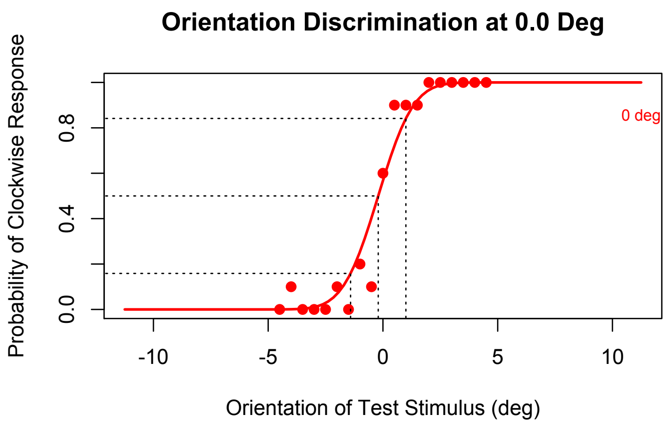
Psychology of Perception

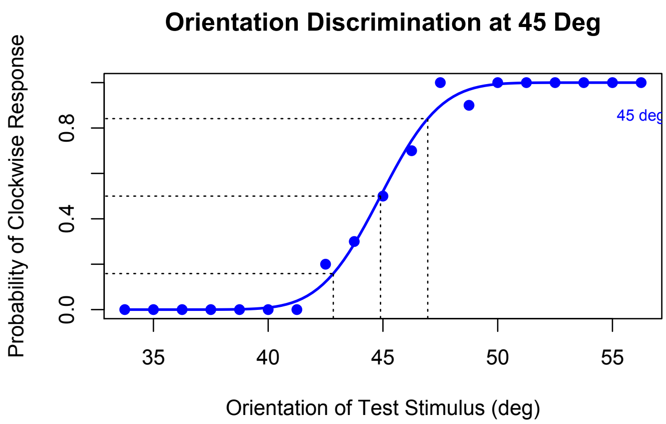
Psychology 4165, Spring 2014

Laboratory 1

Noisy Representations:

The Oblique Effect





Page intentionally blank

Introduction

Classical methods of psychophysics involve the measurement of two types of sensory thresholds: the absolute threshold, RL (*Reiz Limen*), the weakest stimulus that is *just* detectable, and the difference threshold, DL (*Differenz Limen*), the smallest stimulus increment away from a standard stimulus that is *just* detectable (also called the Just-Noticeable Difference, the JND). Gustav Theodor Fechner (1801–1887), in *Elemente der Psychophysik* (Fechner, 1860) introduced three psychophysical methods for measuring absolute and difference (JND) thresholds: the method of adjustment; the method of limits; the method of constant stimuli.

The purpose of this laboratory is to give you experience with the measurement and computation of the JND for discriminating different visual orientations using the method of constant stimuli and to test the hypothesis that discrimination at 45 degree angle is worse than at 0.0 deg. Worse performance for oblique orientations relative to vertical and horizontal orientations is found in numerous visual phenomena, and was named the "Oblique Effect" in 1972 by Stuart Appelle (Appelle, 1972). More recent papers discuss the cause of the oblique effect (Freeman, Brouwer, Heeger, & Merriam, 2011; McMahon & MacLeod, 2003; Meng & Qian, 2005; Nasr & Tootell, 2012; Westheimer, 2003).

Experiment

You will determine difference thresholds for visual orientation using the method of constant stimuli for two different visual orientations: 0.0 deg and 45.0 deg. The visual stimuli will be patches of visual grating patterns, known as Gabor patches. The orientation of the patterns will vary around 0.0 deg (vertical) and 45.0 deg (oblique). You will test the hypothesis that the just noticeable difference in orientation around vertical is different than that around the oblique.

