

Colors

Ervin Madaha, Jennifer Wilken, Shuaib Yunus

5/21/2018

Setup

Load and format the data in a reusable way:

The data is formatted in a long table format where each row is to represent a single color thus all computations can be made row wise without any matrix shifting. Additional columns were added such as sheets, Xc,Yc (as color card coordinates per color), X,Y (as sheet coordinates per color)

```
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)
```

Plotting Means, Variances of the DE values

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" # Unnoticeable to the human eye
  else if (1 < x && x < 2) "yellowgreen" # barely noticeable
  else if (2 < x && x < 3) "yellow" # noticeable difference on a keen eye
  else if (3 < x && x < 10) "orange" # distinguishable
  else if (10 < x ) "red" # clearly distinct
}
```

Plot DE values with the 2nd color scheme:

```
xspace<-cbind(1:100,1:100)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")
```

```

start<-5
width<-8
spacing<-2
title(" Plot of mean DE values in the 8x8 scheme of the colorcard")
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 of mean DE values in the 8x8 scheme of the colorcard



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<-10
spacing<-1
title("Plot of mean(left) vs the std deviation(right) of DE values")
for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    rect(xleft,ybottom,xleft+width/2,ybottom+width,col=getDEcolor(DEM_avgcard[i,j]),
        border = FALSE)
  }
}

