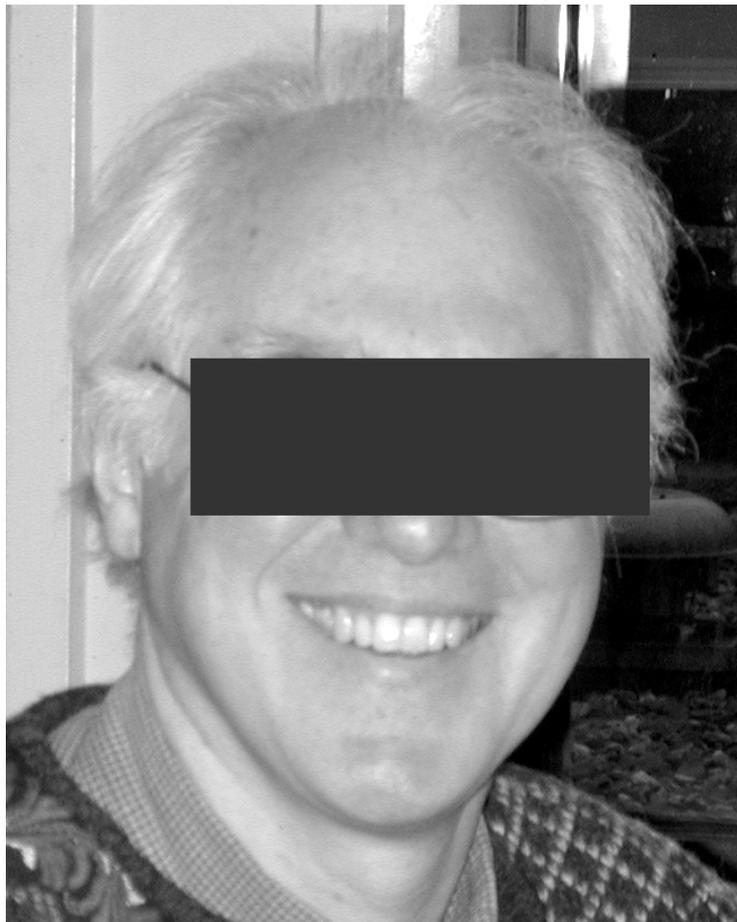


Psychology of Perception
Psychology 4165, Summer 2010
Laboratory 2

Face Recognition: Are Eyes Important?



Psychology of Perception
Psychology 4165
Section 100
Summer 2010

Lewis O. Harvey, Jr.–Instructor
Miranda M. Rieter–Assistant
Clare E. Sims –Assistant
MUEN D156, 09:15–10:50 M–F

Intentionally Blank

Introduction

Recognition of human faces is a remarkably good. Even after 50 years have passed, people are able to choose which of two photographs is a high school classmate with an accuracy of almost 90 percent correct (Bahrick, Bahrick, & Wittlinger, 1975). The eyes and surrounding areas of the face are thought to be very important for face recognition (Bruce, 1988; Vinette, Gosselin, & Schyns, 2004). In this lab you test the hypothesis that obscuring the eyes makes it more difficult to later recognize a face.

There are two principle theories that describe how observers detect weak stimuli: The High Threshold Model (HTM) and Signal Detection Theory (SDT). In this lab you will also test the predictions of these two models for face recognition.

First you will be shown a series of 64 photographs of faces to remember. Some will have their eyes covered and some will not. Then you will be tested with another series of 128 faces, half of which you have seen before and half of which are new. Your task is to decide on each trial whether or not the face had been previously seen. You will use a six-point confidence rating scale. A six-point rating scale corresponds to having five different decision criteria. From your data you will calculate five hit rate false alarm rate (HR-FAR) pairs (one pair for each of these response criteria) for the two types of faces. The two detection models will be compared to see how well each model can predict the observed data.