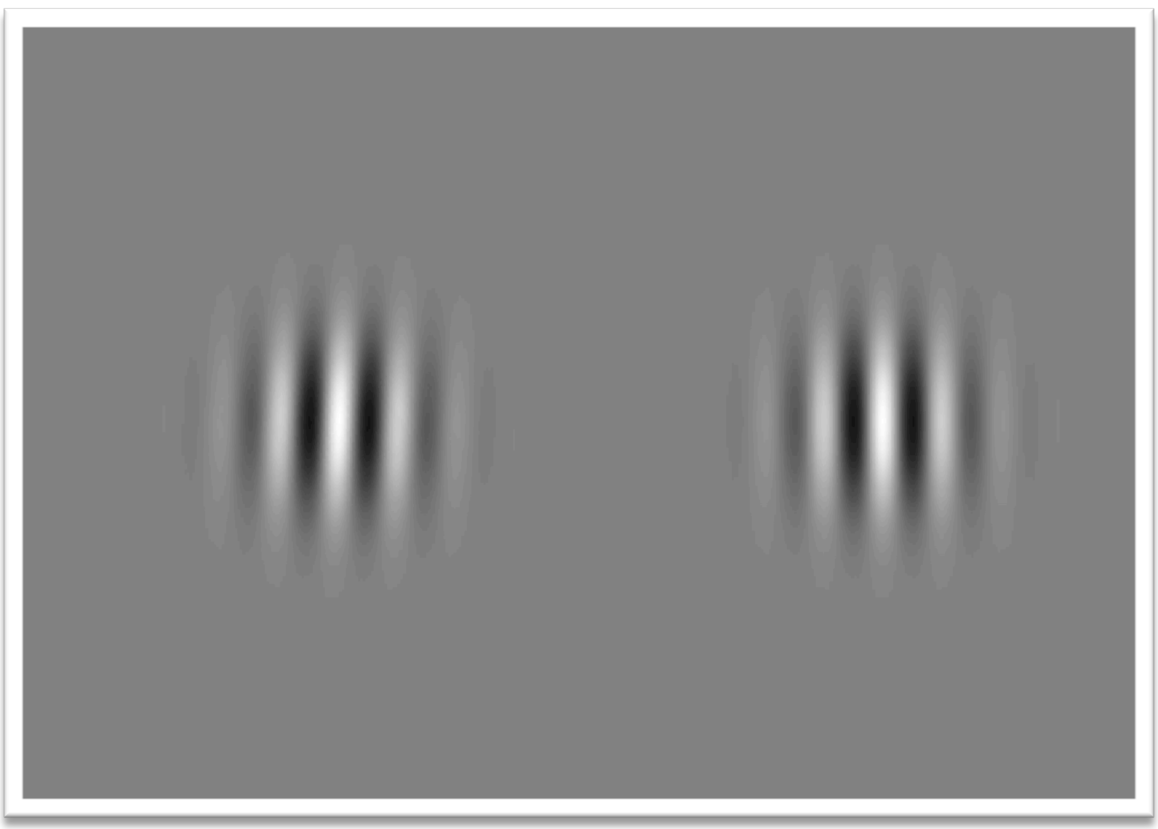
Procedure

In the method of constant stimuli, a standard stimulus is compared a number of times with test stimuli of slightly different orientation. When the difference between the standard and the comparison stimulus is large, the subject nearly always can correctly judge whether the test stimulus is rotated clockwise or counterclockwise relative to the standard. When the difference is small, however, errors are often made. The difference threshold is the transition point between differences large enough to be easily detected and those too small to be detected.

The experiment will be run under computer control using PsychoPy, a popular program written in Python by Jonathan Peirce, at the University of Nottingham, England (Peirce, 2007, 2009). PsychoPy allows you to run experiments with carefully controlled visual and auditory stimuli and to collect response data and reaction times. You will execute the experiment script: *Orientation JND Exp.py* found in the Lab_1_Tools folder that you should

download from the class web site:
http://psych.colorado.edu/~lharvey/P4165/P4165_2013_Fall/p4165(2013_Fall).html .
When you run the script, enter three initials as a subject identifier and the number 10 in the repetitions box.

The computer will randomly decide which of the two orientations to test first: 0 or 45 deg. On each trial you will first be presented with the standard stimulus (either 0 or 45 deg, depending on the condition) followed by a test stimulus. The test will be rotated slightly counterclockwise or clockwise relative to the standard. You must judge which by pressing the left arrow key (if you think the second stimulus is counterclockwise) or the right arrow key (if you think it was clockwise). The computer will record your responses on each trial. Some of the judgments will be easy and some will be difficult. The whole experiment should not take more than 30 minutes. Here are two examples of Gabor patches: one is oriented at 0.0 degrees (vertical) and the other is tilted clockwise by 2.0 degrees. Can you see the difference?



Data Tabulation and Analysis

1. The results of your experiment will be saved in a text file in the data folder whose name is your initials with the date and time added to it. The file extension is .csv, for comma separated values. If you double click on the file name, it will open in Excel. Do not modify this data file. It represents a lot of

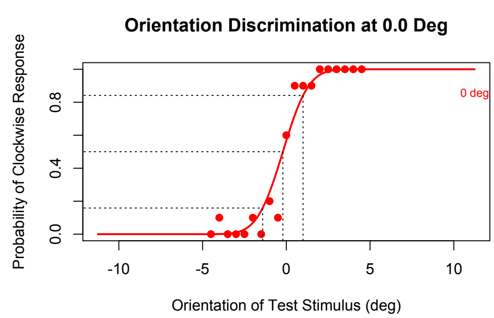
judgments and work on your part.

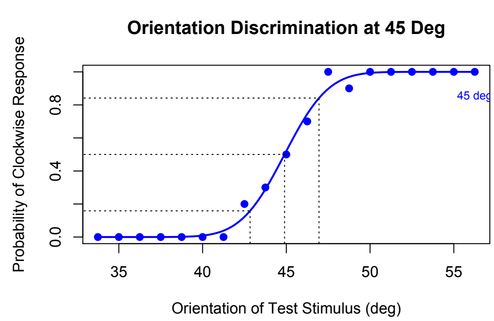
2. Use the R commands listed in the file *lab1_glm.R* (reproduced at the end of this handout) to carry out your data analysis. There are two basic steps to the analysis: 1) the generalized linear model function of R (`glm()`) is used to fit a smooth S-shaped psychometric function to your 0.0 degree and 45.0 degree data; and 2) make graphs of your results plotted in two separate figures.

