An extension of within-subject confidence intervals to models with crossed random effects

Stephen Politzer-Ahles[1]

[1]*Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong; ORCID: 0000-0002-5474-7930*

*Address correspondence to

Stephen Politzer-Ahles

Department of Chinese & Bilingual Studies

The Hong Kong Polytechnic University

Hung Hom, Kowloon

Hong Kong

E-mail: stephen.politzerahles@polyu.edu.hk

Abstract

A common problem in displaying within-subject data is that of how to show confidence intervals that accurately reflect the pattern of significant differences between conditions. The Cousineau-Morey method (Cousineau, 2005; Morey, 2008) is a widely used solution to this issue; however, this method only applies to experimental designs with only one repeated-measures factor (e.g., subjects). Many experimental designs in fields such as psycholinguistics and social psychology use crossed random effect designs where, e.g., there are repeated measures both for subjects and stimuli. For such designs, extant methods for showing within-subject intervals would require first aggregating over stimuli, and thus such intervals might be a less accurate reflection of the statistical significance patterns if the data are actually analyzed using a method that takes both random effects into account (e.g., linear mixed-effects models). The present paper proposes an extension of the method described above to address this problem; the proposal is to scale the data using a mixed-effects model, rather than using the means from each subject, and then calculate confidence intervals from the data scaled thusly. Analysis of a sample of crossed random effect datasets reveals that intervals calculated using this method give a slightly more accurate reflection of the pattern of statistical significance in the between-condition differences.

Keywords: confidence intervals, repeated measures, linear mixed effects, within-subjects confidence intervals, crossed random effects

1. Introduction

Displaying informative summaries of data is a common challenge in experimental psychology. One useful tool for displaying and interpreting data is the confidence interval (see, e.g., Cumming, 2014), which allows readers to see not just a single parameter of a dataset (e.g., the mean), but also an estimate of what that parameter differs from statistically; compared to a single mean, a confidence interval is more useful for showing what null hypotheses are consistent with that parameter and what null hypotheses can be ruled out.[1]

There is a challenge, however, when it comes to using confidence intervals to represent a pattern of significant differences between conditions. For a design with two conditions to be compared, the only way to accurately show the significance of the between-condition difference is to plot the confidence interval of the difference (which Rouder and Morey, 2005, refer to as a "relational CI"), not the confidence intervals of each condition's mean (which Rouder and Morey refer to as an "arelational CI"; see Blouin & Riopelle, 2005; Cumming & Finch, 2005; Franz & Loftus, 2012). For example, for the data shown in Figure 1, confidence intervals of each condition's mean (panel B) are uninformative with respect to whether the differences between conditions are significant, but confidence intervals of the pairwise differences (panel C) reveal that each pair of conditions is significantly different, as the intervals do not cross zero.

However, plotting each pairwise difference between conditions is not always a feasible way to display data. When plotting the results of an experiment with a large

[1] Confidence intervals are often said to be useful for indicating the precision of a parameter estimate or indicating the plausible values of a parameter, but these inferences are not strictly justified (Morey, Hoekstra, Rouder, Lee, and Wagenmakers, 2016); confidence intervals are related to null hypothesis statistical testing and do not assign likelihoods.

number of conditions, showing all possible pairwise comparisons may be prohibitive in terms of space or complexity. A design with *K* levels has (*K*(*K*-1))/2 pairwise differences, so an experiment with, e.g., 8 conditions has 28 pairwise differences (plus main effects and interactions, in the case of a factorial design) that could be shown. Even with a small number of conditions, standard practice in many fields is still to show each condition mean, rather than each difference.

