Colors

Ervin Madaha, Jennifer Wilken, Shuaib Yunus

5/16/2018

Setup

Load and format the data in a reusable way:

```r
masterCard<-read.csv2("MasterColorCard.csv")

raw<-read.csv2("LabMeasurements-Color-Card.csv")

sheets<-rep(1:13,42)
raw$sheet<-sheets

column_names<-names(raw)[names(raw)!=c('Row','Column','sheet')]
l_data<-reshape(raw,varying = column_names, direction= "long", sep="" )
l_data$Xc<-floor(l_data$time/10)
l_data$Yc<-l_data$time-10*l_data$Xc
l_data$time<-NULL

l_data$X<-max(l_data$Xc)*(l_data$Row-1)+l_data$Xc
l_data$Y<-max(l_data$Yc)*(l_data$Column-1)+l_data$Yc

#Join the 2 data frames based on color coordinates (8x8)
joined_d<-merge(l_data,
                masterCard,
                by.x=c('Xc','Yc'),
                by.y=c('Crow','Ccol'),suffixes = c("_measured","_master"))
#center lab on master
joined_d$L_centered<-joined_d$L_measured-joined_d$L_master
joined_d$a_centered<-joined_d$a_measured-joined_d$a_master
joined_d$b_centered<-joined_d$b_measured-joined_d$b_master

#Compute sd based on center
L_sd<-sqrt(sum(joined_d$L_centered**2)/dim(joined_d)[1])
a_sd<-sqrt(sum(joined_d$a_centered**2)/dim(joined_d)[1])
b_sd<-sqrt(sum(joined_d$b_centered**2)/dim(joined_d)[1])
joined_d$L_scaled<-joined_d$L_centered/L_sd
joined_d$a_scaled<-joined_d$a_centered/a_sd
joined_d$b_scaled<-joined_d$b_centered/b_sd

#Most likely useless for pca
joined_d$X_scaled<-(joined_d$X-mean(joined_d$X))/sd(joined_d$X)
joined_d$Y_scaled<-(joined_d$Y-mean(joined_d$Y))/sd(joined_d$Y)
joined_d$sheet_scaled<-(joined_d$sheet-mean(joined_d$sheet))/sd(joined_d$sheet)
```

Delta E computation

Load the library we'll be using:

```
library("colorscience")
```

Let's create a convenience function for getting a Delta E (DE) vector from 2 L,a,b dataframes:

```
getDEvector <-function(x, y){
  res <- 1:dim(x)[1]
  for (i in 1:dim(x)[1]){
    res[i] <- deltaE2000(data.matrix(x[i,]),data.matrix(y[i,]))
  }
  return(res)
}
```

Let's add a column for Delta E values from L, a, b values of measured versus master to our dataframe:

```
joined_d$DE <-
  getDEvector(joined_d[, c("L_measured", "a_measured", "b_measured")],
              joined_d[, c("L_master", "a_master", "b_master")])
```

Center and scale the Delta E column:

```
DE_sd<-sqrt(sum((joined_d$DE-mean(joined_d$DE))**2)/dim(joined_d)[1])
joined_d$DE_scaled <- (joined_d$DE -mean(joined_d$DE))/DE_sd
```

Create a 64x546 matrix of DE values to recreate the color card DEM (DE Matrix). Each row is a color on the 8x8 color card and columns are each color card (13x42(7x6))

```
DEM <-  matrix(joined_d$DE,64, byrow = TRUE)
DEM_mean <- rowMeans(DEM) # mean of each row to reduce each color to a single value
DEM_sd <- apply(DEM,1,sd) # Interesting to see how each color varies
```

Create an 8x8 matrix representing a card:

```
DEM_avgcard <- matrix(DEM_mean,8,byrow = TRUE)
DEM_sd_scaledcard <- matrix(DEM_sd/max(DEM_sd),8,byrow = TRUE)
```

Plots

One way to spread the color to express the DE spread across the color card:

```
colrange <- colorRampPalette(c("yellowgreen","yellow","red"))(100)
```

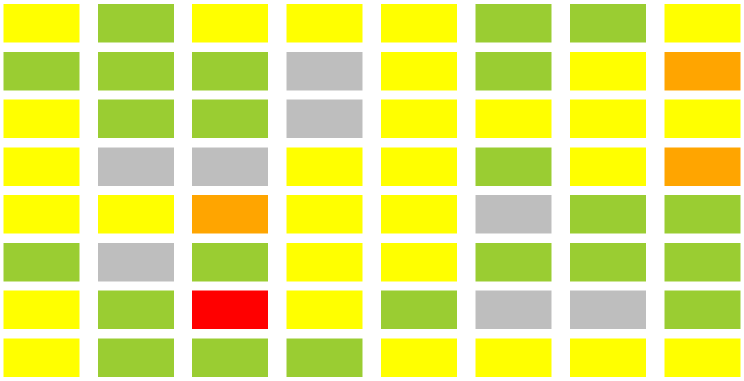
Another way to express the color based on the relative human distinction:

```
getDEcolor <-function(x){
  if (x < 1) "grey" # Unnoticable to the human eye
  else if (1 < x && x < 2) "yellowgreen" # barely noticable
  else if (2 < x && x < 4) "yellow" # noticable difference on a keen eye
  else if  (4 < x && x < 10) "orange" # distinguishable
  else if (10 < x ) "red" # clearly distinct
}
```