3. The results of the fitting are stored in R objects `glm00` and `glm45`. These results may be viewed using the `summary()` command: `summary(glm00)` and `summary(glm45)`. The mean and standard deviations of the best-fitting Gaussian distributions are in R objects `mu00`, `sd00`, `mu45`, and `sd45` respectively. Copy these values (mu, sd, and AIC) to the table on page 8.

4. **The JND**: There are two ways to estimate the JND. One way is to compute the reciprocal of the steepness of the best-fitting psychometric function. The steepness is given by the glm coefficient corresponding to testOrientation (remember to get these values using the `summary()` command). So the steeper the function, the smaller the JND. Computed this way, one JND is equivalent to one standard deviation of the Gaussian distribution underlying the psychometric function. The second, equivalent method, is to use the difference, in degrees, between the orientation corresponding to the 0.84 point on the ordinate, and the orientation corresponding to the 0.16 point on the ordinate divided by 2.0. Once you know the JND values, do they look the same?

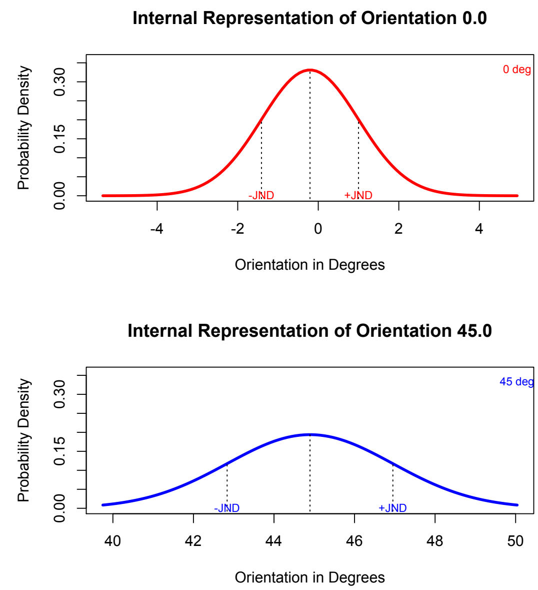
5. Prepare two graphs illustrating your results. Figure 1 should be a plot of your observed psychometric function data for the 0 deg and 45 deg standards along with the best-fitting S-shaped psychometric function. The graphic commands to make the figure on the front of this handout and below are given in the file "lab1_glm.R". Use `help(plot)` and modify the plotting parameters to achieve the kind of plot that appeals to you. Your Figure 1 should look like the graph to the right. In the R-script below, the two graphs are encapsulated in functions, `plot1()` and `plot2()` so you can redraw them any time by giving either command. You save a graph in R as a





file by clicking on the graph window and choosing save from the File menu.

6. The second figure plots the noisy representations of the 0.0 and 45 degree standard orientations using `dnorm()` using the mean and standard deviation values derived from the `glm()` model. Check out the graphing commands in the lab1_glm.R file. Your Figure 2 should like something like this:



7. **Hypothesis Testing**: Based on the curve fitting results (`summary(glm00)` and `summary(glm45)`) can you figure out whether or not your steepness values are the same for the two orientations?

8. When you have a lot of data from different people it is a good idea to make graphs of them so you get an idea what the data look like and whether or not there are differences among different groups or levels of factors. This strategy is part of what is called exploratory data analysis (Tukey, 1977). Three such plots are called histograms, strip charts and box plots. The R commands to produce them are given in the file "lab1_lme.R".
9. **Hypothesis Testing Using Group Data**: We will assemble your individual data into a single data file that will be available for the next lab meeting. Test the hypothesis, using R, that the value of the JND is the same for 0 degrees as for 45 degrees. The appropriate analysis is a repeated measures analysis of variance. We cannot use a standard ANOVA for repeated measures: We need to use a so-called linear mixed model instead. The `lme()` command in R is one way to do this analysis. The R commands for doing this analysis are contained in the file "lab1_lme.R" which is reproduced at the end of this handout. Also test the hypothesis that the goodness-of-fit (AIC) is the same for the two orientation conditions using `lme()`.

Lab Report

Your lab report should be brief and contain five sections: cover sheet, introduction, methods, results, and discussion. These sections should conform to the American Psychological Association (APA) style (American Psychological Association., 2010) as described in Chapter 13 of the Martin textbook (Martin, 2007). The results section should have the graphs described above and a table giving the JND for the 0.0 and 45.0 degree conditions. Why might there be a difference between the two orientations? What causes the oblique effect?

The report is due at the beginning of lab meeting on **4 & 6 February 2014**. Late labs will receive a grade of zero. All lab reports must be prepared with a word processor. This lab report is worth 30 points.

References

American Psychological Association. (2010). *Publication manual of the American Psychological Association* (6th ed.). Washington, DC: American Psychological Association.

Appelle, S. (1972). Perception and discrimination as a function of stimulus orientation: The oblique effect in man and animals. *Psychological Bulletin, 78*, 266–278.

Fechner, G. T. (1860). *Elemente der Psychophysik*. Leipzig, Germany: Breitkopf and Härtel.