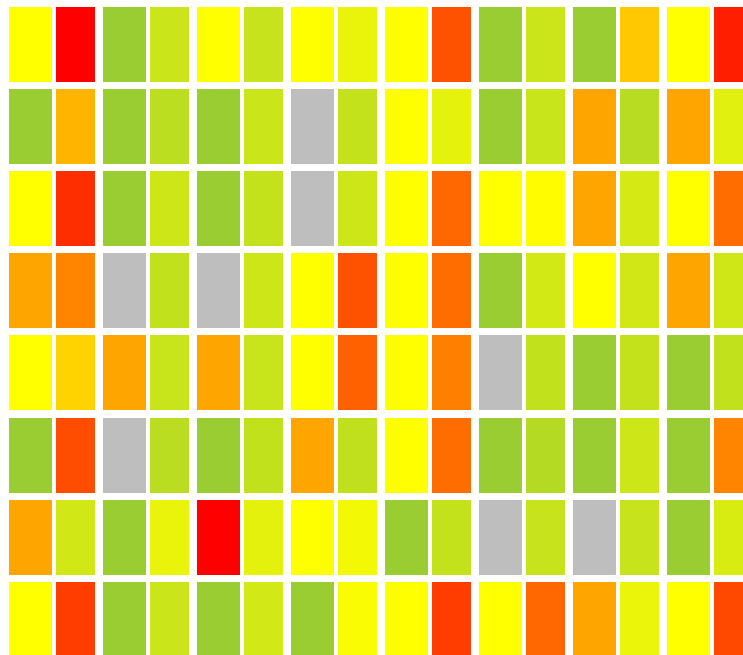
```

```

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

```

Plot of mean(left) vs the std deviation(right) of DE values



Perform PCA on the data, which tells us that there is one funny outliers:

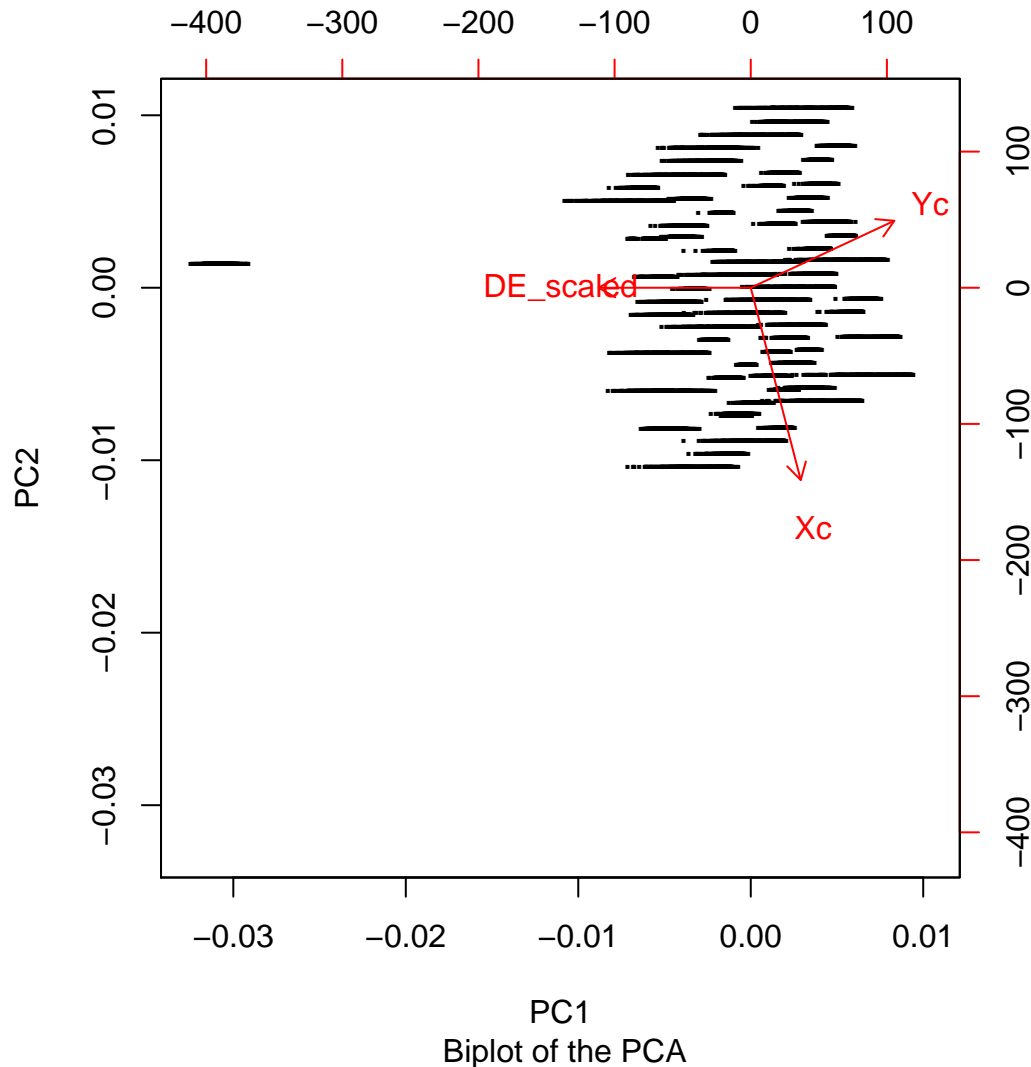
```

summarized<-aggregate(.~joined_d$Xc+joined_d$Yc,joined_d,FUN = mean)
summarized$dE<-0
for(i in 1:dim(summarized)[1]){
  summarized$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","Xc","Yc")],scale. = TRUE)
biplot(pca.out, xlab=rep(".", nrow(joined_d)))
title(sub = " Biplot of the PCA")
```



Try a logarithmic scale:

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

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

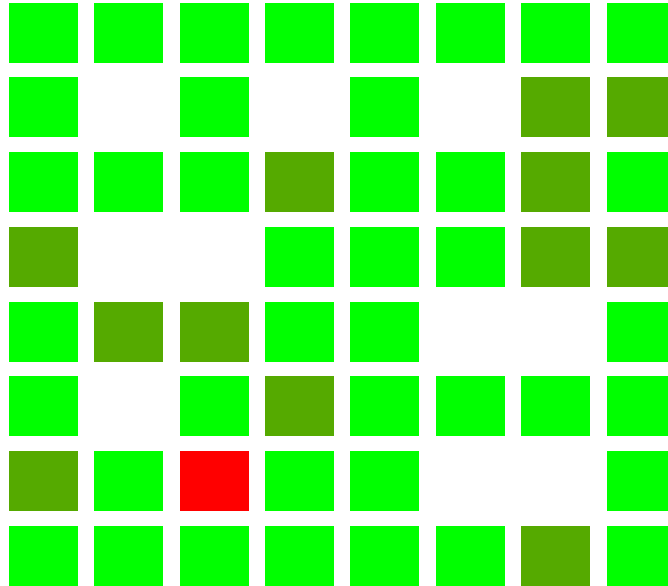
for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    log_dE<-log(summarized$dE[which(summarized$Xc==i&summarized$Yc==j)])
    rect(xleft,ybottom,xleft+width,ybottom+width,col=dEpalette[ceiling(log_dE)],
```

```

    border = FALSE)
  }
}
title("Plot of the DE values in log scale")

```

Plot of the DE values in log scale



Plot the means and variances of all sheets (right: means, left: variances).

