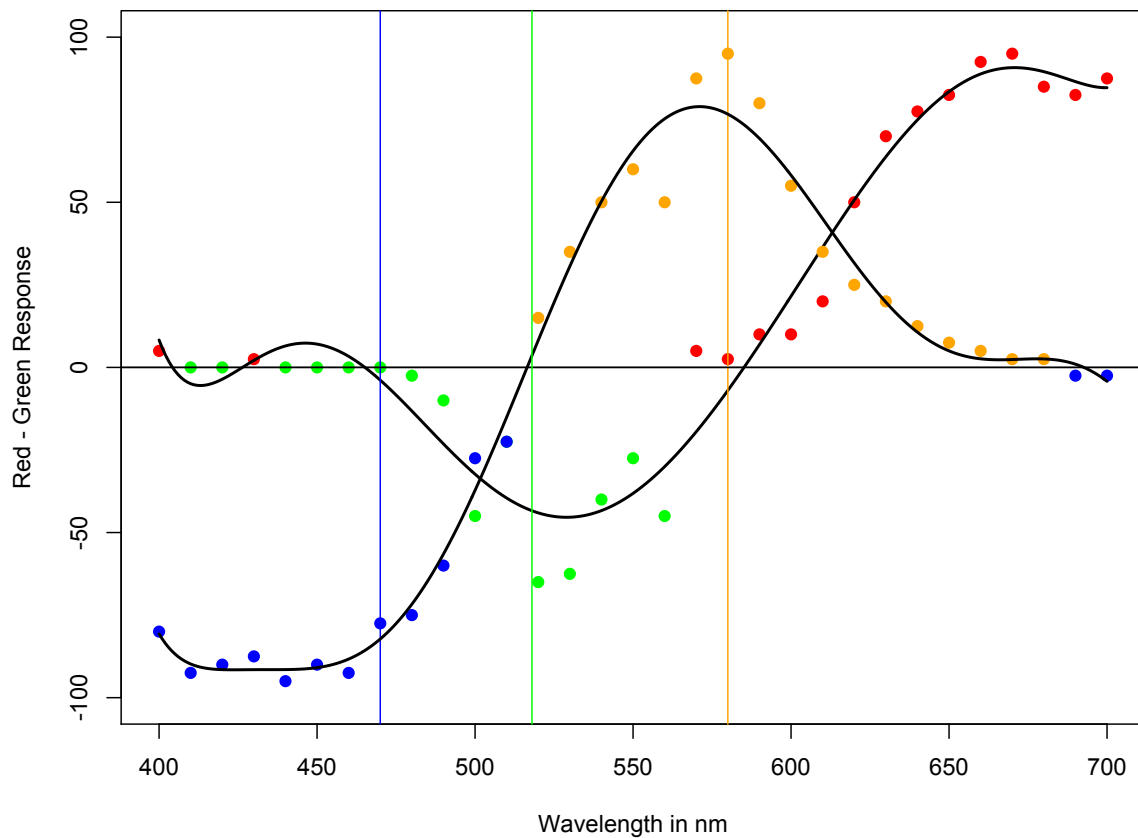# Psychology of Perception

Psychology 4165, Fall 2010

Laboratory 3: Hue-Naming Functions

Fig 3: Chromatic Mechanisms

Page intentionally blank

Lab 3: Hue Naming

Introduction

Color is a psychological experience composed of at least three psychological dimensions: **hue**, **saturation**, and **brightness**. Hue is the experience that we describe with color name labels such as red or blue. Saturation describes the intensity of the hue experience, ranging from hueless to deeply saturated. Pink, for example, is a desaturated red. Brightness is the dimension of experience that permits us to order colors from dark to light. Yellow, for example, usually appears brighter than navy blue. In this experiment you will explore the color experience of hue and gain some insight into the perceptual mechanisms that create it.

Physical light is composed of discrete packets of energy called quanta. A quantum of light has only a single physical property that can be described in one of three ways: By the energy in the quantum; by the frequency of the quantum; or by the wavelength of the quantum. Quanta having wavelengths in the range of approximately 400 to 700 nm (1 nanometer = 10^{-9} meter) are called photons because, under the right circumstances, they can lead to visual experiences. Light has no color; color is an experience created by the visual system in response to stimulation by light.

There are hundreds of different color names in English and in most languages. In spite of the plethora of color words, people with normal color vision can describe almost all colors as being composed of a percentage of red, yellow, green, and blue. For example, one might describe orange as being 60% red and 40% yellow or lime as 70% green and 30% yellow and so on. In this experiment you are going to view monochromatic lights (light composed of a single wavelength) and estimate the percentage of red, yellow, green, and blue making up the color experience you have of each one.