Freeman, J., Brouwer, G. J., Heeger, D. J., & Merriam, E. P. (2011). Orientation Decoding Depends on Maps, Not Columns. *The Journal of Neuroscience, 31*(13), 4792-4804. doi: 10.1523/jneurosci.5160-10.2011

Martin, D. W. (2007). *Doing psychology experiments* (7th ed.). Belmont, CA: Thomson Wadsworth.

McMahon, M. J., & MacLeod, D. I. A. (2003). The origin of the oblique effect examined with pattern adaptation and masking. *Journal of Vision, 3*(3), 230–239.

Meng, X., & Qian, N. (2005). The oblique effect depends on perceived, rather than physical, orientation and direction. *Vision Research, 45*(27), 3402-3413. doi: http://dx.doi.org/10.1016/j.visres.2005.05.016

Nasr, S., & Tootell, R. B. H. (2012). A Cardinal Orientation Bias in Scene-Selective Visual Cortex. *The Journal of Neuroscience, 32*(43), 14921-14926. doi: 10.1523/jneurosci.2036-12.2012

Peirce, J. W. (2007). PsychoPy--Psychophysics software in Python. *Journal of Neuroscience Methods, 162*(1-2), 8-13. doi: 10.1016/j.jneumeth.2006.11.017

Peirce, J. W. (2009). Generating stimuli for neuroscience using PsychoPy. *Frontiers in Neuroinformatics, 2*(January), 1–8. doi: 10.3389/neuro.11.010.2008

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.

Westheimer, G. (2003). Meridional anisotropy in visual processing: implications for the neural site of the oblique effect. *Vision Research, 43*(22), 2281-2289. doi: http://dx.doi.org/10.1016/S0042-6989(03)00360-2

Curve-Fitting Summary from glm()