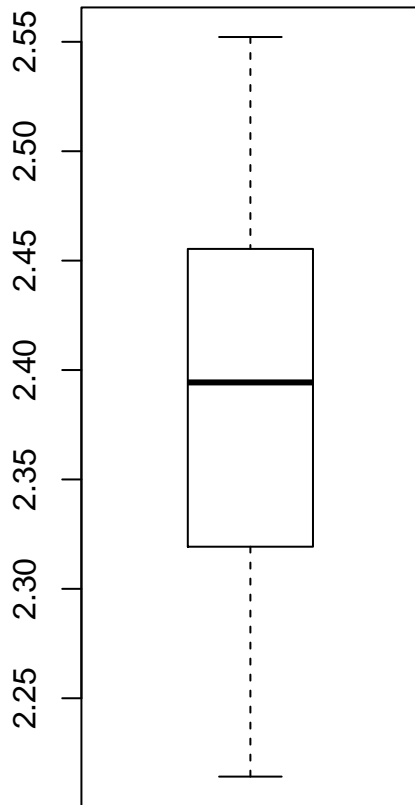
```

par(mfrow=c(1,2))
sheets <- matrix(nrow=13,ncol=2688)
sheetMeans <- c()
sheetVars <- c()

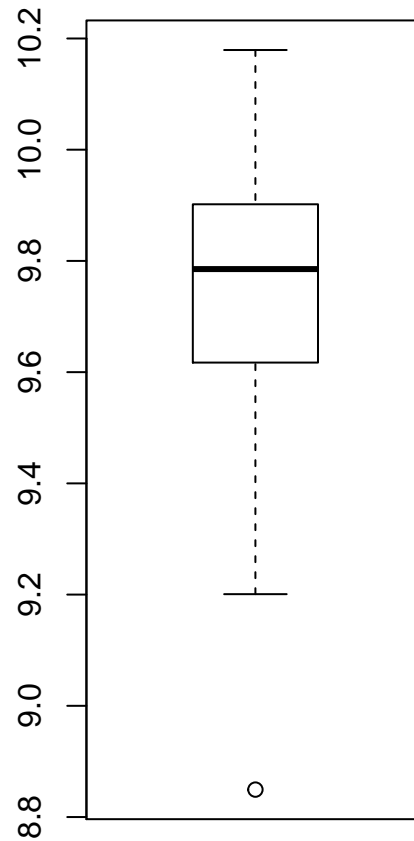
for (i in 1:13) {
  sheets[i,] <- joined_d$DE[which(joined_d$sheet==i)]
  sheetMeans[i] <- mean(joined_d$DE[which(joined_d$sheet==i)])
  sheetVars[i] <- var(joined_d$DE[which(joined_d$sheet==i)])
}

boxplot(sheetMeans, xlab = "Means of all DE values per sheet" )
boxplot(sheetVars, xlab = "Variance of all DE values per sheet")

```



Means of all DE values per sheet



Variance of all DE values per sheet

```
par(mfrow=c(1,1))
```

Color Comparison in RGB

Building the mean of L,a,b of all printed colorcards (8x8)

```
Xc <- rep(1:8, each =8)
Yc <- rep(1:8,8)
Labavg <- data.frame(Xc,Yc)
L <- zeros(8*8,1)
a <- zeros(8*8,1)
b <- zeros(8*8,1)

for( i in 1:8){
  row <- joined_d[joined_d$Xc == i,]
  for( j in 1:8){
    color <- row[row$Yc == j,]
    L[(i-1)*8+j] <- mean(color$L_measured)
    a[(i-1)*8+j] <- mean(color$a_measured)
    b[(i-1)*8+j] <- mean(color$b_measured)
  }
}
```

```

Labavg$L <- L
Labavg$a <- a
Labavg$b <- b