The objectives of this laboratory exercise are:

1. To test whether or not seeing the eyes enhance your ability to later recognize faces.
2. To test which model, the High Threshold Model or the Signal Detection Theory Model, better predicts your observed data in the face recognition experiment;

Experimental Procedure

You will first be shown a series of 64 target faces. Thirty-two of the faces have the eyes covered over and the other 32 are normal faces with the eyes visible. Study each face carefully during the 2 sec exposure time. Then in the test phase of the experiment, you will be shown a series of 128 test faces. Half of them will be target faces and half of them will be new faces that you have not previously seen. Using the six-point rating scale below rate each face on your confidence that it is a new face or a target face:

- 1 = certain the face was not seen before
- 2 = perhaps the face was not seen before
- 3 = guessing that the face was not seen before
- 4 = guessing that the face was seen before
- 5 = perhaps the face was seen before
- 6 = certain the face was seen before

The experiment will be run on your computer using the program PsyScope.

Download the Lab2_Tools folder from the course web page:

[http://psych.colorado.edu/~lharvey/P4165/P4165_2010_Spring/p4165\(2010_Spring\).html](http://psych.colorado.edu/~lharvey/P4165/P4165_2010_Spring/p4165(2010_Spring).html)

Open the folder and double click on the PsyScope script in the experiment folder. Select Run from the Run menu. Give your data file a short name. Don't worry about the difficulty of the task. Making these decisions sometimes is frustrating. Just relax and respond with a number from 1 to 6. Be sure, though, to use all of the response categories. Your data will be recorded in the data file that you named.

Data Analysis

1. Data Tabulation And Transformation

Your data analysis begins with a series of transformations of the raw data through the following sequence:

1. response scale frequencies;
2. response scale probabilities;
3. cumulative probabilities (hit rates and false alarm rates);
4. z-score transformation of the cumulative probabilities.

First: Count up the number of times you used each of the rating categories for the new face trials, for the old faces no-eyes trials, and for the old faces with eyes trials. These are the response scale frequencies. Record them in the column labeled “Rating Frequency” in Appendix II. First, modify your data file so that it can be read by R and the frequencies tabulated. Open your data file by double-clicking on it. Locate the header line (line 16) and select the word “Condition.” Change “Condition” to three words with a tab or space between them: “Gender,” “Target,” and “Eyes.” These words will be labels for the corresponding data columns when the file is read by R. After you run the R script (lab2_RawData.R) the tabulated frequencies will be found in R objects **tab.rec**.

Second: Use the computer program *RscorePlus* to compute the remaining data transformations. First double click on the data template file (rscoreDataTemplate.txt) so that you can edit it in Microsoft Word. Enter the frequency of each confidence rating for the three types of stimulus trials: new faces, old faces that were previously seen without eyes, old faces that were previously seen with eyes. Insert your name in the title line. Make sure that you save this file as text only. Quit Word. Now double click on *RscorePlus*, and type in your input file name at the prompt. You should analyze your data twice: once for the Gaussian signal detection model and once for the high threshold model. For each analysis, the program will produce three output files having the same name as your input file, but with different extensions:

1. myfile_gauss.txt a text file with the printed output of each analysis
2. myfile_gauss.grf a text file to be imported into R for making graphs
3. myfile_gauss.alt a results file for statistical analysis (you will not use this file)

Print the *_*gauss*.txt and the *_*htm*.txt files by double clicking on each file to open it in Microsoft Word. Before printing, you should select all the text and set the font size to 8 and the font type to Courier to make the printed results easier to read. Print the *gauss* and the *htm* file.

2. ROC Analysis

The high threshold model of detection predicts that the ROC will be a straight line when plotted in probability coordinates. The signal detection theory model predicts that the ROC will be a straight line when plotted in z-score (i.e., *qnorm()*) coordinates. The first step, therefore, is to plot your HR, FAR pairs on two types of graphs: one with linear probability coordinates and one with z-score coordinates. You may use the commands in file lab2_ROC_Graphs.R.