		Mean		Std Deviation		Index-of-Fit: AIC	
Name	Order	mu00	mu45	sd00	Sd45	00 deg	45 deg

```
# PSYC 4165 Lab 1
# Generalized Linear Model (glm)
# script file for using the R command glm() to compute the best-fitting
# Gaussian integral (Gaussian CDF) to psychometric function data.
# Lewis O. Harvey, Jr.
# Department of Psychology
# University of Colorado
# 18 January 2014

# ****************************************************
# Analyze the data from the csv file
# for Orientation JND lab individual data
# The raw data file produced by the PsychoPy script
# contains the trial by trial results.
# ****************************************************
fn <- file.choose()
df.raw <- read.csv(fn, header=TRUE)

# put the two sets of data (0 and 45 degrees) into separate data frames
# to facilitate the separate analyses
df.raw.00 <- subset(df.raw, standardOrientation==0)
df.raw.45 <- subset(df.raw, standardOrientation==45)

# compute the probability of making a "clockwise" judgment for each
# of the test stimuli for each of the standard orientations.
# Since clockwise is coded as 1 and counterclockwise as 0,
# the mean of the responses is the probability.
# Also compute the total number of clockwise and counterclockwise responses
# for each condition, needed for glm()
t1 <- tapply(df.raw.00$n_clockwise, INDEX=list(df.raw.00$testOrientation), FUN=mean)
t2 <- tapply(df.raw.00$n_clockwise, INDEX=list(df.raw.00$testOrientation), FUN=sum)
t3 <- tapply(df.raw.00$n_counterclockwise, INDEX=list(df.raw.00$testOrientation),
 FUN=sum)

t4 <- tapply(df.raw.45$n_clockwise, INDEX=list(df.raw.45$testOrientation), FUN=mean)
t5 <- tapply(df.raw.45$n_clockwise, INDEX=list(df.raw.45$testOrientation), FUN=sum)
t6 <- tapply(df.raw.45$n_counterclockwise, INDEX=list(df.raw.45$testOrientation),
 FUN=sum)

# form the data frames for glm() and for plotting the probabilities
df.00 <- data.frame(testOrientation=as.numeric(names(t1)), p_clock=t1, n_clock=t2,
 n_counter=t3, row.names=1:length(t1))
df.45 <- data.frame(testOrientation=as.numeric(names(t4)), p_clock=t4, n_clock=t5,
 n_counter=t6, row.names=1:length(t4))

df.00 <- data.frame(df.00, orientation_offset = df.00$testOrientation - 0.0,
 standard_orientation = 0)
df.45 <- data.frame(df.45, orientation_offset = df.45$testOrientation - 45.0,
 standard_orientation = 45)

# compute the best-fitting Gaussian integrals for each set of data
# using a genealized linear model, glm().
# The "probit" link specifies that a Gaussian probability distribution
# is being fit to the data.
# The "binomial" family specifies that the dependent variable
# (number of clockwise and counterclockwise judgments) are drawn from
# a binomial distribution.
glm.00 <- glm(cbind(n_clock, n_counter) ~ testOrientation,
 data = df.00,
 family = binomial(link = "probit"))
glm.45 <- glm(cbind(n_clock, n_counter) ~ testOrientation,
 data = df.45,
 family = binomial(link = "probit"))

# extract the glm coefficients beta0 and beta1 for each standard orientation
b0.00 <- coefficients(glm.00)[1]
b1.00 <- coefficients(glm.00)[2]

b0.45 <- coefficients(glm.45)[1]
b1.45 <- coefficients(glm.45)[2]

# compute the mean and standard deviations of the Gaussian
```

```
# probability density functions from the glm solutions

# mean of the 0 deg representation
mu.00 <- -b0.00 / b1.00
# standard deviation of the 0 deg representation
sd.00 <- 1 / b1.00
# mean of the 45 deg representation
mu.45 <- -b0.45 / b1.45
# standard deviation of the 45 deg representation
sd.45 <- 1 / b1.45

# get the cumulative probability corresponding to plus and minus 1 standard deviation
# that we define to correspond to 1 JND (these should be 0.16 and 0.84)
pm.JND <- pnorm(-1.0)      # probability 1 sd below the mean
pp.JND <- pnorm( 1.0)      # probability 1 sd above the mean

# weights corresponding to 1 JND above and below the means
q1 <- qnorm(pm.JND, mu.00, sd.00)# lower jnd for 0 deg
q2 <- qnorm(pp.JND, mu.00, sd.00)# upper jnd for 0 deg
q3 <- qnorm(pm.JND, mu.45, sd.45)# lower jnd for 45 deg
q4 <- qnorm(pp.JND, mu.45, sd.45)# upper jnd for 45 deg

# compute the JNDs from lower and upper 1 sd points
jnd.00 <- (q2 - q1) / 2.0         # jnd in degrees
jnd.45 <- (q4 - q3) / 2.0         # jnd in degrees


# *****************************************************
# Plotting Section
# Define three function that draw the plots
# plot1() two psychometric functions in two panels
# plot1a() two psychometric functions in one panel
# plot2() two Gaussian distributions in two panels
# To create the plots, type one of the commands
# *****************************************************

# test stimuli used under the 100 and 200 standard conditions
ts.00 <- df.00$testOrientation
ts.45 <- df.45$testOrientation

# plot Figure 1, the psychometric functions
# To draw the plot, just type the function name: plot1()
plot1 <- function() {
 par(mfrow = c(2,1))
 # find the bigest range of test increments and set
 # the x-axis min and max for both plots so the
 # range is the same for 0.0 and 45.0
 range.plot <- range(df.raw$testIncrement)
 # plot 0.0 results
 xxmin <- (0.0 + range.plot[1])
 xxmax <- (0.0 + range.plot[2])
 plot(p_clock ~ testOrientation,
       data = df.00,