```

Conversion of Lab into RGB for master and measured color cards (some colors are truncated to fit RGB from Lab scale) And a Display of the master color card(left) vs the average of all printed color cards (right) :

```

xspace<-cbind(1:100,1:100)
plot(xspace,type="n",axes=FALSE,
      xlab = " Mean of measured(right) vs Master(left) colors ",ylab = "")
title("Color card comparison")
start<-5
width<-8
spacing<- 2

labVals <- c()
for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    labVals <- Labavg[(Labavg$Xc==i & Labavg$Yc==j), ][,c("L", "a", "b")]

    b <- convertColor(c(labVals$L, labVals$a, labVals$b),
                      from="Lab",to="Apple RGB",clip=TRUE)

    tryCatch({
      cola <- rgb(b[1,][1], b[1,][2], b[1,][3])
      rect(xleft,ybottom,xleft+width/2-spacing/8,ybottom+width,col=cola,border = FALSE)
    }, warning = function(w) {
    }, error = function(e) {print(c(i,j,b))})
  }
}

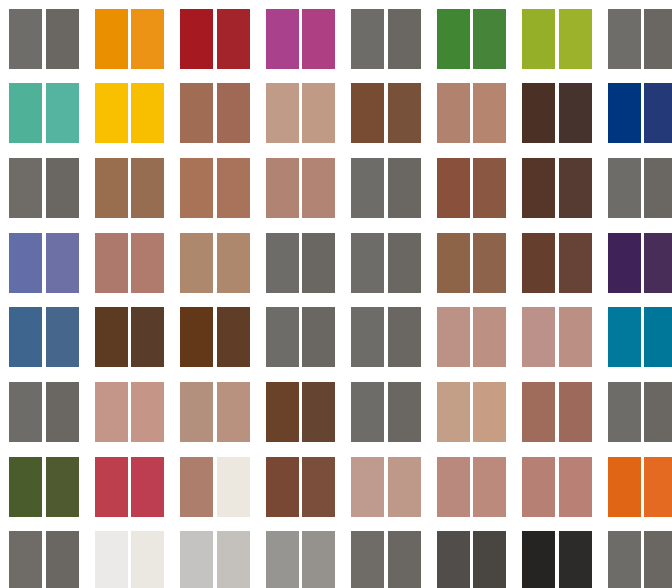
labVals <- c()
for( i in 1:8){
  for( j in 1:8){
    xleft<-i*(width+spacing)
    ybottom<-j*(width+spacing)
    labVals <- masterCard[(masterCard$Crow==i & masterCard$Ccol==j), ][,c("L", "a", "b")]

    b <- convertColor(c(labVals$L, labVals$a, labVals$b),
                      from="Lab",to="Apple RGB",clip=TRUE)

    tryCatch({
      cola <- rgb(b[1,][1], b[1,][2], b[1,][3])
      rect(xleft+width/2+spacing/8,ybottom,xleft+width,ybottom+width,
           col=cola,border = FALSE)
    }, warning = function(w) {
    }, error = function(e) {print(c(i,j,b))})
  }
}