One such field is psycholinguistics. I reviewed papers published within the last year in two highly-regarded psycholinguistics journals, *Journal of Memory and Language* (N=39) and *Language, Cognition and Neuroscience* (N=46) and found that, of 121 relevant figures and tables that showed any kind of error bar, 92 (76%) showed only individual conditions, not between-condition differences (see Supplementary File 1; 163 figures and tables are listed there, but 42 of these did not show or describe any sort of error bar and thus are not discussed here). Of the 29 figures and tables that did show between-condition differences, 21 of these (72%) were regression tables with coefficients and standard errors. Only 4 papers in the sample directly showed between-condition differences in figures. Nevertheless, practitioners often attempt to use the pattern of confidence interval overlap to make inferences about which pairs of conditions are have significantly different means. For example, one paper made explicit reference to the pattern of confidence interval overlap in the interpretation of between-condition differences (Gould et al., 2016:243, “The means of two conditions are considered statistically different if the mean of one condition is not captured by the 95% CI of the other condition”—which is not strictly correct for the type of confidence interval used in this example), and indeed Belia and colleagues (2005) have provided empirical evidence

that researchers rely (often erroneously) on comparing whether arelational confidence intervals of condition means overlap in order to make conclusions about significant differences.Therefore, what is often desirable is a way to show each condition's mean (or some other relevant sample statistic) along with an interval that allows rough inferences about which conditions it differs from. A reader can then compare separate confidence intervals of two different conditions to see how much they overlap; while this is only a rough heuristic for judging whether the two conditions differ (Cumming & Finch, 2005), it can be a more efficient means of displaying data compared to showing all pairwise differences. Rouder and Morey (2005) likewise recommend the display of individual means along with [arelational] confidence intervals, rather than or in addition to display of the differences, for parsimonious data visualization, while reminding that these need to be supplemented with direct statistical comparisons.

1a. The reason for within-subject intervals

On top of this issue, an additional challenge is raised when it comes to displaying confidence intervals for *within-subjects* designs (for review see Baguley, 2012; Blouin & Rioppele, 2005; Cousineau, 2005; Franz & Loftus, 2012; Loftus & Masson, 2004; Morey, 2008). As Loftus and Masson (1994) demonstrate, when a dataset includes conditions with repeated measures (e.g., when a single participant contributes a data point to each condition in an experiment), then standard confidence intervals around each mean are even less informative for illustrating the pattern of significant differences between conditions. Loftus and Masson describe a memory experiment in which each participant

attempts to recall words in different conditions; while the within-subject differences between conditions are very systematic, there is large between-subject variance in the average number of words recalled across conditions, and thus the standard confidence intervals for the conditions are very wide, overlapping substantially and giving the impression that the between-condition differences are not significant (Figure 1B). While this problem could be resolved by showing the confidence intervals of the between-condition differences instead of the confidence intervals of each condition's mean (as done in Figure 1C), this is rarely done, and in some cases it is not feasible, as described above.

As another way to address this problem of within-subjects data, several methods have been proposed for calculating *within-subjects intervals* (see Baguley, 2012, for review).[2] What these methods have in common is that they adjust the size of a confidence interval such that they only reflect variance in within-subject effects; whether or not two confidence intervals sufficiently overlap then roughly corresponds to whether or not the difference between the associated conditions is nonsignificant. While the intervals produced by these methods are not true confidence intervals, and are subject to Cumming and Finch's (2005) caveats about trying to compare two confidence intervals, they can be useful for making a quick visual summary of the dataset that allows very rough inferences about which pairwise comparisons will probably be significant. One type of within-subject interval (Cousineau-Morey intervals) is shown in Figure 1E.

---

[2]Throughout this paper I refer to these as " intervals" rather than "*confidence* intervals" because, as noted originally by Loftus and Masson (1994), these intervals are not true confidence intervals (i.e., it is not the case that if you repeat the experiment 100 times then 95 of the 95% confidence intervals would contain the population parameter).

