

# Functions for traditional and multilevel approaches to signal detection theory

DANIEL B. WRIGHT

*Florida International University, Miami, Florida*

AND

RUTH HORRY AND ELIN M. SKAGERBERG

*University of Sussex, Brighton, England*

In the present article, functions written in the freeware R are presented that calculate several measures from traditional signal detection theory for each individual in a sample, along with summary statistics for the sample. Bias-corrected and accelerated bootstrap confidence intervals are also produced. Arguments are made for using an alternative approach—multilevel generalized linear models—and a function is presented for it. These functions are part of the R package `sdtalt`, which is available on the Comprehensive R Archive Network. Recent data from memory recognition studies are used to illustrate these functions.

Methods from signal detection theory (SDT) are popular in many areas of science, including psychology (Swets, 1996; Wickens, 2002). The traditional approach to SDT within psychology is to calculate a measure of diagnosticity (or accuracy) for each individual and then to perform statistics on these aggregate numbers. There are several different measures that could be used, along with arguments among methodologists about their relative merits, but a popular measure in psychology is  $d'$ . In SDT terminology,  $d'$  is the distance, in standard deviations, between the distribution for items with whichever characteristic is being looked for and the distribution for items without this characteristic. Calculating aggregate values for each individual and then performing statistics on them has both theoretical and practical disadvantages. A theoretical disadvantage is the fact that participants' values will be weighted equally, which may not be appropriate if they have been involved in different numbers of trials. A practical disadvantage is the fact that it is difficult to include variables that vary among trials, such as response time (RT).

The traditional SDT model, sometimes called the *Gaussian/normal model*, is equivalent to a probit regression (DeCarlo, 1998). In medical diagnostics, the logistic model is usually used. This model is equivalent to running a logistic regression for each individual (Zhou, Obuchowski, & McClish, 2002). These two approaches, which are both generalized linear models (GLMs; McCullagh & Nelder, 1989), yield nearly equivalent results. The main statistics of accuracy for these models are  $d'$  and  $\ln\text{OR}$  (i.e., the log-odds-ratio) for the Gaussian model and the logistic model, respectively. And, except at the extremes, these two measures are approximately propor-

tional ( $d' \approx .6 \ln\text{OR}$ ). The flexibility of GLMs allows variables that can take different values for each trial to be easily incorporated, thus addressing the practical disadvantage listed above. Analyzing the data with a multilevel model addresses the theoretical disadvantage from above. It allows the analysis to be conducted in a single step and weighs the data according to the number of trials, subject to some further assumptions. For example, it is usually assumed that the  $d'$  or the  $\ln\text{OR}$  values come from a normally distributed population (for recent reviews of multilevel modeling written for psychologists, see Baayen, Davidson, & Bates, 2008; Hoffman & Rovine, 2007; and Wright & London, 2009).

The remainder of the present article is divided into four main sections. The two main functions described require different data input formats. In the first section, functions that allow the user to move between these formats are described. In the next section, the function `sdt` is presented for the traditional approach. Several different measures are reported for each individual participant, and the user has the option of reporting the mean (or a trimmed mean) for the sample. Users can also enter their own statistics into the function. Bias-corrected and accelerated (BCa) bootstrap confidence intervals can be printed for all of these measures. Next, the `mlmsdt` function is presented for the multilevel approach. The user can include both numeric and categorical covariates, and can see how these relate to diagnosticity for either the normal or the logistic model. The function produces what is called an *S4 class object*, which allows many sample and individual statistics to be evaluated. The final section is a summary of these functions.

R was used for these functions because of its popularity, because it is free, and because of the quality and quantity of available functions (R Development Core Team, 2008). We will assume that users have some experience with R or S-Plus (and that they have access to the Internet). The functions and example data sets have been combined into the package `sdtalt`, which is part of the Comprehensive R Archive Network (CRAN). To load the functions onto your hard drive from within R, type

```
install.packages("sdtalt")
```

You may need to choose among locations from which to download the package (there are about 50 mirror sites for CRAN). To make the functions active, type

```
library(sdtalt)
```

The present article is based on `sdtalt 0.1-0`.

One area in which SDT is common in psychology is memory recognition research (Banks, 1970; Wright, Gabbert, Memon, & London, 2008). To make the descriptions more concrete, examples from memory recognition will be used.

### Changing the Format:

#### `format2to4` and `format4to2`

These two functions are used to change the data between the formats used for the `sdt` and `mlmsdt` functions. For `format2to4`, three variables (subject number, true state of the item, and what the participant says) are entered, with a separate line for each trial. If you have a within-subjects design, then let each condition by person correspond to a different subject number, or run the conditions separately. The function assumes that saying “old” and being old (using the memory recognition terms) have higher values than do saying “new” and being new. The output is an  $n \times 5$  numeric data matrix, where the  $n$  refers to the subject number, with columns named `subno`, `hits`, `fa`, `misses`, and `cr`. You can change the column names returned by specifying `cnames = c("c1", "c2", "c3", "c4", "c5")`, where `c1`–`c5` are the names you want.

$$\begin{array}{c}
 \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix}
 \end{array}
 \begin{array}{c}
 \xrightarrow{\text{format2to4}} \\
 \xleftarrow{\text{format4to2}}
 \end{array}
 \begin{array}{c}
 \begin{bmatrix} 1 & 2 & 3 & 1 & 1 \\ 2 & 1 & 1 & 0 & 1 \end{bmatrix}
 \end{array}$$

For `format4to2`, you enter an  $n \times 5$  numeric data matrix, where  $n$  is the number of subjects (with the column ordering the same as the output for `format2to4`), and it outputs a data frame with three columns (corresponding to the input for `format2to4`). This function also allows you to name the variables that are output. The defaults—

`subno`, `isold`, and `saysold`—are based on a typical memory recognition study, but SDT has many other applications. To change these, you can write, for example, `cnames = c("partno", "true", "says")` within the function.