```


Color card comparison



Mean of measured(right) vs Master(left) colors

Outliers

Which color deviates the most across all sheets from the master (highest delta E)? Bottom rectangle is what the color looks like in the master sheet. Top rectangle is what the average of the most deviated color looks like.

```
maxDeVal <- max(summarized$dE, na.rm = TRUE)
maxCol <- summarized[which.max(summarized$dE),]

start<-1
width<-30
spacing<-2
xspace<-cbind(1:width*2.5,1:width*2.5)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")

a <- convertColor(c(maxCol$L_master, maxCol$a_master, maxCol$b_master),
  from="Lab",to="sRGB",clip=NA)
b <- convertColor(c(maxCol$L_measured, maxCol$a_measured, maxCol$b_measured),
  from="Lab",to="sRGB",clip=NA)
```

```

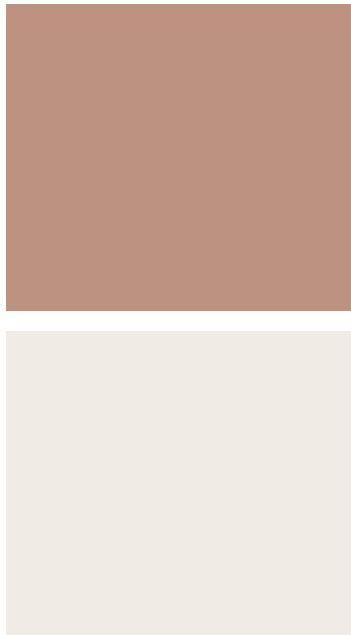
a_color <- rgb(a[1,][1], a[1,][2], a[1,][3])
b_color <- rgb(b[1,][1], b[1,][2], b[1,][3])

xleft<-0.7*(width+spacing)
ybottom<-0*(width+spacing)
rect(xleft,ybottom,xleft+width,ybottom+width,col=a_color,border = FALSE)

xleft<-0.7*(width+spacing)
ybottom<-1*(width+spacing)
rect(xleft,ybottom,xleft+width,ybottom+width,col=b_color,border = FALSE)
title(" Furthest outlying color cell comparison", sub = "Top: Mean of Measured, Bottom: Master")

```

Furthest outlying color cell comparison



Top: Mean of Measured, Bottom: Master

The delta of the above E colors is 25.2923476, which means (according to this chart) the colors are more similar than opposite. However, the difference is perceptible to the common human eye.

Which color deviates the least (lowest delta E)? Bottom rectangle is what the color looks like in the master sheet. Top rectangle is what the average of the most deviated color looks like.

```

minDeVal <- min(summarized$dE, na.rm = TRUE)
minCol <- summarized[which.min(summarized$dE),]

```

```

start<-1
width<-30
spacing<-2
xspace<-cbind(1:width*2.5,1:width*2.5)
plot(xspace,type="n",axes=FALSE,xlab = "",ylab = "")

a <- convertColor(c(minCol$L_master, minCol$a_master, minCol$b_master),
                  from="Lab",to="sRGB",clip=NA)
b <- convertColor(c(minCol$L_measured, minCol$a_measured, minCol$b_measured),
                  from="Lab",to="sRGB",clip=NA)
a_color <- rgb(a[1,][1], a[1,][2], a[1,][3])
b_color <- rgb(b[1,][1], b[1,][2], b[1,][3])

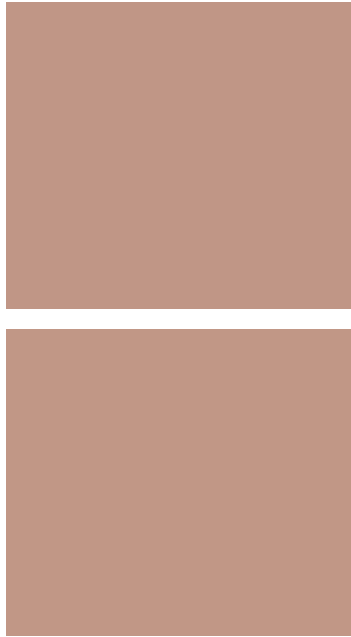
xleft<-0.7*(width+spacing)
ybottom<-0*(width+spacing)
rect(xleft,ybottom,xleft+width,ybottom+width,col=a_color,border = FALSE)

xleft<-0.7*(width+spacing)
ybottom<-1*(width+spacing)
rect(xleft,ybottom,xleft+width,ybottom+width,col=b_color,border = FALSE)

title("Color cell with the least difference", sub = "Top: Mean of Measured, Bottom: Master")

```

Color cell with the least difference



Top: Mean of Measured, Bottom: Master

HeatMap

The delta of the above E colors is 0.2361991, which means (according to this chart) that the difference between the colors is not perceptible to the common human eye.

Heat map of the summarized Lab values across all sheets against their master values:

```
x <- summarized[,c("L_measured", "a_measured", "b_measured",  
                  "L_master", "a_master", "b_master")]  
  
# Use 'scale' to normalize (right)  
heatmap(as.matrix(x), scale="column", col = terrain.colors(256), margins=c(8,5),  
        main = "Heatmap of Lab values (master & measured)")
```

Heatmap of Lab values (master & measured)