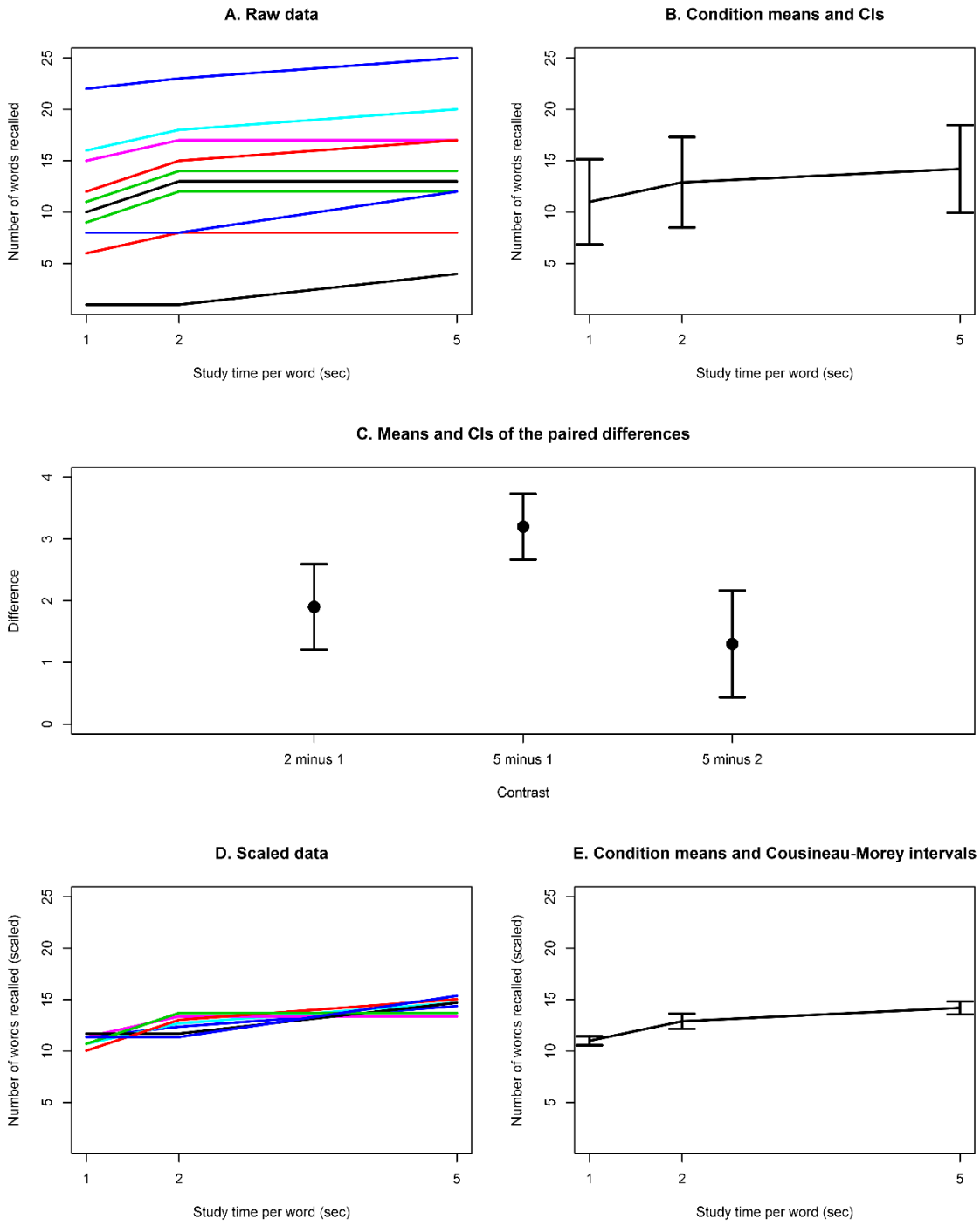


Figure 1. Example of within-subjects intervals. (A) Sample repeated-measures data from Loftus & Masson (1994); each line represents one subject's data in three conditions. (B) Standard confidence intervals calculated across participants for each condition, which

do not accurately reflect that the between-condition differences are statistically significant in a repeated-measures analysis. (C) 95% confidence intervals of the difference between each pair of conditions. (D) The data scaled by subtracting the subject's mean from each data point and adding back the grand mean, which removes between-subject baseline variability and preserves just the pattern of within-subject differences. (E) 95% within-subject intervals calculated using the method described by Cousineau (2005) and updated by Morey (2008), which are narrower and more accurately reflect the significant between-condition differences.