Methods

Procedure: Form groups of 2 or 3. Each will view monochromatic light projected on a white screen by a Bausch and Lomb monochrometer. Thirty-one separate wavelengths ranging from 400 nm to 700 nm in 10 nm steps will be viewed. These wavelengths should be presented in a random order. Record the size of the colored spot and your viewing distance from it. Viewing distance should remain constant throughout the experiment.

Color Naming Procedure: Present the 31 wavelengths one at a time in random order. On each trial, write down on your data sheet (see Appendix I) what percentage of the color experience evoked by the wavelength is red. Write down 0 percent if there is no red in your experience. This task may seem very strange at first. You might want to practice a bit before you start in earnest. After all 31 wavelengths have been judged for their redness, repeat the process for yellow, then for green and finally for blue. Be sure to judge all 31 wavelengths for a given color before moving on to the next color. Use a different random order of the wavelengths for each color judgment.

Pure Color Procedure: After you have collected the color naming data above, use the method of adjustment to determine the wavelength that gives you the

psychological pure colors of blue, green, and yellow. If you have a dichromatic color deficiency, find the wavelength that appears hueless or neutral white or gray. Record these wavelengths on your data sheet in Appendix II. Make at least 10 trials per color (i.e., blue, green, and yellow). Compute the mean wavelength for each color and enter them on the group data sheet. These means can be computed in R (see below).

Individual Data Analyses: Download the Lab3_Tools file. Open the Excel file "Hue_Name_Template.txt" in Excel and transfer your data from the table in Appendix 1 to the appropriate column. Note that the last three columns will each have only 10 numbers (the pure color wavelengths from Appendix II), not 31 like the other columns. Save the data in a **text** file with your own name. Be sure to save the file as a text file, not an Excel file.

The first step in the data analysis is to compute the opponent responses by subtracting the percent green from the percent red and by subtracting the percent blue from the percent yellow. Also compute the total percentage at each wavelength. If red, yellow, green, and blue are sufficient to describe all color experiences, these totals should be 100 percent.

The second step is to compute the mean wavelength giving pure blue, pure green and pure yellow and add them to the data frame. These steps may be accomplished by the following R code:

```
df <- cbind(df, red_green = df$red - df$green)
df <- cbind(df, yellow_blue = df$yellow - df$blue)
df <- cbind(df, total = df$red + df$yellow + df$green + df$blue)

# Compute the mean wavelengths of the pure colors

pure.blue <- mean(df$pure_blue, na.rm = TRUE)
pure.green <- mean(df$pure_green, na.rm = TRUE)
pure.yellow <- mean(df$pure_yellow, na.rm = TRUE)
```

Make two sets of graphs presenting your data: Wavelength should be plotted on the abscissa and percent on the ordinate. Plot your red and green data on one graph and your yellow and blue data on a second graph. The R code for producing these graphs is given in the appendix. These two graphs should look like the ones on the next page (Figure 1). The R-script defines four functions for drawing the four graphs: `plot1()`, `plot2()`, `plot3()`, and `plot4()`.

Now plot the two chromatic processes represented by the difference between red and green and between yellow and blue. Use an 8th order polynomial to fit a smooth function through your data. Plot the data and the smooth function. Add to this graph vertical lines to mark the position of the wavelength corresponding to pure blue, green, and yellow. Your graphs should resemble those on the next page (Figure 2). The R code to do the polynomial fit and to plot the data is given in the Appendix. You may want to plot the two opponent responses on the same graph in a Figure 3, as is shown on the front page of this document.

From your chromatic process curves in Figure 2, determine the wavelengths giving psychologically pure blue (where red and green are zero), green (where blue