An obvious question is why we have not allowed both of the main functions to accept data in either format, recognize the form of the input (by looking at the dimensions of the input data frame), and then call the appropriate function. This would be simple to do and would be appropriate if we were producing a stand-alone package. However, because this is part of R, we assume that users will want to perform other statistics on these reformatted data frames. Of course, if users do not want to produce two commands or for some reason are against creating a new data object, they can simply nest the reformatting functions within the others.

### Traditional SDT Function: `sdt`

For any individual, the situation in which SDT has traditionally been used can be summarized by four counts: hits, false alarms, misses, and correct rejections.<sup>1</sup> These can be summarized within a  $2 \times 2$  table (like Table 1) for each individual. The interest is usually in the association between the rows and the columns. Despite the simplicity of Table 1, there has been more than a century of disagreement about how to measure this association, and there is still disagreement about when to use each of the measures (for a recent review, see Kraemer, 2006; for historical coverage, see Goodman & Kruskal, 1979, chap. 2). Rather than getting entangled in these debates, the function `sdt` calculates 15 of the most common measures. The equations for these, shown in Table 2, were taken from Kraemer (2006), Stanislaw and Todorov (1999), and Swets (1996, chap. 3). These references should be consulted for comparisons among the measures.

R functions have two types of input: information that is required and options that can be input, but if they are absent, they revert to default values (for an introduction to R designed for psychologists, see chap. 1 of Wright & London, 2009). The function `sdt` requires four variables: the numbers of hits, false alarms, misses, and correct rejections for each participant. If the data are in the format of variables for the participants' responses (i.e., old or new) and for the true state of the item (i.e., old or new), they can be changed into the frequencies of hits, false alarms, misses, and correct rejections by using the function `format2to4`. An example of this will be shown below. The function `sdt` will work for an individual participant by entering single values for each of these [e.g., `sdt(32, 45, 63, 44)`], but the more common use would be to analyze multiple participants; therefore, four variables—each of the same length—would

**Table 1**  
**A  $2 \times 2$  Cross-Tabulation**

		What the Person Says	
		"Old"	"New"
True State of the Item	Old	hit	miss
	New	false alarm	correct rejection

**Table 2**  
**The Statistics Reported by sdt, Their Names**  
**Used With the meas Option, and Their Formulas**

Statistic	Name	Formula
Hit rate	HR	$\frac{\text{hits}}{\text{hits} + \text{misses}}$
False alarm rate	FAR	$\frac{\text{false alarms}}{\text{false alarms} + \text{correct rejections}}$
$d'$	d	$z_{\text{HR}} - z_{\text{FAR}}$
C	cstdt	$-.5(z_{\text{HR}} + z_{\text{FAR}})$
$A'$	A	$.5 + \text{sign}(\text{HR} - \text{FAR}) \frac{(\text{HR} - \text{FAR})^2 +  \text{HR} - \text{FAR} }{4(\max(\text{HR}, \text{FAR}) - \text{HR} \cdot \text{FAR})}$
$B''$	B	$\text{sign}(\text{HR} - \text{FAR}) \frac{\text{HR}(1 - \text{HR}) - \text{FAR}(1 - \text{FAR})}{\text{HR}(1 - \text{HR}) + \text{FAR}(1 - \text{FAR})}$
$\ln \beta$	lnbeta	$\frac{z_{\text{FAR}}^2 - z_{\text{HR}}^2}{2}$
$\beta$	beta	$e^{\ln \beta}$
Odds ratio	OR	$\frac{\text{hits} \cdot \text{correct rejections}}{\text{misses} \cdot \text{false alarms}}$
$\ln \text{OR}$	lnOR	$\ln(\text{OR})$
Weighted $\kappa$	kappa	$t1/t2$
$\varphi$	phi	$\frac{t1}{\sqrt{(\text{hits} + \text{false})(\text{false} + \text{cor})(\text{misses} + \text{cor})(\text{hits} + \text{misses})}}$
Yule's $Q$	Q	$t1/(\text{hits} \cdot \text{correct rejections} + \text{misses} \cdot \text{false alarms})$
Choice-theory measure $\eta$	eta	$\sqrt{\frac{\text{false alarms} \cdot \text{misses}}{\text{hits} \cdot \text{correct rejections}}}$
Proportion correct	PC	$\frac{\text{hits} + \text{correct rejections}}{\text{hits} + \text{correct rejections} + \text{misses} + \text{false alarms}}$

Note—Let  $w$  = weight for weighted  $\kappa$  (default is .5 for Cohen's  $\kappa$ ),  $t1$  =  $\text{hits} \cdot \text{correct rejections} - \text{misses} \cdot \text{false alarms}$ , and  $t2$  =  $w(\text{hits} + \text{false alarms})(\text{false alarms} + \text{correct rejections}) + (1 - w)(\text{misses} + \text{correct rejections})(\text{hits} + \text{misses})$ . Equations are taken from Kraemer (2006), Stanislaw and Todorov (1999), and Swets (1996). If a flattening constant is used, it is added to each cell before these formulas are used.  $z_x$  is the normal deviate for  $x$ . So,  $z_{.975}$  is 1.96.  $\text{sign}$  gives  $-1$  for negative values,  $1$  for positive values, and  $0$  if  $0$ . There is a typo in Equation 5 of Stanislaw and Todorov for  $\ln \beta$ .

be entered. If the variables have different lengths, an error is returned and the function does not run.