       type = "p",
       col = "red",
       pch = 19,
       xlim = c(xxmin, xxmax),
       ylim = c(0, 1),
       xlab="Orientation of Test Stimulus (deg)",
       ylab="Probability of Clockwise Response",
       main="Orientation Discrimination at 0.0 Deg")
 # plot the predicted smooth curves
 x <- seq(from = xxmin, to = xxmax, length.out = 100)
 y <- predict(glm.00, data.frame(testOrientation=x), type = "response")
 lines(y ~ x, lwd = 2, col = "red")

 # label the line
 text(xxmax, .85, "0 deg", col="red",  cex=0.75)

 # plot the alpha markers (the point of subjective equality)
 lines(c(mu.00, mu.00), c(-0.1, 0.5), lty=3)
 lines(c(-20, mu.00), c(0.5, 0.5), lty=3)
```

```
 # now draw the vertical JND lines (lty=3 makes the line dashed)
 lines(c(q1, q1), c(-0.1, pm.JND), lty=3)
 lines(c(q2, q2), c(-0.1, pp.JND), lty=3)

 # draw the horizontal JND probability lines
 lines(c(-20, q1), c(pm.JND, pm.JND), lty=3)
 lines(c(-20, q2), c(pp.JND, pp.JND), lty=3)

 # plot 45 deg results
 xxmin <- (45.0 + range.plot[1])
 xxmax <- (45.0 + range.plot[2])
 plot(p_clock ~ testOrientation,
       data = df.45,
       type = "p",
       col = "blue",
       pch = 19,
       xlim = c(xxmin, xxmax),
       ylim = c(0, 1),
       xlab="Orientation of Test Stimulus (deg)",
       ylab="Probability of Clockwise Response",
       main="Orientation Discrimination at 45 Deg")

 # plot the predicted smooth curves
 x <- seq(from = xxmin, to = xxmax, length.out = 100)
 y <- predict(glm.45, data.frame(testOrientation=x), type = "response")
 lines(y ~ x, lwd = 2, col = "blue")

 # label each line
 text(xxmax, .85, "45 deg", col="blue", cex=0.75)

 # plot the alpha markers (the point of subjective equality)
 lines(c(mu.45, mu.45), c(-0.1, 0.5), lty=3)
 lines(c(20, mu.45), c(0.5, 0.5), lty=3)

 # now draw the vertical JND lines (lty=3 makes the line dashed)
 lines(c(q3, q3), c(-0.1, pm.JND), lty=3)
 lines(c(q4, q4), c(-0.1, pp.JND), lty=3)

 # draw the horizontal JND probability lines
 lines(c(20, q3), c(pm.JND, pm.JND), lty=3)
 lines(c(20, q4), c(pp.JND, pp.JND), lty=3)

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

# plot Figure 1a, the psychometric functions drawn as offsets
# from the standard orientation
# To draw the plot, just type the function name: plot1a()
plot1a <- function() {
 # find the bigest range of test increments and set
 # the x-axis min and max for both plots so the
 # range is the same for 0.0 and 45.0
 range.plot <- range(df.raw$testIncrement)
 # plot 0.0 results
 xxmin <- (0.0 + range.plot[1])
 xxmax <- (0.0 + range.plot[2])
 plot(p_clock ~ orientation_offset,
       data = df.00,
       type = "p",
       col = "red",
       pch = 19,
       xlim = c(xxmin, xxmax),
       ylim = c(0, 1),
       xlab="Difference between Test and Standard Orientation (deg)",
       ylab="Probability of Clockwise Response",
       main="Figure 1: Orientation Discrimination Psychometric Functions")
 points(p_clock ~ orientation_offset,
       data = df.45,
       pch = 19,
       col = "blue")
```

```
glm.00.offset <- glm(cbind(n_clock, n_counter) ~ orientation_offset,
      data = df.00,
      family = binomial(link = "probit"))
glm.45.offset <- glm(cbind(n_clock, n_counter) ~ orientation_offset,
      data = df.45,
      family = binomial(link = "probit"))

# extract the glm coefficients beta0 and beta1 for each standard orientation
b0.00.offset <- coefficients(glm.00.offset)[1]
b1.00.offset <- coefficients(glm.00.offset)[2]

b0.45.offset <- coefficients(glm.45.offset)[1]
b1.45.offset <- coefficients(glm.45.offset)[2]

# compute the mean and standard deviations of the Gaussian
# probability density functions from the glm solutions

# mean of the 0 deg representation
mu.00.offset <- -b0.00.offset / b1.00.offset
# standard deviation of the 0 deg representation
sd.00.offset <- 1 / b1.00.offset
# mean of the 45 deg representation
mu.45.offset <- -b0.45.offset / b1.45.offset
# standard deviation of the 45 deg representation
sd.45.offset <- 1 / b1.45.offset

# get the cumulative probability corresponding to plus and minus 1 standard deviation
# that we define to correspond to 1 JND (these should be 0.16 and 0.84)
pm.JND <- pnorm(-1.0)     # probability 1 sd below the mean
pp.JND <- pnorm( 1.0)     # probability 1 sd above the mean

# weights corresponding to 1 JND above and below the means
q1.offset <- qnorm(pm.JND, mu.00.offset, sd.00.offset)     # lower jnd for 0 deg
q2.offset <- qnorm(pp.JND, mu.00.offset, sd.00.offset)     # upper jnd for 0 deg
q3.offset <- qnorm(pm.JND, mu.45.offset, sd.45.offset)     # lower jnd for 45 deg
q4.offset <- qnorm(pp.JND, mu.45.offset, sd.45.offset)     # upper jnd for 45 deg

# compute the JNDs from lower and upper 1 sd points
jnd.00.offset <- (q2 - q1) / 2.0       # jnd in degrees
jnd.45.offset <- (q4 - q3) / 2.0       # jnd in degrees

# plot the predicted smooth curves
x <- seq(from = xxmin, to = xxmax, length.out = 100)
y <- predict(glm.00.offset, data.frame(orientation_offset=x), type = "response")
lines(y ~ x, lwd = 2, col = "red")
y <- predict(glm.45.offset, data.frame(orientation_offset=x), type = "response")
lines(y ~ x, lwd = 2, col = "blue")