1b. The problem of crossed random effects

The proposed methods, however, have thus far only been implemented for designs with a single random effect, typically Subjects (hence the name "within-subject intervals"). Many research paradigms, however, have multiple random effects. In psycholinguistics, for example, experiments that cross Subjects and Items (or “Stimuli”) are common (Baayen, Davidson, & Bates, 2008; Chang & Lane, 2016; Judd, Westfall, & Kenny, 2012). For instance, a semantic priming experiment might examine whether people respond faster to words presented on a screen (e.g., *DOCTOR*) if the word is preceded by a related word (e.g., *nurse … DOCTOR*) rather than an unrelated word (e.g., *table … DOCTOR*). Usually such an experiment not only contains multiple subjects, but also multiple words. Just as each subject will contribute data points in each condition (i.e., a given subject will complete both Related and Unrelated trials), so will each word (e.g., *DOCTOR* may appear in both Related trials and Unrelated trials). In such an experiment,

both Subjects and Items (words) are random effects with repeated measures. Extant methods for calculating within-subject intervals would require first aggregating over items to get condition means for each subject, or aggregating over subjects to get condition means for each item, and then calculating within-subject intervals (or within-item intervals) across those aggregated means. This, however, would not be an accurate reflection of the actual statistical analysis of these data: for the past decade, the field of psycholinguistics has been moving away from using separate by-subject and by-item analyses, and towards using linear mixed effect models that take into account both subject and item variance simultaneously (Baayen et al., 2008; see also Chang & Lane, 2016). Thus, if a researcher intends to display within-subject intervals to summarize experimental data from a repeated-measures paradigm, it would be ideal to use intervals that reflect the corresponding statistical analysis as accurately as possible (while keeping in mind the limitation that such intervals are still only a rough heuristic and are not true confidence intervals).

This problem is illustrated in Figure 2, which shows simulated data from a semantic priming paradigm like that described above (the data are available in Supplementary File 2). As shown in panel A, the priming effect (the difference between unrelated and related prime conditions) in this dataset is substantially more variable across subjects than it is across items. Therefore, while the effect is in fact close to significant at the α=.01 level (b=-27.04, SE=12.48, t=-2.17, when using a maximal linear mixed-effects model [Barr et al., 2013]), the 99% confidence interval of the by-subject effects nevertheless crosses zero. This problem could be avoided by showing the confidence interval of the coefficient (i.e., the Unrelated-Related difference) from the

mixed effects model; however, as discussed above, researchers often desire to show each condition mean rather than the mean and confidence interval of the pairwise difference. The problem of crossed random effects also affects the plot of condition means: as shown in panel B, the within-subject (solid blue) and within-item (solid black) intervals look very different, with the non-overlapping within-item intervals suggesting a significant difference whereas the substantially overlapping within-subject intervals suggest a non-significant difference. If a researcher created a plot showing only within-subject intervals, the significance pattern suggested by the figure would be inconsistent with the results of inferential statistics.