RscorePlus computes the parameters of the best-fitting detection model using a maximum-likelihood technique. The goodness-of-fit of the model is computed using the

chi-square statistic, χ^2 . These results are given in your printed output. Judging from the chi-square and from visual inspection of your graphs, which model, the high-threshold or the signal detection, gives a better description of your data? If you can reject one of the models, then you should not use the sensitivity and response bias measures from that model.

3. Signal Detection Model

RscorePlus computes the mean and standard deviation of the best-fitting model from the data. It also computes the position of the five decision criteria. These data are on the printed output. You can draw the Gaussian SDT model, with criteria, using the R script in the file `lab2_SDT_Model_Graph.R`. When you execute the script, it will ask you for the input file. Choose the `*Gaussian_grf.txt` file. You should edit the script to put in your own name on the graphs. You may have to adjust the position of the text labels. These labels are positioned at the bottom of the script.

4. Goodness of Fit

The ultimate test of each model is by how well it predicts the observed data. Since the parameters of each model were computed using a maximum-likelihood regression, the appropriate test is to calculate how well each model predicts the six response frequencies under the three stimulus conditions. The goodness of fit of these predictions can be formally computed using the chi square (χ^2) test. The value of χ^2 for each model is given on your printed output pages. Note that the degrees of freedom for the High Threshold Model are different than for the Signal Detection Model because the High Threshold Model has fewer free parameters to be computed from the data. To compare the fit of models having different numbers of free parameters we use a modified version of χ^2 that penalizes models with more free parameters. This modified χ^2 is called the Akaike Information Criterion (AIC) after the scientist who developed it (Akaike, 1974). Which model fits your observed data better (i.e., has the lower AIC)? Can you reject the hypothesis that each model is a good predictor of the data?

5. Group Data

When you have finished your *RscorePlus* analysis, enter the appropriate results from the printed output on the summary sheet in Appendix IV. Transfer these results to the group sheet that we will have for you so that we can prepare a group data for further analysis. You will test two null hypotheses:

1. H_0 : The signal detection model is just as good as the high threshold model;
2. H_0 : that seeing the eyes does not improve face recognition.

For the first hypothesis, compute a repeated measures analysis of variance on AIC, the modified *chi-square*, the badness-of-fit measure we are using (the higher the AIC, the worse the fit). For the second hypotheses, compute a repeated measures analysis of variance on d_a , the index of sensitivity in the signal detection model. The script file `lab2_group_analysis.R` contains the commands for carrying out these statistical tests with R.

6. Further reading

There is a large literature on signal detection theory. Although no additional reading is needed for this assignment, here are some references that you might wish to read if you want to enhance your understanding in this area. There are several books that are worthwhile (Egan, 1975; Green & Swets, 1966/1974; Macmillan & Creelman, 2005; McNicol, 1972; Swets & Pickett, 1982; Wickens, 2002). Here are some journal articles that will introduce you to the research literature (Harvey, 1992; Krantz, 1969; Simpson & Fitter, 1973; Swets, 1961, 1986a, 1986b; Swets, Tanner, & Birdsall, 1961).

Lab Report

Your lab report should contain five parts: Cover Page, 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. Your conclusions should be based on your statistical analyses, not your unsupported speculation. In the discussion you can let your creativity run wild. Give the reader your interpretation of the results. Discuss any implications and leads for further research. Laboratory reports must be typed, double-spaced on 8.5 x 11 paper with at least 1 inch margins. Conciseness and clarity are extremely important characteristics of good scientific writing. Strive for them. Worth 40 points. **Due in lab on 14 and 15 June 2010.**