# label each line
text(xxmax, .85, "0 deg",  col="red",  cex=0.75, pos=2)
text(xxmax, .78, "45 deg", col="blue", cex=0.75, pos=2)

# plot the vertical alpha markers (the point of subjective equality)
lines(c(mu.00.offset, mu.00.offset), c(-0.1, 0.5), lty=3, col="red")
lines(c(mu.45.offset, mu.45.offset), c(-0.1, 0.5), lty=4, col="blue")
# plot the horzontal alpha markers
lines(c(-15, mu.00.offset), c(0.5, 0.5), lty=3)
lines(c(-15, mu.45.offset), c(0.5, 0.5), lty=3)

# now draw the vertical JND lines (lty=3 makes the line dashed)
lines(c(q1.offset, q1.offset), c(-0.1, pm.JND), lty=3, col="red")
lines(c(q2.offset, q2.offset), c(-0.1, pp.JND), lty=3, col="red")
lines(c(q3.offset, q3.offset), c(-0.1, pm.JND), lty=4, col="blue")
lines(c(q4.offset, q4.offset), c(-0.1, pp.JND), lty=4, col="blue")

# draw the horizontal JND probability lines
lines(c(-20, q1.offset), c(pm.JND, pm.JND), lty=3)
lines(c(-20, q2.offset), c(pp.JND, pp.JND), lty=3)
lines(c(-20, q3.offset), c(pm.JND, pm.JND), lty=3)
lines(c(-20, q4.offset), c(pp.JND, pp.JND), lty=3)
}
```

```
# plot Figure 2, Gaussian stimulus representations
# To draw the plot, just type the function name: plot2()
plot2 <- function() {
 par(mfrow = c(2,1))

 sd.max <- max(sd.00, sd.45)
 xxmin.00 <- (mu.00 - 2.5 * sd.max)
 xxmax.00 <- (mu.00 + 2.5 * sd.max)

 xxmin.45 <- (mu.45 - 2.5 * sd.max)
 xxmax.45 <- (mu.45 + 2.5 * sd.max)

 # generate a range of x values
 x.00 <- seq(from = xxmin.00, to = xxmax.00, length.out = 100)
 x.45 <- seq(from = xxmin.45, to = xxmax.45, length.out = 100)
 d.00 <- dnorm(x.00, mu.00, sd.00)             # probability densities
 d.45 <- dnorm(x.45, mu.45, sd.45)             # probability densities
 dmax  <- max(d.00, d.45)                      # maximum density value

 # plot 0.0 deg orientation density function
 plot(d.00 ~ x.00,
       type = "l",
       lwd = 3,
       col = "red",
       xlim = c(xxmin.00, xxmax.00),
       ylim = c(0, dmax * 1.08),
       xlab = "Orientation in Degrees",
       ylab = "Probability Density",
       main = "Internal Representation of Orientation 0.0")

 # probability density at each mean
 dmu.00 <- dnorm(mu.00, mu.00, sd.00)
 lines(c(mu.00, mu.00), c(-0.1, dmu.00), lty=3)

 # probability densities at each JND point
 d1 <- dnorm(q1, mu.00, sd.00)
 d2 <- dnorm(q2, mu.00, sd.00)

 # now draw the vertical JND lines (lty=3 makes the line dashed)
 lines(c(q1, q1), c(-0.1, d1), lty=3)
 lines(c(q2, q2), c(-0.1, d2), lty=3)

 # label the distribution
 text(xxmax.00, dmax, "0 deg", col="red",  cex=0.75)

 # label the JND markers
 text(q1, 0.001, "-JND", col="red",  cex=0.75)
 text(q2, 0.001, "+JND", col="red",  cex=0.75)

 # plot 45 deg orientation density function
 plot(d.45 ~ x.45,
       type = "l",
       lwd = 3,
       col = "blue",
       xlim = c(xxmin.45, xxmax.45),
       ylim = c(0, dmax * 1.08),
       xlab = "Orientation in Degrees",
       ylab = "Probability Density",
       main = "Internal Representation of Orientation 45.0")

 # probability density at each mean
 dmu.45 <- dnorm(mu.45, mu.45, sd.45)
 lines(c(mu.45, mu.45), c(-0.1, dmu.45), lty=3)

 # probability densities at each JND point
 d3 <- dnorm(q3, mu.45, sd.45)
 d4 <- dnorm(q4, mu.45, sd.45)

 # now draw the vertical JND lines (lty=3 makes the line dashed)
 lines(c(q3, q3), c(-0.1, d3), lty=3)
 lines(c(q4, q4), c(-0.1, d4), lty=3)
```

```
 # label the distribution
 text(xxmax.45, dmax, "45 deg", col="blue", cex=0.75)

 # label the JND markers
 text(q3, 0.001, "-JND", col="blue", cex=0.75)
 text(q4, 0.001, "+JND", col="blue", cex=0.75)

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

plot1()
```

```
# PSYCH 4165 Lab 1
# command file for computing a repeated measures
# analysis of variance using lme() in R.
# Lewis O. Harvey, Jr.
# Department of Psychology
# University of Colorado
# 26 January 2014

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

# read in group data in wide format (from Excel text file)
fn <- file("lab_1_group_data_wide_Spring_2014.csv")
df.wide <- read.csv(fn, header=TRUE)

# *******************************p***********************
# 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("mu00","mu45"), c("sd00", "sd45"), c("aic00", "aic45")),
 idvar = "subject",
 times = c("00", "45"),
 v.names = c("mu", "sd", "aic"),
 timevar = "standard",
 direction = "long")
rownames(df.long) <- 1:dim(df.long)[1]
df.long$standard <- factor(df.long$standard) # make standard a factor
# ******************************************************

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

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

# ******************************************************
# plotting section
# define eight functions to plot the two sets of graphs
# plot1() JND data
# plot2() JND data
# plot3() AIC data
# plot4() AIC data
# plot5() scatter plots of mu, sd, aic
# plot6() scatter plots for 0.0 deg standard
# plot7() scatter plots for 45.0 deg standard
# plot8() mu vs sd plot
# To make the plots, type the command

# ******************************************************
# Make a strip chart of JND for each standard orientation
plot1 <- function() {
 # set graphic parameters to plot two graphs in one panel
 par(mfcol = c(1,2))