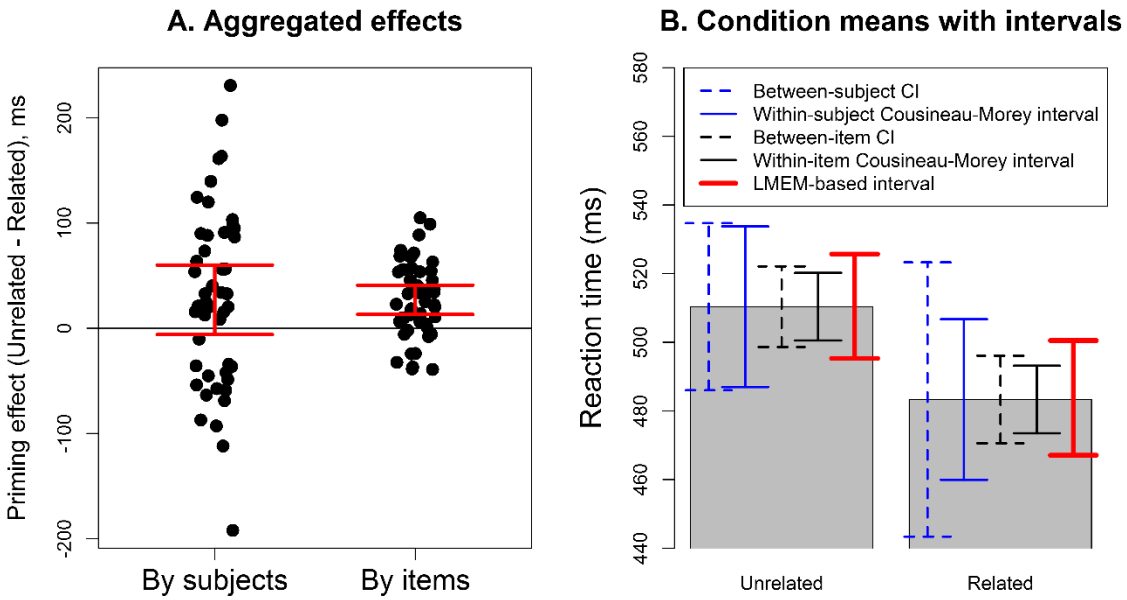


Figure 2. Simulated data from a semantic priming experiment with a design that is both within subjects and within items. (A) Pairwise differences (Unrelated minus Related) for each subject and for each item (one dot represents one subject or one item); error bars show the standard 99% confidence interval of the mean of the differences. (B) Condition

means shown together with error bars depicting several types of interval at the 99% confidence level.

Previous methods for calculating within-subject intervals (e.g., the Cousineau-Morey method and the Loftus & Masson method) were never intended to be used in designs with crossed random effects; therefore, their inapplicability to the sorts of data described above is not a bug, but a feature. Nonetheless, such designs are common in cognitive psychology. Therefore, new methods that allow practitioners to visualize patterns of significant differences, in a similar way as allowed by within-subjects intervals, would be of value.

2. A solution: linear mixed effect model residuals

One solution to the problem of crossed random effects is to scale the dependent variable using a linear mixed effects model. To motivate this solution, it is informative to compare it to the scaling method used to calculate Cousineau-Morey intervals (Cousineau, 2005; Morey, 2008), illustrated in Figure 1D.

Deriving Cousineau-Morey intervals involves the following steps:

1. Scale the data to remove the between-subject baseline differences. This is done by calculating the mean of the dependent variable (e.g., number of words recalled in Figure 1, or reaction time in the semantic priming example) for each participant. The relevant participant's mean is then subtracted from each data point,

preserving the within-participant pattern of effects but removing between-participant baseline differences.

2. Add the original (prior to scaling) grand mean back to each data point to ensure that the overall dataset has the same mean as the original data, simply with between-subject variance removed.

3. Calculate within-subject intervals from the scaled data using the typical formula for a confidence interval (times a bias-removing scaling factor described by Morey, 2008).

It should be clear that the first step of this process, subtracting the relevant participant's mean, is equivalent to *residualization*. In regression, the difference between an observed value and the predicted value is a residual. When no other predictors are present, the predicted value for a given participant is that participant's mean. Thus, subtracting the participant's mean from each of that participant's data points is exactly the same as regressing the dependent variable on Subject (treated as a fixed factor) and extracting the residuals. The code given in Listing 1 demonstrates that this is true, using the sample data from Loftus and Masson (1994; see Figure 1 and discussion above); it shows that the scaled values obtained by subtracting each participant's mean are exactly the same as the regression residuals (r=1).

Listing 1. R code to demonstrate that scaled values based on the Cousineau approach are identical to residuals from ordinary least squares regression.

```
# Re-create the Loftus & Masson data
LoftusMasson <- melt(
    cbind(
          c(10,13,13),
          c(6,8,8),
          c(11,14,14),
          c(22,23,25),
          c(16,18,20),
          c(15,17,17),
          c(1,1,4),
          c(12,15,17),
          c(9,12,12),
          c(8,8,12)
    ),
    varnames=c("Condition","Subject"),
    value.name="Recall"
)

# Calculate the mean for each Subject in the LoftusMasson dataset
subjectmeans <- aggregate( LoftusMasson$Recall,
list(LoftusMasson$Subject), mean )

# Calculate the grand mean of the dataset
grandmean <- mean( LoftusMasson$Recall )

# Scale the data by the Cousineau method: from each datapoint,
subtract the
#    Subject mean and add the grand mean
LoftusMasson$CMscaled <- LoftusMasson$Recall -
subjectmeans[ LoftusMasson$Subject, "x" ] + grandmean


# Alternative method: calculate an intercept-only regression model
for each
#   Subject (using the function lme4::lmList), which just gets each
Subject's
#   mean, and then extract the residuals from that model and add the
grand mean
library(lme4)
model <- lmList( Recall ~ 1 | Subject, LoftusMasson )
LoftusMasson$residualscaled <- resid(model) + grandmean

# A plot demonstrating that these scaling methods give the same
results
plot( CMscaled ~ residualscaled, LoftusMasson,
    main=paste( "r =", cor( LoftusMasson$CMscaled,
LoftusMasson$residualscaled ) )
)
```