References

- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *AC-19*(6), 716–723.
- Bahrick, H. P., Bahrick, P. O., & Wittlinger, R. P. (1975). Fifty years of memory for names and faces: A cross-sectional approach. *Journal of Experimental Psychology: General*, *104*(1), 54–75.
- Bruce, V. (1988). *Recognising faces*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Egan, J. P. (1975). *Signal Detection Theory and ROC Analysis*. New York: Academic Press.
- Green, D. M., & Swets, J. A. (1966/1974). *Signal detection theory and psychophysics* (A reprint, with corrections of the original 1966 ed.). Huntington, NY: Robert E. Krieger Publishing Co.
- Harvey, L. O., Jr. (1992). The critical operating characteristic and the evaluation of expert judgment. *Organizational Behavior & Human Decision Processes*, *53*(2), 229–251.
- Krantz, D. H. (1969). Threshold theories of signal detection. *Psychological Review*, *76*(3), 308–324.
- Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide* (2nd ed.). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- McNicol, D. (1972). *A primer of signal detection theory*. London: George Allen & Unwin.
- Simpson, A. J., & Fitter, M. J. (1973). What is the best index of detectability? *Psychological Bulletin*, *80*(6), 481–488.
- Swets, J. A. (1961). Is there a sensory threshold? *Science*, *134*(3473), 168–177.

- Swets, J. A. (1986a). Form of empirical ROCs in discrimination and diagnostic tasks: Implications for theory and measurement of performance. *Psychological Bulletin*, 99(2), 181–198.
- Swets, J. A. (1986b). Indices of discrimination or diagnostic accuracy: Their ROCs and implied models. *Psychological Bulletin*, 99(1), 100–117.
- Swets, J. A., & Pickett, R. M. (1982). *Evaluation of diagnostic systems: methods from signal detection theory*. New York: Academic Press.
- Swets, J. A., Tanner, W. P., Jr., & Birdsall, T. G. (1961). Decision processes in perception. *Psychological Review*, 68(5), 301–340.
- Vinette, C., Gosselin, F., & Schyns, P. G. (2004). Spatio-temporal dynamics of face recognition in a flash: It's in the eyes. *Cognitive Science*, 28(2), 289–301.
- Wickens, T. D. (2002). *Elementary signal detection theory*. New York: Oxford University Press.

**Appendix I: Summary of Raw Data from R Analysis of Data File
 Rating Frequencies for the RscorePlus input file**

	[1]	[2]	[3]	[4]	[5]	[6]	Total
New Faces							64
Old Faces no eyes							32
Old Faces with eyes							32

