can do this with `broom::augment` function (`broom` is a package loaded by `tidymodels`). The `rename` command just makes the naming a little nicer.

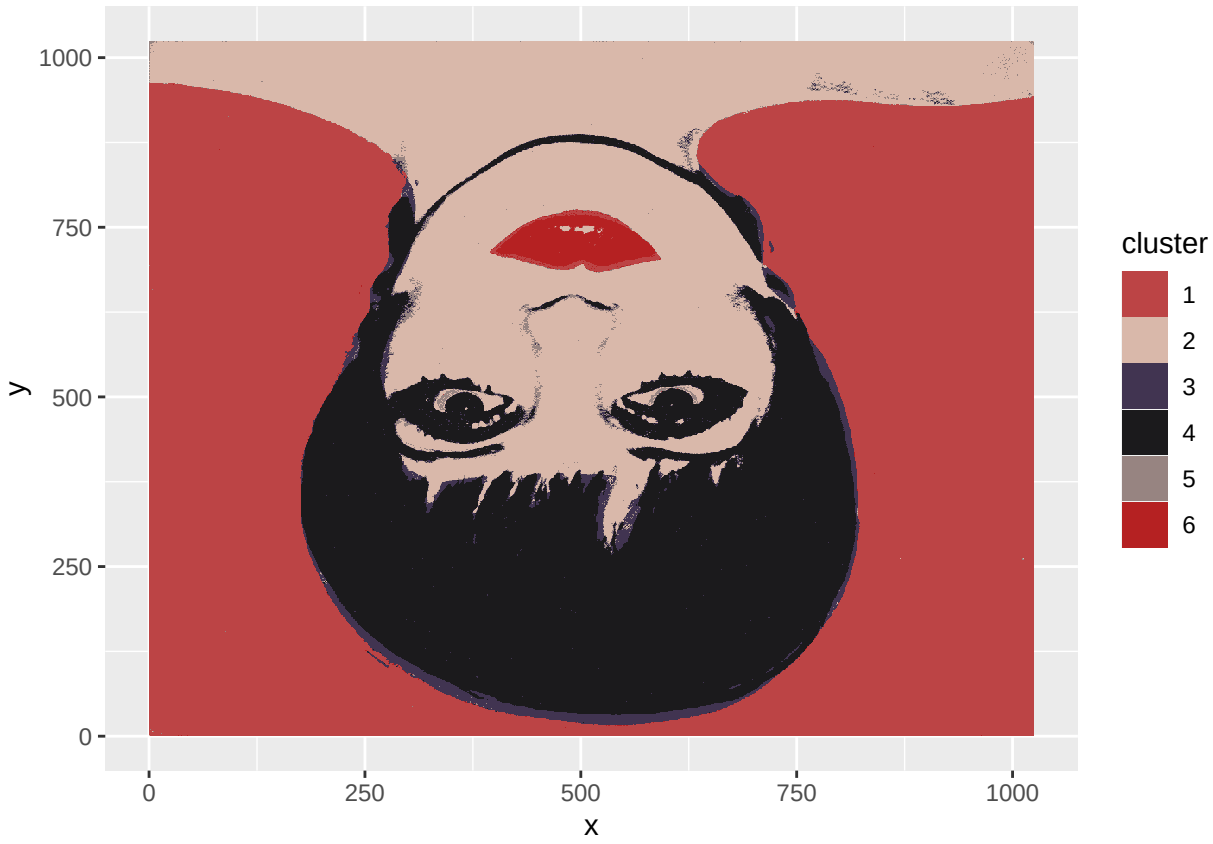
```
tidy_data <- augment(kclust6, tidy_data) %>%
  rename(cluster = .cluster)

glimpse(tidy_data)
```

```
## Rows: 1,048,576
## Columns: 6
## $ x       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
## $ y       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ R       <dbl> 0.7960784, 0.5882353, 0.6470588, 0.6941176, 0.7098039, 0.71764~
## $ G       <dbl> 0.5647059, 0.3019608, 0.2784314, 0.2588235, 0.2509804, 0.25882~
## $ B       <dbl> 0.5411765, 0.2745098, 0.2470588, 0.2235294, 0.2156863, 0.23137~
## $ cluster <fct> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
```

We can now plot the clustered picture.

```
ggplot(tidy_data, aes(x = x, y = y, fill = cluster)) +
  geom_tile() +
  scale_discrete_manual(aesthetics = "fill", values = centres6$col)
```




We can see that Liza is upside down.

```
ggplot(tidy_data, aes(x = x, y = y, fill = cluster)) +
  geom_tile() +
  scale_discrete_manual(aesthetics = "fill", values = centres6$col) +
  scale_y_reverse() +
  theme_void()
```


cluster
1
2
3
4
5
6