and yellow are zero), and yellow (where red and green are zero). You may use the R functions defined in the script (`predict.rg(nm)` and `predict.yb(nm)`) to try different wavelength values to get close to zero. Or you could use `uniroot()` to find the zero point for you. If you use `uniroot()`, constrain the limits of nm values to search based on the graphs. Transfer the zero-crossing wavelengths for blue, green and yellow to Table III in the appendix. Transfer the mean pure wavelengths for blue, green and yellow to the Table IV in the appendix. How do these wavelengths compare with those measured directly? In your discussion relate your findings to the opponent-process theory of color vision. Finally, copy the results from Tables III and IV to the data sheet at the front of the class.
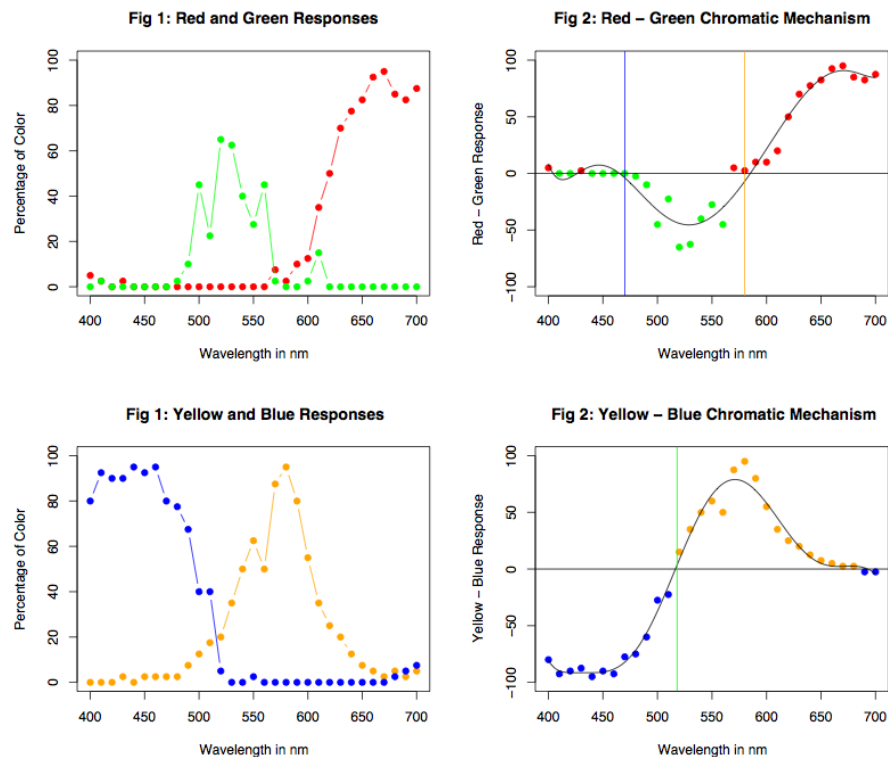
Compare your data (graph) with those of your group members. How are they alike and how do they differ? If you have someone in your group who has a color deficiency examine his (most likely it would be a male) data carefully and compare them with the members of your group who have normal trichromatic color vision.

Now prepare a fourth figure plotting the total percentage response against wavelength. Use a linear regression (`lm(total ~ wavelength)` in R) to compute the best-fitting line. Is the slope of this line zero? Are there wavelengths where having only four color names might not be sufficient to describe your experiences?

Group Data Analysis: After we collect the results of your individual analyses, we will prepare a data file containing these data from the class. The data file lab3_group_data_wide.txt contains the mean wavelengths corresponding to psychologically pure blue, pure green, and pure yellow for each observer in the

Lab 3: Hue Naming

class. Also are the neutral point wavelengths that should correspond to the same colors. Examine the data to see if there is much consensus among observers. You could plot the wavelengths for the three colors on a strip chart or make a box and whiskers plot to look at the spread of your data. You could also look at the means and standard deviations for each color. Finally, test the hypothesis that the wavelengths giving pure blue, pure green and pure yellow are the same (the null hypothesis). In addition test the null hypothesis that gender makes no difference on the wavelengths giving pure colors. The series of commands in R that create these graphs and compute the statistics are given at the end of this handout.

Laboratory Report

Your lab report should contain five parts: *Cover Sheet, Introduction, Methods, Results,* and *Discussion*. In the *Introduction* explain why you did the experiment. In the *Methods* section describe what you did. In the *Results* section present your findings, including graphs of your data. In the *Discussion* of your results, here are some important questions to answer. Is there a systematic relationship between wavelength of light and the percentages of red, yellow, green, and blue experience evoked by it? Are there any points in the wavelength spectrum that give rise to a unique hue? A unique hue would occur at a wavelength that gave 100% of one color name and 0% of the other three. Look at the relationship among the four curves. Do pairs of curves seem to have a special relationship with one another? Relate your findings to the trichromatic theory of color vision proposed by Helmholtz and to the opponent-process theory proposed by Hering. Examine the spread of the unique hue wavelength across the students in the class. Is there good agreement or not?