The data from Horry and Wright's (2008) study of memory for faces of white and black people for 45 white participants will be used to illustrate this function. The hits, false alarms, misses, and correct rejections for the white faces are stored in `face4w`, which is downloaded with the functions, and the individual variables can be accessed as `face4w$hits`, `face4w$fa`, `face4w$miss`, and `face4w$cr` (or by using the `attach` or `with` functions; see Crawley, 2007, for details on using these functions). The following are the first three lines of these data:

```
face4w[1:3,]
  subno hits fa misses cr
1     5   41  8      9  22
2     6   36  6     14  24
3     7   36  6     14  24
```

The default analysis can be run with

```
stw <- sdt(face4w$hits, face4w$fa,
           face4w$miss, face4w$cr)
```

To input the variables in a different order, you must assign them to their named slots:

```
sdt(fas=face4w$fa, misses=face4w$miss,
    cr=face4w$cr, hits=face4w$hits)
```

The function returns a  $45 \times 15$  matrix. The 45 is for the number of participants in this study, and the 15 is for all of the statistics listed in Table 2. The statistics for the first 3 participants in the file can be seen in Figure 1. The function returns all of the statistics because "all" is the default value for the option `meas`. If you wanted only, for example,  $d'$  and  $A'$ , you would set `meas=c("d", "A")` [or `meas=c('d', 'A')`] in the command line, and a  $45 \times 2$  matrix would be returned. If the output is assigned to an object, such as

```
stw <- sdt(face4w$hits, face4w$fa,
           face4w$miss, face4w$cr,
           meas=c("d", "A"))
```

then these values can be used in combination with the original data for analyses either by calling the individual statistics, as in `stw$d`, or by combining these statistics with the original data file [i.e., `cbind(face4w, stw)`]. Doing this would be appropriate for a between-subjects design. Because Horry and Wright used a within-subjects design, comparing white participants' memories for white and black faces, they reran the analyses with the

stw[1:3, ]					
	HR	FAR	d	csdt	A
1	0.82	0.2666667	1.538291	-0.1462197	0.8573356
2	0.72	0.2000000	1.424463	0.1293899	0.8430556
3	0.72	0.2000000	1.424463	0.1293899	0.8430556
	B	lnbeta	beta	OR	lnOR
1	-0.1397487	-0.2249284	0.7985734	12.52778	2.527948
2	0.1150442	0.1843110	1.2023898	10.28571	2.330756
3	0.1150442	0.1843110	1.2023898	10.28571	2.330756
	kappa	phi	Q	eta	PC
1	0.5496689	0.5498618	0.8521561	0.2825290	0.7875
2	0.4936709	0.5041184	0.8227848	0.3118048	0.7500
3	0.4936709	0.5041184	0.8227848	0.3118048	0.7500

Figure 1. The 15 statistics output by `sdt` for the first 3 participants in Horry and Wright (2008).

data for the memory of black faces (stored in `face4b` and with the output object named `stb`) and ran statistical tests comparing these. For example, to compare the  $A'$  values:

```
t.test(stw$A, stb$A, paired)
```

This  $t$  test shows a significant difference, with the mean being higher for white than for black faces [ $M_{\text{white}} = .90$  and  $M_{\text{black}} = .86$ ;  $t(44) = 4.57$ ,  $p < .001$ ]. Running the same statistics on  $d'$  will produce an error because of infinite values. This error can be addressed with the `flat` and `bound` options of `sdt`, which will be discussed shortly.

By default, `sdt` produces no output directly to the screen. If `pmeans` is set to `TRUE`, the means and standard deviations for the sample are reported. For example:

```
stw <- sdt(face4w$hits, face4w$fa,
  face4w$miss, face4w$scr,
  meas=c("d", "A"), pmeans=TRUE)
```

```
[1] The means:
```

```
  d      A
Inf 0.903
```

```
[1] The standard deviations:
```

```
  d      A
NaN 0.0496
```

One of the  $d'$  values is infinite, because 1 participant had no false alarms. This makes the mean infinite and the standard deviation undefined. Options that can be used in this situation will be described below.

All of the optional arguments for `sdt` are shown in Table 3. Some deserve more explanation. The `flat` option allows you to add a value to the counts for hits, false alarms, misses, and correct rejections. This is usually called a *flattening constant* and is useful for preventing infinite values for some statistics. There are arguments for and against using a flattening constant. When used, .5 is the norm, but the choice can make a difference. Because the flattening constant is added to all cells, as the flattening constant increases, the different statistics should all move toward their “no association” values. If you do not use a flattening constant, you may get values of `Inf` and `-Inf` for some of the statistics, as was shown previously.

Table 3  
The Optional Arguments for the Function `sdt`

Option	Description	Default	Notes
<code>flat</code>	flattening constant	0	Adds the value <code>flat</code> to each cell. For some statistics, this is necessary to avoid infinite values.
<code>trim</code>	trimmed mean	0	Allows trimmed means to be used. Default is the ordinary mean; <code>trim=.5</code> produces medians.
<code>pmeans</code>	print means	FALSE	Prints the means (or trimmed means) and standard deviations for each of the statistics.
<code>meas</code>	which statistics	"all"	Those listed in Table 2. For multiple statistics, enclose them in a list, as in <code>meas=c("kappa", "eta")</code> .
<code>wk</code>	weighting for $\kappa$	.5	The weighting for $\kappa$ . Defaults to Cohen's $\kappa$ , where the two types of errors have equal weighting.
<code>runboot</code>	run bootstrap	FALSE	Runs bootstraps on the means (or trimmed means) using <code>boot</code> (Canty & Ripley, 2008), which must be installed and attached. The function reports BCa nonparametric intervals if <code>pmeans</code> is <code>TRUE</code> .
<code>R</code>	# replications	2000	Number of replications for bootstraps. Efron and Tibshirani (1993) suggested 2,000 for confidence intervals.
<code>confl</code>	confidence level	.95	The confidence interval level for bootstraps. Values above 1 produce an error.
<code>other</code>	if user's statistic is present	FALSE	If a user-defined statistic is to be reported.
<code>newst</code>	user's statistic	nil	The user's statistic. If no statistic is entered and <code>other=TRUE</code> , an error occurs.
<code>bound</code>	bounding statistics	FALSE	Sets infinite values to the nearest finite value.