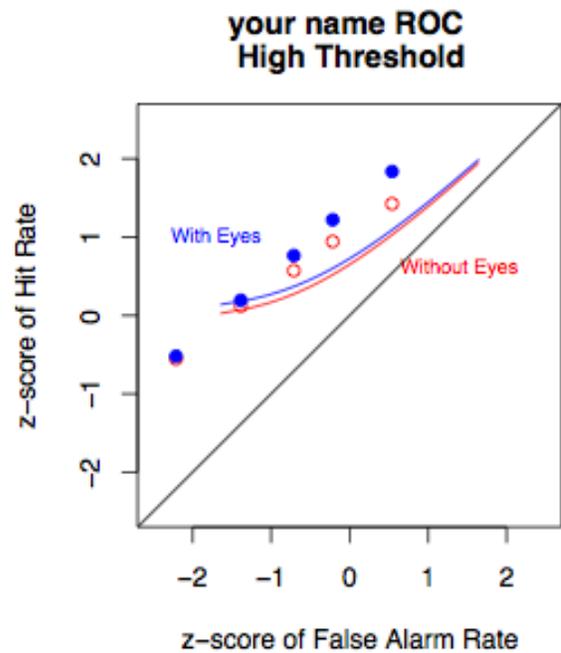
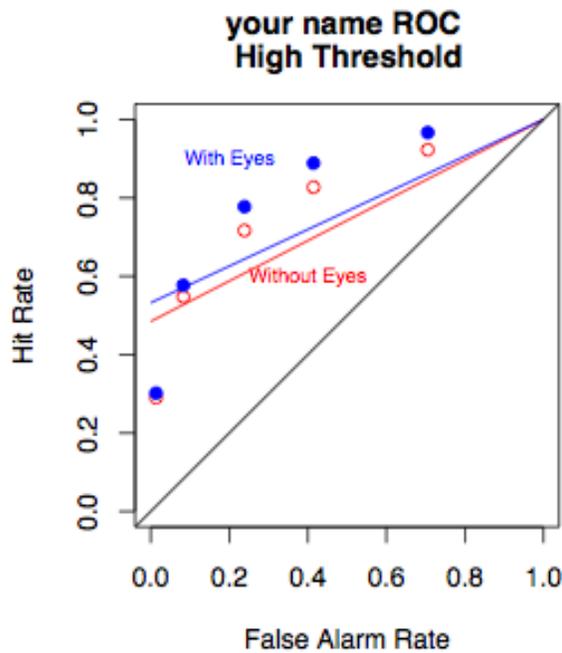
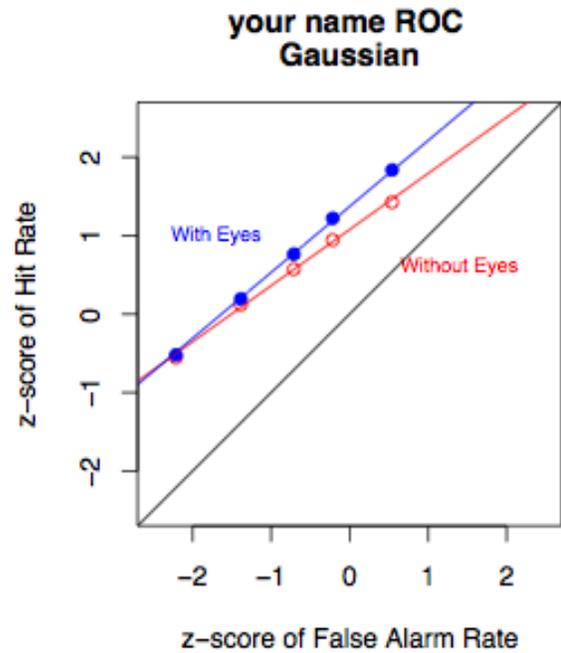
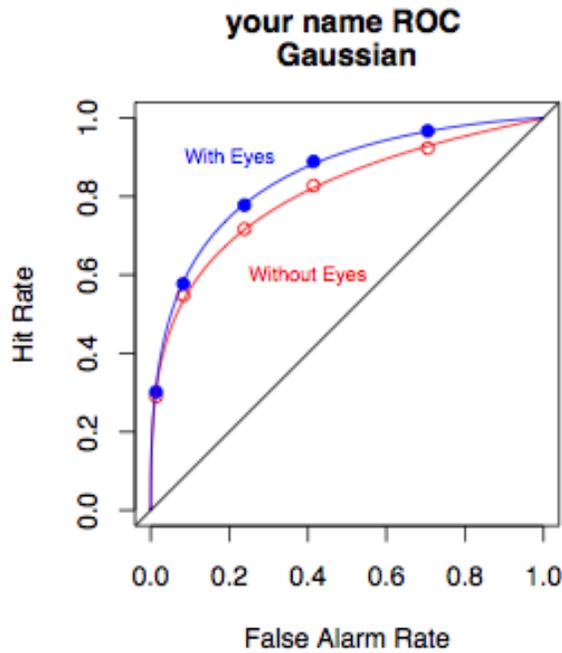
**Appendix II: Summary of Results of Model-Fitting with RscorePlus
 Signal Detection Theory Results**

Sensory Process Sensitivity (d_a)		Goodness-of-Fit Measures		
Without eyes	With eyes	Chi-Square	Probability	AIC