Your lab report should be brief and contain six sections: cover page, introduction, methods, results, discussion, and references. These sections should conform to the American Psychological Association (APA) style as described in Chapter 13 of the Martin book. The results section should contain the graphs plotting your data and the opponent-process transformation. The report is due at lab meeting (**12 and 14 September 2010**). Late labs will receive a grade of zero. All lab reports must be prepared with a word processor. It is worth 50 points.

Lab 3: Hue Naming

Appendix I: Percent Color Experience

	Wavelength	Red	Yellow	Green	Blue
1	400 nm				
2	410 nm				
3	420 nm				
4	430 nm				
5	440 nm				
6	450 nm				
7	460 nm				
8	470 nm				
9	480 nm				
10	490 nm				
11	500 nm				
12	510 nm				
13	520 nm				
14	530 nm				
15	540 nm				
16	550 nm				
17	560 nm				
18	570 nm				
19	580 nm				
20	590 nm				
21	600 nm				
22	610 nm				
23	620 nm				
24	630 nm				
25	640 nm				
26	650 nm				
27	660 nm				
28	670 nm				
29	680 nm				
30	690 nm				
31	700 nm				

Appendix II: Psychologically Pure Colors

Trial	Wavelength for psychologically pure color		
	Blue	Green	Yellow
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
Mean			

Appendix III: Opponent-Process Neutral Points from Curve Fitting

	Wavelength of Zero-Crossing Points		
	Blue	Green	Yellow
nm			

Appendix IV: Psychologically Pure Colors from Method of Adjustment

	Mean Wavelength for Pure Color		
	Blue	Green	Yellow
nm			

Psychology of Perception
Psychology 4165-100
Fall 2010

Lewis O. Harvey, Jr.–Instructor
Miranda M. Rieter–Assistant
09:00–9:50 MWF

Lab 3: Hue Naming

```r
# lab3_Graphs.R
# command file for analyzing individual hue naming data
# Lew Harvey
# 23 September 2010

# read file into data frame
df <- read.delim(file.choose(), header=TRUE)

# Compute opponent processs and add columns to data frame
df <- cbind(df, red_green = df$red - df$green)
df <- cbind(df, yellow_blue = df$yellow - df$blue)
df <- cbind(df, total = df$red + df$yellow + df$green + df$blue)

# Compute the mean wavelengths of the pure colors
pure.blue   <- mean(df$pure_blue,   na.rm=TRUE)
pure.green  <- mean(df$pure_green,  na.rm=TRUE)
pure.yellow <- mean(df$pure_yellow, na.rm=TRUE)

# fit a polynomial to the opponent process data and
rg.poly <- lm(red_green ~ poly(wavelength, degree = 8), data = df)
yb.poly <- lm(yellow_blue ~ poly(wavelength, degree = 8), data = df)

# define the functions for finding the roots
# of the opponent process equations
predict.rg <- function(nm){predict(rg.poly, data.frame(wavelength = nm))}
predict.yb <- function(nm){predict(yb.poly, data.frame(wavelength = nm))}

# ------------------------------------------------------
# plot the individual color percentages (Figure 1)
# ------------------------------------------------------
plot1 <- function() {
    par(mfrow=c(2,1))    # two graphs per figure
    # red and green percentages
    plot(red ~ wavelength, data = df,
        col = "red", type = "o", pch = 19,
        xlim = c(400, 700),
        ylim = c(0, 100),
        xlab = "Wavelength in nm",
        ylab = "Percentage of Color",
        main = "Fig 1a: Red and Green Responses")
    points(green ~ wavelength, data = df,
        col = "green", type = "o", pch = 19,
        xlim = c(400, 700),
        ylim = c(-100, 100))

    # yellow and blue percentages
    plot(yellow ~ wavelength, data = df,
        col = "orange", type = "o", pch = 19,
        xlim = c(400, 700),
        ylim = c(0, 100),
        xlab = "Wavelength in nm",
        ylab = "Percentage of Color",
        main = "Fig 1b: Yellow and Blue Responses")
    points(blue ~ wavelength, data = df,
        col = "blue", type = "o", pch = 19,
        xlim = c(400, 700),
        ylim = c(-100, 100))
    par(mfrow=c(1,1))
}

# ------------------------------------------------------
```