An alternative for dealing with infinite values is to set `bound=TRUE`. Doing this changes infinite values for individuals to the nearest finite value within the sample. If there are no finite values, then warnings are produced. It is worth stressing that many statistical procedures will work in R, even with infinite values for some individuals in the sample. For example, trimmed means and the Wilcoxon procedures can be used to compare  $d'$  values when some values are infinite.

The option for having trimmed means is included to make the sample statistics more robust. Other robust estimators could have been used, but the simplicity of trimming makes it an attractive option. Wilcox (2004) suggested a .20 trim (so, set `trim=.2` to follow Wilcox's advice). The weighting value for  $\kappa$  can vary by setting `wk` to values between 0 and 1, inclusive. Use values near 0 if you are more worried about false alarms than about misses, and use values near 1 if you are more worried about misses than about false alarms. The default, .5, gives these errors equal weight (what Cohen [1960] proposed). Kraemer (2006) recommended using  $\kappa$  and provided further details about it.

If `pmeans` is `TRUE`, setting `runboot` to `TRUE` will cause nonparametric bias-corrected and accelerated confidence intervals, which are the type of bootstrap interval that was recommended by Efron and Tibshirani (1993), to be reported. Bootstrapping is done for the sample statistics, not for the statistics for each individual. The package `boot` (Canty & Ripley, 2008) is used for bootstrapping. The default confidence level is 95%, but this can be controlled by setting `confl` to values between 0 and 1.

The final two options allow users to include their own functions. The functions must have four arguments in the order listed below. Consider the following example. Suppose you wanted to report Hamann's coefficient. An R function for this is

```
HC <- function(hits,fas,misses,cr)
  return((hits+cr-fas-misses)/
    (hits+cr+fas+misses))
```

When you call the `sdt` function, you would include `other=TRUE` and `newst=c(HC)`. The output includes this statistic in its final column, labeled `newstat`. If you have multiple user-defined statistics, then include these within `c()`. The flattening constant and bounding, if applicable, are automatically applied to any user-defined statistic entered.

For example, to calculate the 90% BCa confidence intervals for the 20% trimmed means of  $d'$  and Hamann's coefficient, both with infinite values bounded, we would write

```
stw <- sdt(face4w$hits,face4w$fa,
  face4w$miss,face4w$cr,bound=TRUE,
  meas=c("d"),pmeans=TRUE,trim=.2,
  runboot=TRUE,confl=.90,
  other=TRUE,newst=c(HC))
```

The result printed to screen is

```
[1] The means:
      d newstat
2.155    0.661
```

```
[1] The standard deviations:
      d newstat
0.703    0.158
[1] **** BCa intervals for d ****
Conf      lb      ub
0.900 2.054 2.388
[1] **** BCa intervals for newstat ****
Conf      lb      ub
0.9000 0.6194 0.6950
```

As was said previously, if a user had data in the form of one trial, only wanted to use a single command, and for some reason did not want to create a new data frame, then he or she could run the function with one trial per line data as (assuming `cd` is a data frame with the appropriate structure)

```
sdt(
  format2to4(cd[,1],cd[,2],cd[,3])[,2],
  format2to4(cd[,1],cd[,2],cd[,3])[,3],
  format2to4(cd[,1],cd[,2],cd[,3])[,4],
  format2to4(cd[,1],cd[,2],cd[,3])[,5])
```

We assume that most users, however, would run these in separate steps.

### Multilevel SDT Function: `mlmsdt`

Several authors have argued for the use of multilevel modeling in general (e.g., Goldstein, 2003), for psychology (e.g., Baayen et al., 2008; Blozis & Traxler, 2007; Hoffman & Rovine, 2007; Locker, Hoffman, & Bovaird, 2007; Wright & London, in press), and for memory research (Wright, 1998). Rather than thinking about data in terms of hits, false alarms, misses, and correct rejections, as you might for the traditional approach, you need to think about predicting what the participant says (i.e., in memory recognition terms, whether the participant says "old" or "new") from the true state of the object (i.e., in memory recognition terms, whether the item is old or new). The model involves predicting the probability of saying "old" from whether the item is old and determining whether any other variables moderate this effect. DeCarlo (1998) described how the generalized linear model can be used as an alternative to traditional SDT. If the default codings for the function are used, the coefficient of a probit regression will be  $d'$  from the Gaussian/normal SDT model, and the coefficient of a logistic regression will be the  $\ln OR$  from the logistic SDT. This is illustrated below.

The advantage of the GLM framework is that it allows the user to include other variables. This is important with SDT because often there are covariates that can take different values for each trial (e.g., RT). If the covariates are binary, like the race of the face in Horry and Wright (2008), traditional methods can be used by calculating measures such as  $d'$  for both conditions. However, if the covariate is continuous, this is more problematic. Within this GLM framework, this is not a problem.

The function we present for conducting the multilevel modeling approach to SDT data is called `mlmsdt`. It uses a function called `lmer`, which is part of the `lme4` package written by Bates, Maechler, and Dai (2008). Bates and colleagues' package builds on `nlme` (Pinheiro & Bates, 2000). People who are experienced with `lmer` may find it easier



**A**

```
[1] Overall estimated lnOR
      Estimate   Std. Error   z value   Pr(>|z|)
0.6286087    0.4071334    1.5439871  0.1225915
```

**B**

```
mlmsdt(s53$subno,s53$isold,s53$sayold,lk="probit")
[1] Overall estimated d'
      Estimate   Std. Error   z value   Pr(>|z|)
0.3925100    0.2533964    1.5489961  0.1213827
```

**Figure 2.** Output for Horry and Wright's (2008) data using (A) `sdt` and (B) `mlmsdt`.

to run a series of multilevel generalized linear regressions directly with `lmer`. With our function, the user enters variables for the participant number, the true state of the item, what the participant says, and then a `data.frame` of covariates. If there are no additional covariates, this can be left blank. The covariates can be numeric, unordered, or ordered factors, and the function treats them accordingly. If the items shown to people are identical and there is a variable for the items, a random variable for item can be included. If the data are counts of hits, false alarms, misses, and correct rejections, the format may be changed using the function `format4to2`, which was discussed previously.