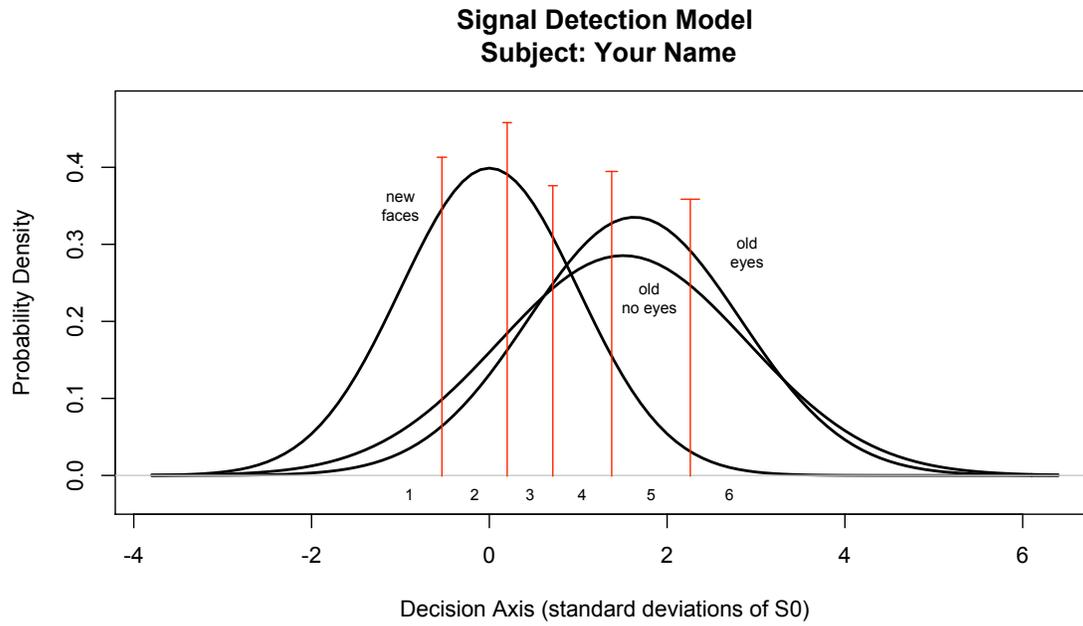
High Threshold Model Results

Sensory Process Sensitivity (p)		Goodness-of-Fit Measures		
Without eyes	With eyes	Chi-Square	Probability	AIC

Appendix III: Recommended format for the ROC graphs



Appendix IV: Recommended format for the SDT model graph



Using R for Lab 2 Analysis

Analysis of raw PsyScope data

```
# lab2_RawData.R
# 4 June 2010
# command file for reading in the PsyScope data file to tabulate
# the response frequencies for use in RscorePlus.
# The PsyScope file should have been edited to replace the column
# header "Condition" with "Gender Target Eyes"

fn <- file.choose()
df <- read.table(fn, header = TRUE, skip = 15, nrows = 64+128)

# make data frame with just the rating trials data
df.rec <- df[ 65:192, ]

# remove the [N/A] level from the key column
levels(df.rec$keys) <- c("[1]", "[2]", "[3]", "[4]", "[5]", "[6]", NA)

# add new column called "Signal" for the SDT analysis
df.rec <- cbind(df.rec, Signal = df.rec$Eyes)

# add a new level to the factor "Signal" and call it "New"
df.rec[, "Signal"] <- factor(df.rec[, "Signal"], levels=c("New",
"NoEyes", "Eyes"))

# set all the new face signal trials to "New"
df.rec[df.rec$Target=="new", "Signal"] <- "New"

# compute frequency contingency table and print it
tab.rec <- with(df.rec, table(Signal, keys))
print(tab.rec)
```