 stripchart(sd ~ standard, data = df,
       method = "jitter", jitter = 0.07,
       vertical = TRUE,
       xlim = c(0.75, 2.25),
       xlab = "Standard Orientation",
       ylab = "Orientation JND in Degrees",
       main = "PSYC 4165 - Spring 2014")

 # Make a box plot of JND for each standard orientation
 boxplot(sd ~ standard, data = df,
       col = "yellow",
       horizontal = FALSE,
       xlab = "Standard Orientation",
       ylab = "Orientation JND in Degrees",
       main = "PSYC 4165 - Spring 2014")
```

```
 par(mfcol = c(1,1))
}

# *******************************************************
# Make a strip chart of JND for each testing order
plot2 <- function() {
 # set graphic parameters to plot two graphs in one panel
 par(mfcol = c(1,2))
 stripchart(sd ~ order, data = df,
       method = "jitter", jitter = 0.07,
       vertical = TRUE,
       xlim = c(0.75, 2.25),
       xlab = "Testing Order",
       ylab = "Orientation JND in Degrees",
       main = "PSYC 4165 - Spring 2014")

 # Make a box plot of mean for each testing order
 boxplot(sd ~ order, data = df,
       col = "yellow",
       horizontal = FALSE,
       xlab = "Testing Order",
       ylab = "Orientation JND in Degrees",
       main = "PSYC 4165 - Spring 2014")
 par(mfcol = c(1,1))
}
# *******************************************************

# Make a strip chart of AIC for each standard orientation
plot3 <- function() {
 # set graphic parameters to plot two graphs in one panel
 par(mfcol = c(1,2))
 stripchart(aic ~ standard, data = df,
       method = "jitter", jitter = 0.07,
       vertical = TRUE,
       xlim = c(0.75, 2.25),
       xlab = "Standard Orientation",
       ylab = "Goodness-of-Fit (AIC)",
       main = "PSYC 4165 - Spring 2014")

 # Make a box plot of PSE for each standard orientation
 boxplot(aic ~ standard, data = df,
       col = "yellow",
       horizontal = FALSE,
       xlab = "Standard Orientation",
       ylab = "Goodness-of-Fit (AIC)",
       main = "PSYC 4165 - Spring 2014")
 par(mfcol = c(1,1))
}
# *******************************************************

# Make a strip chart of AIC for each order
plot4 <- function() {
    # set graphic parameters to plot two graphs in one panel
    par(mfcol = c(1,2))
    stripchart(aic ~ order, data = df,
              method = "jitter", jitter = 0.07,
              vertical = TRUE,
              xlim = c(0.75, 2.25),
              xlab = "Testing Order",
              ylab = "Goodness-of-Fit (AIC)",
              main = "PSYC 4165 - Spring 2014")

    # Make a box plot of aic for each order
    boxplot(aic ~ order, data = df,
           col = "yellow",
           horizontal = FALSE,
           xlab = "Testing Order",
           ylab = "Goodness-of-Fit (AIC)",
           main = "PSYC 4165 - Spring 2014")
    par(mfcol = c(1,1))
}
# *******************************************************
```

```
# make scatter plots of all three continuous dependent variables
plot5 <- function() {
 par(mfcol = c(1,1))
 pairs(df[, c("mu", "sd", "aic")])
 par(mfcol = c(1,1))
}
# *******************************************************

# make scatter plots of all three continuous dependent variables
# for the 0 deg standard
plot6 <- function() {
 par(mfcol = c(1, 2))
 pairs(subset(df, standard == "00")[, c("mu", "sd", "aic")], main = "0 deg Standard")
 par(mfcol = c(1, 1))
}
# *******************************************************

# make scatter plots of all three continuous dependent variables
# for the 45 deg standard
plot7 <- function() {
 par(mfcol = c(1, 2))
 pairs(subset(df, standard == "45")[, c("mu", "sd", "aic")], main = "45 deg Standard")
 par(mfcol = c(1, 1))
}
# *******************************************************

# plot the JND of 0.0 deg against the JND for 45 deg
# *******************************************************
plot8 <- function() {
 df.x <- subset(df, standard == "00")
 df.y <- subset(df, standard == "45")
 plot(df.y[,"sd"] ~ df.x[,"sd"],
       pch = 19,
       xlab = "JND for 0.0 degrees",
       ylab = "JND for 45.0 degrees",
       main = "JND vs Standard Orientation")
 lm.mod <- lm(df.y[,"sd"] ~ df.x[,"sd"])
 abline(lm.mod)
 print(summary(lm.mod))
}

# compute the mean JNDs
print(" ", quote = FALSE)
print("mean JND of orders", quote = FALSE)
print(with(df,tapply(sd, order, "mean")))
print(" ", quote = FALSE)
print("mean JND of standards", quote = FALSE)
print(with(df,tapply(sd, standard, "mean")))

# compute the mean AICs
print(" ", quote = FALSE)
print("mean AIC of orders", quote = FALSE)
print(with(df,tapply(aic, order, "mean")))
print(" ", quote = FALSE)
print("mean AIC of standards", quote = FALSE)
print(with(df,tapply(aic, standard, "mean")))

# use lme() to compute the ANOVA on JNDs
print(" ", quote = FALSE)
print("Results of Linear Mixed Effects Analysis of Variance on JND", quote = FALSE)

lme.jnd <- lme(sd ~ standard * order, data = df, random = ~ 1 | subject)
print(anova(lme.jnd))

# use lme() to compute the ANOVA on AICs
print(" ", quote = FALSE)
print("Results of Linear Mixed Effects Analysis of Variance on AIC", quote = FALSE)

lme.aic <- lme(aic ~ standard * order, data = df, random = ~ 1 | subject)
print(anova(lme.aic))
plot1()
```