Most of the time, users will have multiple participants and at least one covariate, but the function can be used with only a single participant and no covariates. Although in practice this situation will be rare, it will be useful for illustrative purposes for us to show the equivalence of the traditional and multilevel approaches to SDT for simple problems. When there is only a single participant, the function calls another function—written for the present article—called `singlelevel`. The printed output is nearly identical to that when multiple participants are included. The function uses the `glm` function rather than the `lmer` function, so there are computational differences.

The data used to illustrate this function come from Skagerberg and Wright's (2008) study of memory suggestibility and interpersonal power. They had three groups. We will start by looking at their control group. These participants saw 50 faces and then were shown these faces again, in addition to 50 new faces, and were asked to say which had been previously shown. The data are stored in `confcontr`, which is downloaded automatically when you access these functions.

The control condition is the simplest type of memory recognition design; for this group, there were no covariates. In total, there were 31 participants in the control condition. For illustration, we will begin by examining just 1 participant in this file (#53). The code

```
s53 <- confcontr[confcontr$subno==53,]
creates an active data frame with just this participant's data. Here are the data for the first three trials for this participant:
```

```
s53[1:3,]
  subno sayold isold
1   53      1      0
2   53      1      1
3   53      1      1
```

When the command

```
mlmsdt(s53$subno,s53$isold,s53$sayold)
```

is run, the function finds that there is only a single participant number. On the screen, it tells the user that it appears there is only a single participant.

```
[1] You appear to have only a single
    participant,
[1] so the function singlelevel will
    be used.
```

These variables are sent to the `singlelevel` function. There, the default logistic regression is run with the `glm` function. The output to the screen is shown in Figure 2A. This output tells the user that the  $\ln OR$  is 0.6286, and it provides the standard error,  $z$ , and  $p$  values. This (the log-odds-ratio) is the default measure of accuracy for the function. A `glm.object` is returned, which can be used in other functions. To get the function to calculate  $d'$ , use the option `lk="probit"`, as shown in Figure 2B. The resulting value is  $d' = 0.3925$ . Other options are described later in the text. To verify that the estimates of  $\ln OR$  and  $d'$  are the same as those found with the traditional approach, the data can be formatted using the `format2to4` function, and the resulting values can be entered into the `sdt` function.

```
cont53 <- format2to4(s53$subno,
  s53$isold,s53$sayold)
sdt(cont53[2],cont53[3],cont53[4],
  cont53[5],meas=c("d","lnOR"))
      d      lnOR
1 0.3925100 0.6286087
```

Usually users would have multiple participants. Here the function is used with all of the data from Skagerberg and Wright's (2008) control group, as shown in Figure 3. The estimated population  $\ln OR$  is 1.727. If you want a probit regression, set `lk="probit"`, and the resulting

```

mod1 <- mlmsdt(confcontr$subno, confcontr$isold, confcontr$sayold)
[1] Overall estimated lnOR
isold
1.727

      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.2184654  0.06973876 -3.132625 1.732506e-03
isold        1.7272227  0.13573596 12.724871 4.301830e-37

```

**Figure 3.** Output from `mlmsdt` showing the log-odds-ratio for the control group of Skagerberg and Wright (2008).

estimate for  $d' = 1.061$ . The default is to allow random variation around both the intercept and the coefficient for accuracy. If you only want a random intercept, then set `vardiff=FALSE`. If a model is not being estimated because of computational problems, this option is worth using because it simplifies the computation. If you rerun the previous analyses with this option, the resulting estimates are  $\ln OR = 1.692$  and  $d' = 1.040$ . If these data are reformatted and entered into `sdt`, the sample means from that function are  $\ln OR = 1.79$  and  $d' = 1.09$ . Since all of these values are estimates of the population means, it is reassuring that they are similar. However, because they are using different methods to take into account variability among participants, some differences are expected.

Skagerberg and Wright (2008) had two other groups of participants, which we will now consider. Participants took part in a power manipulation task after viewing the original 50 faces. In the low-power condition, participants had to design a restaurant and have it critically evaluated. Those in the high-power condition did the evaluation. Participants in these two conditions were paired together and took part in a joint recognition task in which for the first 50 trials, one person responded first, and then the second person responded. Then the order was switched for the final 50 trials (see Wright, Mathews, & Skagerberg, 2005, for details of the procedure). The participants could say either “old” or “new.” The data are stored in `confexp`, which is downloaded automatically when you access these functions.

One practical advantage of `mlmsdt` over `sdt` occurs when there are covariates that vary trial by trial. Although methods like those used previously for Horry and Wright’s (2008) data can be used when the covariates are categorical, when they are continuous, this method becomes too cumbersome to be of use. Skagerberg and Wright’s (2008) study had two additional covariates. A between-subjects variable was whether participants were assigned to the low-power or to the high-power condition. A within-subjects variable was what the other participant in the pair said for each trial. In the present article, we let this variable have three levels: whether the other person in the pair said the face was new, whether the participant was responding first in the pair, and whether the other participant in the pair said the face was old. The variable can be defined as either as `.ordered` or as `.factor`; the only difference between these is the contrasts that are used for them in the regression. The contrasts can be over-

ridden with the `contrasts` function in R: The default for `ordered` is a series of polynomial contrasts, and the default for `factor` is to use the first category as a reference category. In the present article, it will be treated as an unordered factor. Skagerberg and Wright’s interest was how much the participant was affected by what the other person said.

The proportion of “old” responses in each condition is shown in Table 4. Some effects are clear from this table. For example, old items are reported “old” more often than are new items (overall, 62% vs. 19%), showing that there is memory. Additionally, there are more “old” responses as you move from left to right along the table (overall, 26%, 41%, and 63%), showing that there is memory conformity. The remaining effects are more difficult to gauge from this table because of individual variation.