Plot DE values with the 2nd color scheme only half:

```
xspace<-cbind(1:100,1:100)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")
start<-5
width<-8
spacing<-2
```

```
for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    rect(xleft,ybottom,xleft+width,ybottom+width,col=getDEcolor(DEM_avgcard[i,j]),border = FALSE)
  }
}
```

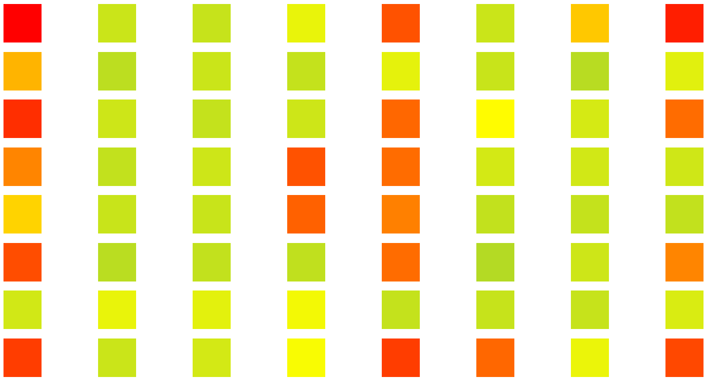


Plot the right half of colors with sd following color scheme 1:

```
xspace<-cbind(1:100,1:100)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")
start<-5
width<-8
spacing<-2

for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    rect(xleft+width/2,ybottom,xleft+width,ybottom+width,col=colrange[round(DEM_sd_scaledcard[i,j]*100)]

  }
}
```



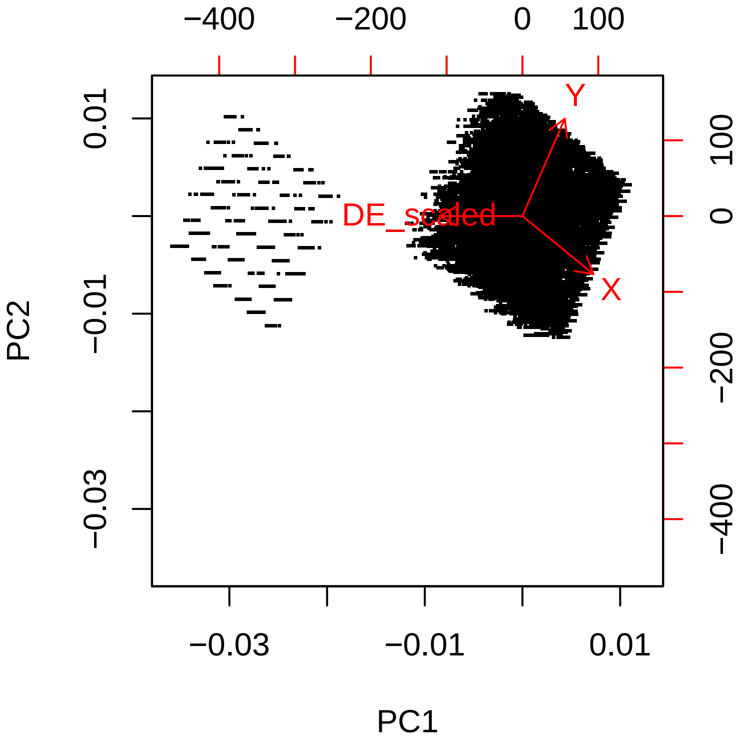
Perform PCA on the data, which tells us that there are some funny outliers:

```
summarized<-aggregate(.~joined_d$Xc+joined_d$Yc,joined_d,FUN = mean)
summarized$dE<-0
for(i in 1:dim(summarized)[1]){
```

```r
  summarized$dE[i]<-deltaE2000(c(summarized$L_measured[i],
                                 summarized$a_measured[i],
                                 summarized$b_measured[i]),
                               c(summarized$L_master[i],
                                 summarized$a_master[i],
                                 summarized$b_master[i]))
}


summarized$dE_scaled<-(summarized$dE-mean(summarized$dE))/sd(summarized$dE)

pca.out<-prcomp(joined_d[c("DE_scaled","X","Y")],scale. = TRUE)
biplot(pca.out, xlabs=rep("·", nrow(joined_d)))
```



Try a different color scheme:

```r
xspace<-cbind(1:100,1:100)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")
start<-5
width<-8
spacing<-2

dEpallette <- colorRampPalette(c("green", "red"))(max(ceiling(log(summarized$dE))))

for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    log_dE<-log(summarized$dE[which(summarized$Xc==i&summarized$Yc==j)])
    rect(xleft,ybottom,xleft+width,ybottom+width,col=dEpallette[ceiling(log_dE)],border = FALSE)
  }
}
```