Drawing ROC Graphs from RscorePlus Output

```
# lab2_ROC_Graphs.R
# 4 June 2010
# command file for making ROC Graphs from RscorePlus
# Two graphs are produced from the *_grf file:
#   probability ROC
#   z-score probability ROC

# read the RscorePlus graphic file into a data frame
fn <- file.choose() # choose a file name
df <- read.delim(fn, header=TRUE) # read data from file

# Edit below to put in your own name
mn = "Your Name"
md = paste(df$Model[1], "Model") # name of model

# plot ROC curve for with without eyes in red and with eyes in blue
par(mfrow = c(1,2)) # two graphs per figure
par(pty = "s") # make plots square

# probability plot
plot(Obs_pYs1 ~ Obs_pYs0,
     data = df,
     pch = 21, col = "red",
     xlim = c(0, 1),
     ylim = c(0, 1),
     xlab = "False Alarm Rate",
     ylab = "Hit Rate",
     main = paste(mn, md, "Probability ROC", sep="\n"))
points(Obs_pYs2 ~ Obs_pYs0, data=df, pch=19, col="blue")
lines(Pred_pYs1 ~ Pred_pYs0, data=df, col="red", lwd = 2)
lines(Pred_pYs2 ~ Pred_pYs0, data=df, col="blue", lwd = 2)
text(.4, .60, "Without Eyes", col="red", cex=0.75)
text(.2, .90, "With Eyes", col="blue", cex=0.75)
abline(0,1)

# z-score plot
plot(Obs_zYs1 ~ Obs_zYs0, data=df, pch=21, col="red",
     xlim=c(-2.5, 2.5),
     ylim=c(-2.5, 2.5),
     xlab="z-score of False Alarm Rate",
     ylab="z-score of Hit Rate",
     main = paste(mn, md, "Z-Score ROC", sep="\n"))
points(Obs_zYs2 ~ Obs_zYs0, data=df, pch=19, col="blue")
lines(Pred_zYs1 ~ Pred_zYs0, data=df, col="red", lwd=2)
lines(Pred_zYs2 ~ Pred_zYs0, data=df, col="blue", lwd=2)
text(1.4, .6, "Without Eyes", col="red", cex=0.75)
text(-1.7, 1, "With Eyes", col="blue", cex=0.75)
abline(0,1)

par(pty = "m")
par(mfrow = c(1,1))
```

Drawing SDT Model from RscorePlus Output

```
# lab2_SDT_Model_Graph.R
# 4 June 2010
# script for drawing a signal detection model
# with decision criteria

# read the RscorePlus graphic file into a data frame
fn <- file.choose()           # choose file
df <- read.delim(fn, header=TRUE) # read data from file

# Edit below to put in your own name
mn = "Your Name"
md = paste(df$Model[1], "Detection Model") # name of model

# plot distribution for with without eyes in red
# and with eyes in blue

# #####
# Figure out how many criteria and how many signal conditions
nsig <- df$nsig[1] # number of signals
nrat <- df$nrat[1] # number of ratings
ncrt <- nrat - 1   # number of criteria

# Plot the signal detection model
# find the maximum density value for setting y-axis
# find the s0 probability density column
ind0 <- match("Pred_Density0", names(df))
ind1 <- ind0 + nsig - 1
max.density <- 1.2 * max(df[, ind0:ind1], na.rm = TRUE)

par(pty = "m")
# plot s0 probability density function
plot(df[, ind0] ~ Xc, data = df, type = "l",
      lwd = 2, col = "black",
      xlim = c(min(df$Xc, na.rm = TRUE), max(df$Xc, na.rm = TRUE)),
      ylim = c(-0.03, max.density),
      xlab = "Decision Axis (standard deviations of S0)",
      ylab = "Probability Density",
      main = paste(mn, md, sep = "\n"))

# plot the remaining probability density functions (s1...)
colors <- rainbow(nsig-1)
colors <- c("red", "blue")
for (i in 1:(nsig-1)){
  lines(df[, ind0+i] ~ Xc, data = df, type = "l",
        lwd = 2, scol = colors[i])}
abline(h = 0, col = "gray", lty = "solid")

# extract the criterion coordinates
crit <- subset(df, Criteria != is.na(Criteria))[c("Xc", "Xc_se", "Criteria")]
xc <- df$Xc[1:ncrt]
```

```
# plot the decision criterion markers
for (i in 1:ncrt){
  segments(crit$Xc[(2*i)-1], crit$Criteria[(2*i)-1], crit$Xc[2*i],
  crit$Criteria[2*i], col="red")}
# plot the standard error of each criterion marker
for (i in 1:ncrt){
  segments(crit$Xc[2*i]-crit$Xc_se[2*i], crit$Criteria[2*i],
  crit$Xc[2*i]+crit$Xc_se[2*i], crit$Criteria[2*i], col = "red")}

# label the response categories
xc.lab <- rep(0, nrat)
xc.lab[2:ncrt] <- xc[1:ncrt-1] + diff(xc)/2
xc.lab[1] <- xc[1] - (diff(xc)/2)[1]
xc.lab[nrat] <- xc[ncrt] + (diff(xc)/2)[ncrt-1]
text(xc.lab, -0.025, labels = 1:nrat, cex = 0.7)

# label the distributions
text(-1, .35, "new\nfaces", cex=0.7)
text(1.8, .23, "old\nno eyes", cex=0.7)
text(2.9, .29, "old\neyes", cex=0.7)
```