The default for `mlmsdt` is to run a multilevel logistic regression. It reports the estimated population  $\ln OR$ , the estimated  $\ln OR$  for each level of each categorical variable, and the change in  $\ln OR$  associated with a change of 1 standard deviation for each numeric variable. The covariates need to be entered as a `data.frame`, as they are below, unless there is just a single covariate; in which case, the variable name suffices. The default modifies the `isold` variable so that it is coded  $-.5$  and  $+.5$ , and it standardizes all numeric covariates. In some circumstances, the user may want to turn this off (set `modify=FALSE`). The default is to look at only the main effects of the covariates and their interactions with `isold`. The interactions among all the covariates can be included in the final model by setting `int=TRUE`. For models of intermediate complexity, the user will need to access the `lmer` function directly (this is shown below). Setting `other` as an ordinal variable will make the coefficients in the regression equation different, but otherwise the models will be the same. We begin by attaching the data set. We do this rather than call it within each command because we will call these variables frequently.

**Table 4**  
The Proportion of “Old” Responses for the Different Conditions in Skagerberg and Wright (2008)

	Other Says “New”	Responds First	Other Says “Old”
New items–low power	.18	.20	.39
Old items–low power	.50	.59	.70
New items–high power	.12	.18	.36
Old items–high power	.49	.65	.72

**A**

```
[1] Overall estimated lnOR
isold
2.098
```

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.5206037	0.06937082	-7.50465	6.159319e-14
isold	2.0980987	0.12695545	16.52626	2.374286e-61

**B**

```
[1] The estimated lnOR values for the groups are:
```

	Estimate	Std. Error	z value	Pr(> z )
cov1	-0.5041015	0.09796694	-5.145629	2.666261e-07
cov2	-0.5397816	0.09885855	-5.460141	4.757564e-08
cov1:isold	1.9323919	0.17549731	11.010949	3.384202e-28
cov2:isold	2.2693244	0.17749401	12.785358	1.979502e-37

**C**

```
[1] The estimated lnOR values for the groups are:
```

	Estimate	Std. Error	z value	Pr(> z )
cov-1	-0.9385945	0.08755942	-10.719515	8.243466e-27
cov0	-0.5401737	0.07557875	-7.147164	8.858883e-13
cov1	0.1670713	0.09980507	1.673976	9.413526e-02
cov-1:isold	1.8870443	0.17154241	11.000454	3.802142e-28
cov0:isold	2.0893947	0.14698408	14.215109	7.383643e-46
cov1:isold	1.5056586	0.19647509	7.663356	1.811360e-14

**Figure 4.** The coefficients from `mlmsdt` for Skagerberg and Wright (2008). (A) Estimate for the overall log-odds-ratio. (B) Estimates for experimental groups. (C) Estimates for whether the other person says “old,” “new,” or nothing.

```
attach(confexp)
task <- as.factor(task)
other <- as.factor(other)
mod1 <- mlmsdt(partno,old,sayold,
  data.frame(task,other))
```

Much output is produced, including the three sets of coefficients shown in Figure 4. Figure 4A shows an overall lnOR of 2.10. Figure 4B shows that the lnOR for the low-power condition, 1.93, is slightly smaller than that for the high-power condition, 2.27. The difference between these is not statistically significant (which is also included in the output [ $\chi^2(1) = 1.78, p = .18$ ], but this output is not shown here, to save space). Figure 4C shows that the lnOR = 1.89 when the other person says “new,” that the lnOR = 2.09 when responding first, and that the lnOR = 1.51 when the other person says “old.” Skagerberg and Wright (2008) were specifically interested in the intercepts. The intercepts show an increase in reporting “old” from when the other person says “new,”

to when the participant responds first, to when the other person responds “old.” This shows the memory conformity effect.

The object returned is the result of the generalized linear multilevel model, which includes the main effects of all the covariates and the interaction between each of these and whether the item is old. It is a `*mer` object, which is an S4 class object (see Chambers, 2008, for details). Although currently most of the R functions do not produce S4 class objects, they are increasingly popular and are used by most of the programmers on the Core Development Team. One practical difference between these and other objects is that to access parts of an S4 object, you use the character “@” rather than “\$”. For example, `mod1@fixef` returns the fixed effects and `summary(mod1)@coefs` returns the fixed effects with some of their statistics.

Skagerberg and Wright’s (2008) main hypothesis was that there would be interactions between the power tasks and how much the participant was influenced by the other

	Estimate	Std. Error	z value	Pr(> z )
task2:other0	0.369102002	0.1596715	2.31163307	2.079791e-02
task2:other1	0.300991897	0.2084633	1.44386010	1.487783e-01

**Figure 5.** Coefficient estimates for the interaction between task and what the other person said for the data from Skagerberg and Wright (2008).



**Table 5**  
**The Optional Arguments for the Function `mlmsdt`**

Option	Description	Default	Notes
<code>covs</code>	covariates	NULL	Indicates whether any covariates, other than whether the item is old or new, are included. If there is only one, then just write the variable name. If there is more than one, it must be a <code>data.frame</code> . The variables should be defined as numeric, ordered, or factor before being entered.
<code>modify</code>	standardizes interval variables	TRUE	Standardizes numeric variables so that they have a mean of 0 and standard deviation of 1.
<code>vardiff</code>	random accuracy	TRUE	Assumes that people vary in their diagnosticity by allowing the coefficient for <code>isold</code> to be a random, normally distributed variable.
<code>lk</code>	link function	"logit"	The link function. Usually either "logit" or "probit", but it will also accept "cauchit", "log", and "cloglog". See help for family in R.
<code>item</code>	allows item variation	NULL	Takes into account the variability among items. Requires that you have a variable for the different stimuli with the same numbers across participants.

person. To include interactions among covariates, set `int=TRUE`. Here this would be

```
mod2 <- mlmsdt(partno,old,sayold,
  data.frame(task,other),int=TRUE)
```

The output to the screen is the same as before, but the object returned includes the interaction between the two covariates and the three-way interaction between all the predictor variables. The coefficients for this model can be examined with `summary(mod2)$coefs`. The coefficients for the interaction of `task:other` are shown in Figure 5, and suggest that this interaction deserves further investigation.