Just like ordinary least squares regression, mixed effects regression can also return residuals.[3] (These residuals, though, from a model in which subjects are treated as a random effect, will not be exactly the same as residuals from a model treating subjects as a fixed effect, which is essentially what is done in the Cousineau-Morey and Loftus & Masson methods. This is because random effects in a mixed effect model undergo shrinkage: rather than reflecting the actual mean for that subject or item, the best linear unbiased predictors for the subject and item intercepts [i.e., the estimates of how much a given subject's or item's mean deviates from the grand mean] are constrained to be somewhat closer to overall intercept of the model; see e.g. Blouin & Riopelle, 2005.)

Importantly, mixed effects regression residuals can be calculated from a model that includes both subjects and items as random effects. Thus, there is no need to aggregate over subjects or items. Rather, the data can be scaled in a way similar to the way data are scaled in the Cousineau-Morey method (step 1 above): fit a model with only a fixed intercept and with random intercepts for subjects and items, extract the residuals, and add the grand mean of the original data. In fact, as Blouin and Riopelle (2005) note, methods that calculate within-subject intervals based on treating subjects as fixed effects (including the Cousineau-Morey and Loftus & Masson methods) are in fact just a special case of methods that use a mixed effect model; the present proposal builds on their

[3]Mixed effect residuals, in which subjects are treated as a random effect, will not be exactly the same as residuals from a model treating subjects as a fixed effect (which is essentially what is done in the Cousineau-Morey and Loftus & Masson methods). This is because random effects in a mixed effect model undergo shrinkage: rather than reflecting the actual mean for that subject or item, the best linear unbiased predictors for the subject and item intercepts (i.e., the estimates of how much a given subject's or item's mean deviates from the grand mean) are constrained to be somewhat closer to overall intercept of the model (see e.g. Blouin & Riopelle, 2005).

observation by extending this concept to mixed effect models with more than one random effect.

2a. How to calculate intervals from scaled data

The next problem is how to use these scaled data to compute an interval. The Cousineau-Morey method simply uses a variation of the standard formula for a confidence interval:

$$\bar{x} \pm (sd(x)/\sqrt{N} \times tinv_{1-\frac{\alpha}{2},\ N-1} \times M)$$

where *x* is the scaled data, *N* is the number of participants, *tinv* is the quantile function for the *t* distribution (returning the critical *t* value for a given significance level and degrees of freedom), and *M* the correction factor from Morey (2008). This method is not applicable for designs from a mixed effect model, however, as the standard formula relies on calculating the critical *t* statistic for a given number of degrees of freedom (*N*-1), whereas the degrees of freedom for a mixed effect model are not known (Baayen et al., 2008). There are methods available to estimate the degrees of freedom, but the best option for the present application is to calculate bootstrap-based confidence intervals of the scaled data for each condition (Davison & Hinkley, 1997) rather than using a confidence interval formula. Bootstrapping for mixed effect models is straightforwardly implemented in the `bootMer` package of the R statistical environment. The `LMEMinterval()` function shown in Listing 2 is a convenience function which scales the dependent variable of a dataset using mixed-effect model residuals, and then

bootstraps the dependent variable to construct an interval; the mixed-effect-model-based intervals shown in Figure 2B were derived using this function.

2b. Implementation and code

As mentioned above, Listing 2 contains R code for a convenience function, `LMEMinterval()`, that calculates bootstrap mixed effect model intervals using the procedure described above. This code assumes that the random effects are crossed, and there is no guarantee that it will work for between-subject or between-item designs. It also requires that the `lme4` package is installed.