Lab 3: Hue Naming

```r
# plot the opponent processes (Figure 2)
# --------------------------------------------------------
plot2 <- function() {
    par(mfrow=c(2,1))     # two graphs per figure
    # plot the red-green process
    df.red <- subset(df, red_green > 0)
    df.green <- subset(df, red_green <= 0)
    plot(red_green ~ wavelength, data = df.red,
        type = "p", col = "red", pch = 19,
        xlim = c(400, 700),
        ylim = c(-100, 100),
        xlab = "Wavelength in nm",
        ylab = "Red - Green Response",
        main = "Fig 2a: Red - Green Chromatic Mechanism")
    points(red_green ~ wavelength, data = df.green,
        type = "p", col = "green", pch = 19)
    abline(h = 0)

    # plot the best-fitting line from the polynomial
    x <- 400:700
    lines(x, predict(rg.poly, data.frame(wavelength = x)), lwd = 2)
    abline(v = pure.blue, col = "blue")
    abline(v = pure.yellow, col = "orange")

    # plot the yellow-blue opponent process
    df.yellow <- subset(df, yellow_blue > 0)
    df.blue <- subset(df, yellow_blue <= 0)
    plot(yellow_blue ~ wavelength, data = df.yellow,
        type = "p", col = "orange", pch = 19,
        xlim = c(400, 700),
        ylim = c(-100, 100),
        xlab = "Wavelength in nm",
        ylab = "Yellow - Blue Response",
        main = "Fig 2b: Yellow - Blue Chromatic Mechanism")
    points(yellow_blue ~ wavelength, data = df.blue,
        type = "p", col = "blue", pch = 19)
    abline(h = 0)

    # plot the best-fitting line from the polynomial
    x <- 400:700
    lines(x, predict(yb.poly, data.frame(wavelength = x)), lwd = 2)
    abline(v = pure.green, col = "green")
    par(mfrow=c(1,1))
}

# --------------------------------------------------------
# plot the opponent processes on one figure (Figure 3)
# --------------------------------------------------------
plot3 <- function() {
    par(mfrow=c(1,1))     # one graph per figure
    # plot the red-green process
    df.red <- subset(df, red_green > 0)
    df.green <- subset(df, red_green <= 0)
    plot(red_green ~ wavelength, data = df.red,
        type = "p", col = "red", pch = 19,
        xlim = c(400, 700),
        ylim = c(-100, 100),
        xlab = "Wavelength in nm",
        ylab = "Red - Green Response",
        main = "Fig 3: Chromatic Mechanisms")
    points(red_green ~ wavelength, data = df.green,
        type = "p", col = "green", pch = 19)
```

Lab 3: Hue Naming

```r
    abline(h = 0)

    # plot the best-fitting line from the polynomial
    x <- 400:700
    lines(x, predict(rg.poly, data.frame(wavelength = x)), lwd = 2)
    abline(v = pure.blue, col = "blue")
    abline(v = pure.yellow, col = "orange")

    # plot the yellow-blue opponent process
    df.yellow <- subset(df, yellow_blue > 0)
    df.blue <- subset(df, yellow_blue <= 0)
    points(yellow_blue ~ wavelength, data = df.yellow,
        type = "p", col = "orange", pch = 19)
    points(yellow_blue ~ wavelength, data = df.blue,
        type = "p", col = "blue", pch = 19)
    abline(h = 0)

    # plot the best-fitting line from the polynomial
    x <- 400:700
    lines(x, predict(yb.poly, data.frame(wavelength = x)), lwd = 2)
    abline(v = pure.green, col = "green")
    par(mfrow=c(1,1))
}


# -------------------------------------------------------
# plot the total percentages (Figure 4)
# -------------------------------------------------------
plot4 <- function() {
    par(mfrow=c(2,1))       # two graphs per figure
    plot(total ~ wavelength, data = df, pch = 19,
        ylim = c(40, 140),
        xlab = "Wavelength in nm",
        ylab = "Total Percentage",
        main = "Fig 4a: Sum of Red, Yellow, Green, and Blue Percentages")
    abline(h = 100)
    # compute a linear regression on total
    tot.lm <- lm(total ~ wavelength, data = df)
    lines(tot.lm$fitted ~ df$wavelength)

    # plot the residual from the total ~ wavelength regression
    plot(tot.lm$residual ~ df$wavelength, pch = 19,
        ylim = c(-50, 50),
        xlab = "Wavelength in nm",
        ylab = "Residuals",
        main = "Fig 4b: Residuals from Linear Regression")
    abline(h=0)
    par(mfrow=c(1,1))
}
```