When there are several covariates, it is likely that the researchers will have many models of interest. The models will depend on the particular hypotheses and may be idiosyncratic. We chose the Skagerberg and Wright (2008) study specifically because their needs were different from those of the authors in most studies, and their study allowed us to illustrate using the `lmer` function directly. The `task:other` interaction effect can be measured by comparing different pairs of models, and for these data, they lead to the same conclusions. For example, we can compare the model with just main effects for `task`, `other`, and `old` with the model that includes these plus the interaction between `task` and `other`, as shown in Figure 6. The result is a statistically significant difference [ $\chi^2(2) = 6.07, p = .05$ ] in the predicted direction, with the effect of `other` being larger for those in the low-power condition than in the high-power condition.

There are fewer options for `mlmsdt` than for `sdt`. Table 5 lists these. One that deserves further discussion is `item`. In traditional SDT analysis, each item is treated as independent and as if drawn from a much larger population of items. The focus is not on item differences unless they correspond to one of the experimental factors (e.g., race of the face; Horry & Wright, 2008). In many studies, however, we would expect differences among the items. Modeling variation among items can be done within `mlmsdt` by changing the option `item` (see Baayen et al., 2008, for using this approach for linguistic data; see Doran, Bates, Bliese, & Dowling, 2007, in relation to educational testing). If `item` is set to the name of the variable corresponding to item number, then what is called a *cross-level model* is calculated with random variation both for participants and for items. For example, we used this model with a recent study on social anxiety and memory suggestibility (Wright, London, & Waechter, in press). We wanted to see whether the fear of negative evaluation (FNE) relates to whether somebody conforms to another person. These data are downloaded with the function, but need to be attached:

```
attach(fne)
fne1 <- mlmsdt(id,isold,sayold,
  data.frame(FNE,other),
  int=T,item=face)
```

The command `summary(fne1)` produces much output from this complex model, including the random effects displayed in Figure 7A. These effects show that there is

```
m1 <- lmer(sayold~task+other+old+(old|partno),family=binomial)
m2 <- lmer(sayold~task*other+old+(old|partno),family=binomial)
anova(m1,m2)
m1: sayold ~ task + other + old + (old | partno)
m2: sayold ~ task * other + old + (old | partno)
      Df      AIC      BIC    logLik   Chisq  Chi Df  Pr(>Chisq)
m1     8   5739.5   5792.0   -2861.8
m2    10   5737.4   5803.0   -2858.7   6.0701     2    0.04807 *
```

**Figure 6.** The models evaluated, using the `lmer` function, for Wright et al.'s (in press) data and the associated fit statistics.

**A**

Random effects:

Groups	Name	Variance	Std. Dev.	Corr
item	(Intercept)	0.46856	0.68451	
subno	(Intercept)	0.05208	0.22821	
	isold	0.23812	0.48797	-0.147

**B**

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.56129	0.20702	-2.711	0.00670 **
isold	1.23565	0.42262	2.924	0.00346 **
FNE	-0.11486	0.08136	-1.412	0.15800
other	0.16874	0.24357	0.693	0.48845
FNE:other	0.40091	0.10187	3.935	8.3e-05 ***
isold:FNE	0.16705	0.16645	1.004	0.31556
isold:other	-0.05216	0.48720	-0.107	0.91474
isold:FNE:other	-0.13348	0.20377	-0.655	0.51245

**Figure 7. Output for (A) random effects and (B) fixed effects for Wright et al.'s (in press) data on memory conformity and fear of negative evaluation.**

much variability among the different stimuli (faces were used). The fixed effects are also included in the summary, and can be seen in Figure 7B. The critical effect was the interaction between FNE and other, which is statistically significant ( $z = 3.94$ ,  $p < .001$ ). It is likely that anyone with an example this complex would want to access the `lmer` function directly, because there are likely several additional effects that would be explored.

In addition to the statistical functions described previously, the `example` and `help` functions [e.g., `example(sdt)` and `help(sdt)`] produce examples and explanations of the functions.

### Summary

Methods from SDT are often used in psychology to describe the association in a  $2 \times 2$  cross-tabulation, such as Table 1. There are several measures for the association, each with its own set of assumptions, and there is much disagreement among methodologists on which to use. Therefore, the `sdt` function allows the user to choose from 15 measures, and users may include their own statistics. Most methodologists do agree that confidence intervals and robust measures are often useful (Wilkinson et al., 1999), and `sdt` reports these. The intervals are the bias-corrected and accelerated bootstrap estimates (Efron & Tibshirani, 1993). These were used rather than asymptotic methods because the shapes of the distributions will vary, and because there are no agreed-on methods for asymptotic intervals for some of the statistics in Table 2.

The traditional method within SDT involves calculating a value of the association for each individual and then running analyses on these aggregate values. The alternative presented in `mlmsdt` consists of multilevel general-

ized linear models. Our function uses Bates et al.'s (2008) `lme4` package and, in particular, their `lmer` function. These functions allow covariates at the trial level to be included in the analyses. The functions are illustrated using recent data. The data sets, functions, and further examples are available from CRAN. They can be downloaded directly from R with

```
install.packages("sdtalt")
library(sdtalt)
```

### AUTHOR NOTE

An R package for the functions described in the present article, along with a manual, are available at <http://cran.r-project.org/web/packages/sdtalt/index.html>. Thanks to Douglas Creelman and Michael Lee for comments on the manuscript. Address correspondence to D. B. Wright, Psychology Department, Florida International University, Miami, FL 33265 (e-mail: [dwright@fiu.edu](mailto:dwright@fiu.edu)).