Computing ANOVAs on Group Data using R

```
# lab2_group_analysis.R
# PSYCH 4165 Lab 2
# Command file for using lme() to compute a
# repeated measures analysis of variance in R.
# Lew Harvey
# 4 June 2010

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

# Read in group data in wide format from Excel text file:
fn <- file("lab2_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("da_no_eyes", "da_with_eyes"), c("htm_no_eyes",
"htm_with_eyes"), c("sdt_AIC", "htm_AIC")),
  v.names = c("da", "htm_p", "aic"),
  timevar = c("eyes", "mod"),
  idvar = "subj",
  times = c("noEyes", "withEyes"),
  direction = "long")

rownames(df.long) <- 1:dim(df.long)[1]
df.long[df.long$mod == "noEyes", "mod"] <- "sdt"
df.long[df.long$mod == "withEyes", "mod"] <- "htm"
df.long$eyes <- factor(df.long$eyes)
df.long$mod <- factor(df.long$mod)
df <- df.long
# *****
# make functions to draw strip and box plots
# plot1() plots AIC fit against the two models
# plot2() plots memory sensitivity for eyes/no eyes
# *****

# Write out a summary of the variables in the data frame:
print(summary(df))

# Test the hypothesis that SDT and HTM are
# equally good at describing the data
plot1 <- function() {
  par(mfcol=c(2,1))  # two plots per figure
```

```
# Make a strip chart of the data
with(df, stripchart(aic ~ mod,
  method = "jitter", jitter = 0.03,
  xlab = "Goodness-of-Fit (AIC)",
  ylab = "Detection Model",
  main = "Lab 2 Group Data"))

# Make a box plot of the data:
boxplot(aic ~ mod, data = df, horizontal = TRUE,
  ylab = "Detection Model",
  xlab = "Badness-of-Fit (AIC)")
par(mfcol=c(1,1))
}

# Compute a repeated measures ANOVA
print("Goodness-of-Fit ANOVA and Means")
a <- lme(aic ~ mod, random = ~ 1 | subj, data = df)
print(anova(a))
with(df, print(cbind(mean = tapply(aic, mod, mean),
  std.dev = tapply(aic, mod, sd),
  n = tapply(aic, mod, length))))

# Test the hypothesis that eyes have no influence on face memory
plot2 <- function() {
  par(mfcol=c(2,1)) # two plots per figure
  # Make a strip chart of the data
  with(df, stripchart(da ~ eyes, method = "jitter", jitter = 0.03,
    xlab = "Sensitivity (Da)",
    ylab = "Eyes (Yes or No)",
    main = "Lab 2 Group Data"))

  # Make a box plot of the data
  boxplot(da ~ eyes, data = df, horizontal = TRUE,
    ylab = "Eyes (Yes or No)",
    xlab = "Sensitivity (Da)")
  par(mfcol=c(1,1)) # two plots per figure
}

# Compute the repeated measures ANOVA
print("d-sub-a ANOVA and Means")
b <- lme(da ~ eyes, random = ~ 1 | subj, data = df)
print(anova(b))
with(df, print(cbind(mean = tapply(da, eyes, mean),
  std.dev = tapply(da, eyes, sd),
  n = tapply(da, eyes, length))))
```