Lab 3: Hue Naming

```r
# lab3_group_analysis.R
# Analysis of Group Hue Naming Data
# Lew Harvey
# 23 September 2010

# load the nlme library to use lme()
library(nlme)

# read in group data in wide format from Excel text file
fn <- file("lab3_group_data_wide.txt")
df.wide <- read.delim(fn, header=TRUE)

# ****************************************************
# reshape the file into long format
# Note: reshape() is a bit of a pain to use but what it does
# is take data that are in a so-called wide format
# (one row per subject, separate columns per dependent variable)
# and put it into the long formate needed by many R analysis routines
# (multiple rows per subject, single column per dependent variable)
df.long <- reshape(df.wide,
    varying = list(c("zeroB", "zeroG", "zeroY"), c("pureB", "pureG", "pureY")),
    idvar = "subj",
    times = c("blue", "green", "yellow"),
    v.names = c("zero_wavelength", "pure_wavelength"),
    timevar = "color",
    direction = "long")
rownames(df.long) <- 1:dim(df.long)[1]
df.long$color <- factor(df.long$color)
# ****************************************************

# use shorter name for the data frame
df <- df.long

# -------------------------------------------------------
# Make Figure 1
# -------------------------------------------------------
plot1 <- function() {
    par(mfcol = c(1,2))
    # Make a strip chart of the zero-crossing data
    stripchart(zero_wavelength ~ color, data = df, method = "jitter", jitter =
0.03,
        ylab = "zero-crossing wavelength (nm)",
        xlab = "Color", ylim = c(400, 650), vertical = TRUE)

    # Make a box plot of the zero-crossing data:
    boxplot(zero_wavelength ~ color, data = df, ylim = c(400, 650),
        ylab = "zero-crossing wavelength (nm)",
        xlab = "Color", ylim = c(400, 650))
    par(mfcol = c(1,1))
}

# -------------------------------------------------------
# Make Figure 2
# -------------------------------------------------------
plot2 <- function() {
    par(mfcol = c(1,2))
    # Make a strip chart of the pure-color data
    stripchart(pure_wavelength ~ color, data = df, method = "jitter", jitter =
0.03,
        ylab = "pure color wavelength (nm)",
        xlab = "Color", ylim = c(400, 650), vertical = TRUE)
```

Lab 3: Hue Naming

```r
    # Make a box plot of the pure-color data:
    boxplot(pure_wavelength ~ color, data = df, ylim = c(400, 650),
        ylab = "pure color wavelength (nm)",
        xlab = "Color", ylim = c(400, 650))
        par(mfcol = c(1,1))
}


# ----------------------------------------------------
# Make Figure 3
# plot the wavelengths for the pure colors from settings
# against the wavelengths derived from zero crossings
# ----------------------------------------------------
plot3 <- function() {
    par(mfcol = c(1,1))
    par(pty = "s") # make graph square
    plot(pure_wavelength ~ zero_wavelength, data = df, type="n",
        xlim = c(400, 650),
        ylim = c(400, 650),
        xlab = "zero-crossing wavelengths",
        ylab = "pure-color wavelength",
        main = "Fig 3: Hue-Naming Laboratory Group Data")
    abline(0,1)
    points(pure_wavelength ~ zero_wavelength, data=subset(df,df$color=="blue"),
col="blue", pch=19)
    points(pure_wavelength ~ zero_wavelength, data=subset(df,df$color=="green"),
col="green", pch=19)
    points(pure_wavelength ~ zero_wavelength, data=subset(df,df$color=="yellow"),
col="orange", pch=19)
    par(pty = "m") # restore normal graph aspect ratio
}

# Testing the hypothesis that color is independent of wavelength and gender
# Compute the mixed-effects (within and between factors ANOVA
a <- lme(pure_wavelength ~ color * gender, random = ~ 1 | subj, data = df)

# print a summary of the analysis of variance
print(anova(a))

# Print a table of means for the three colors
print("color means")
print(tapply(df$pure_wavelength, df$color, mean, na.rm=TRUE))
print(tapply(df$zero_wavelength, df$color, mean, na.rm=TRUE))

# Print a table of means for the genders
print("gender means")
print(tapply(df$pure_wavelength, df$gender, mean, na.rm=TRUE))
print(tapply(df$zero_wavelength, df$gender, mean, na.rm=TRUE))

# Print a table of means for gender and colors
print("gender and color means")
print(tapply(df$pure_wavelength, INDEX=list(df$gender, df$color), mean,
na.rm=TRUE))
print(tapply(df$zero_wavelength, INDEX=list(df$gender, df$color), mean,
na.rm=TRUE))

# Print correlation between the two ways of measuring hue
print("correlation between pure color wavelengths and zero-crossing wavelengths")
print(cor(df$pure_wavelength, df$zero_wavelength))
```