### REFERENCES

- BAAYEN, R. H., DAVIDSON, D. J., & BATES, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory & Language*, *59*, 390-412.
- BANKS, W. P. (1970). Signal detection theory and human memory. *Psychological Bulletin*, *74*, 81-99.
- BATES, D., MAECHLER, M., & DAI, B. (2008). *lme4: Linear mixed-effects models using Eigen and Eigen* (Version 0.999375-20) [Computer software]. Available at <http://cran.r-project.org/>.
- BLOZIS, S. A., & TRAXLER, M. J. (2007). Analyzing individual differences in sentence processing performance using multilevel models. *Behavior Research Methods*, *39*, 31-38.
- CANTY, A., & RIPLEY, B. (2008). *boot: Bootstrap R (S-Plus) Functions* (R package Version 1.2-32) [Computer software]. Available at <http://cran.r-project.org/>.
- CHAMBERS, J. M. (2008). *Software for data analysis: Programming with R*. New York: Springer.

- COHEN, J. (1960). A coefficient of agreement for nominal scales. *Educational & Psychological Measurement*, **20**, 37-46.
- CRAWLEY, M. J. (2007). *The R book*. Chichester, U.K.: Wiley.
- DECARLO, L. T. (1998). Signal detection theory and generalized linear models. *Psychological Methods*, **3**, 186-205.
- DORAN, H., BATES, D., BLIESE, P., & DOWLING, M. (2007). Estimating the multilevel Rasch model: With the lme4 package. *Journal of Statistical Software*, **20** (2).
- EFRON, B., & TIBSHIRANI, R. J. (1993). *An introduction to the bootstrap*. Boca Raton, FL: Chapman & Hall/CRC.
- GOLDSTEIN, H. (2003). *Multilevel statistical methods* (3rd ed.). London: Edward Arnold.
- GOODMAN, L. A., & KRUSKAL, W. H. (1979). *Measures of association for cross classifications*. New York: Springer.
- HOFFMAN, L., & ROVINE, M. J. (2007). Multilevel models for the experimental psychologist: Foundations and illustrative examples. *Behavior Research Methods*, **39**, 101-117.
- HORRY, R., & WRIGHT, D. B. (2008). I know your face but not where I saw you: Context memory is impaired for other race faces. *Psychonomic Bulletin & Review*, **15**, 610-614.
- KRAEMER, H. C. (2006). Correlation coefficients in medical research: From product moment correlation to the odds ratio. *Statistical Methods in Medical Research*, **15**, 525-545.
- LOCKER, L., HOFFMAN, L., & BOVAIRD, J. A. (2007). On the use of multilevel modeling as an alternative to item analysis in psycholinguistic research. *Behavior Research Methods*, **39**, 723-730.
- MCCULLAGH, P., & NELDER, J. A. (1989). *Generalized linear models* (2nd ed.). London: Chapman and Hall.
- PINHEIRO, J. C., & BATES, D. M. (2000). *Mixed-effects models in S and S-Plus*. New York: Springer.
- R DEVELOPMENT CORE TEAM (2008). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. Available at [www.R-project.org](http://www.R-project.org).
- SKAGERBERG, E. M., & WRIGHT, D. B. (2008). Manipulating power can affect memory conformity. *Applied Cognitive Psychology*, **22**, 207-216.
- STANISLAW, H., & TODOROV, N. (1999). Calculation of signal detection theory measures. *Behavior Research Methods, Instruments, & Computers*, **31**, 137-149.
- SWETS, J. A. (1996). *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Mahwah, NJ: Erlbaum.
- WICKENS, T. D. (2002). *Elementary signal detection theory*. New York: Oxford University Press.
- WILCOX, R. R. (2004). *Introduction to robust estimation and hypothesis testing* (2nd ed.). San Diego: Academic Press.
- WILKINSON, L. & THE TASK FORCE ON STATISTICAL INFERENCE, AMERICAN PSYCHOLOGICAL ASSOCIATION, SCIENCE DIRECTORATE, WASHINGTON DC, US. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, **54**, 594-604.
- WRIGHT, D. B. (1998). Modelling clustered data in autobiographical memory research: The multilevel approach. *Applied Cognitive Psychology*, **12**, 339-357.
- WRIGHT, D. B., GABBERT, F., MEMON, A., & LONDON, K. (2008). Changing the criterion for memory conformity in free recall and recognition. *Memory*, **16**, 137-148.
- WRIGHT, D. B., & LONDON, K. (2009). *Modern regression techniques using R: A practical guide for students and researchers*. London: Sage.
- WRIGHT, D. B., & LONDON, K. (in press). Multilevel modelling: Beyond the basic applications. *British Journal of Mathematical and Statistical Psychology*.
- WRIGHT, D. B., LONDON, K., & WAECHTER, M. (in press). Social recognition memory and social anxiety. *Applied Cognitive Psychology*.
- WRIGHT, D. B., MATHEWS, S. A., & SKAGERBERG, E. M. (2005). Social recognition memory: The effect of other people's responses for previously seen and unseen items. *Journal of Experimental Psychology: Applied*, **11**, 200-209.
- ZHOU, X.-H., OBUCHOWSKI, N. A., & MCCLISH, D. K. (2002). *Statistical methods in diagnostic medicine*. New York: Wiley.

## NOTES

1. Graded responses and multiple-choice formats can also be incorporated within the SDT framework. Wickens (2002) provided good coverage of these. These situations are not included in the functions in the present article.

2. Typical reasons why a model fails to converge include having high collinearity, having only a small number of individuals, having only a small number of trials per individual, and including too many coefficients in the model.

(Manuscript received March 22, 2008;  
revision accepted for publication September 29, 2008.)