To use the function, you must pass it a data frame and a formula using the same formula syntax used in the `lme4` package. See Baayen et al. (2008), Chang & Lane (2016), *inter alia*, for introductions to the `lme4` syntax.

Listing 2. An R convenience function for calculating LMEM-based intervals.

```
LMEMinterval <- function(
      formula,
      data,
      boot.type="percentile",
      conf=.95,
      nsim=NULL){

      # This convenience function calculates LMEM-based "confidence" intervals for
      #     a given design and dataset.
      # Parameters:
      #     formula: a usual model formula, with one DV and one or more IVs. Currently
      #           this function is able to handle functions in DVs (e.g., using log(RT)
      #           rather than RT as a DV), but not in IVs. And I haven't done much testing
      #           of DVs with functions so there may be bugs; I prefer just creating a
      #           new column in the data frame (e.g., creating a logRT column).
      #           Also note that this is currently only implemented for single-factor
      #           designs. If you have a factorial (e.g. 2x2) design, this function will
      #           collapse it into a single-factor (e.g. 1x4) design.
      #     data: a data frame with repeated measures data
      #     conf: The confidence level (between 0 and 1) for the CI to plot. Defaults to .95.
      #     boot.type: which type of bootstrap to use. Defaults to "percentile". If set to
      #           anything else, it will instead use normal bootstrap. Percentile bootstrap
      #           is more accurate but slower, as it requires more iterations to get accurate.
      #     nsim: Number of bootstrap replicates to use. By default this will be 2000 if
      #           boot.type=="percentile" and 200 otherwise, but you can set `nsim` to override
      #           that.



      # Load the lme4 package
      require( lme4 )


      # Figure out the DV and the IVs.
      # This doesn't use all.var() because that strips away functions,
```

```
      #     whereas sometimes your formula might be like log(DV) ~
      #     rather than just DV.
      vars <- rownames(attr(terms(formula),"factors"))

      # Figure out the DV
      DV <- vars[1]

      # Figure out what the random effects are. The first line finds which
      #     IVs look like random effects terms (which ones have pipes), and
      #     the next line grabs the stuff after the pipe
      ranef_idx <- which( unlist( lapply( vars, function(x){ length( grep("|", x, fixed=T ) ) } ) )>0 )
      grouping.vars <- unlist( lapply( vars[ranef_idx], function(x){ strsplit( x, " | ",
fixed=T )[[1]][2] } ) )

      # Figure out the fixed IVs
      IVs <- vars[-c(1,ranef_idx)]

      # handles cases where the DV has a function around it (e.g. when the DV
      #     is `log(RT)` rather than just `RT`
      realDV <- all.vars(formula)[1]
        if( DV != realDV ){
            func <- gsub( paste0("(",realDV,")"), "", DV, fixed=T )
            DV <- realDV
            data[,DV] <- unlist(lapply( data[,DV], func ) )
      }



      ### A function to do the scaling. It first fits an intercept-only model to the data,
      ###   then subtracts the residuals and adds the intercept (the grand mean)
      LMEscale <- function( formula, data ){
            model <- lmer( formula, data )
            data$LMEscale <- as.numeric( resid(model) + fixef(model)["(Intercept)"] )
            return(data)
      }

      # Scale the data, using a model with only a fixed intercept and random intercepts
      lmerformula <- paste( DV, " ~ 1 + ", paste( "(1|", grouping.vars, ")", collapse=" + " ) )
```

```
	data <- LMEscale( lmerformula, data )


	### The rest of the code handles making bootstrap CIs of the scaled data. The general
	###   procedure is as follows: to get the bootstrap CIs we have to fit an lmer
	###   model to the scaled data. To make the models more likely to converge, we want
	###   to fit the models without random correlation parameters, and to do this
	###   I use a hack based on https://rpubs.com/Reinhold/22193, which requires first
	###   calculating a temporary model [which may not converge] and then extracting
	###   dummy coefficients directly from its model matrix to construct the good model.
	###   Finally, we bootstrap the good model a lot of times and use the bootstrap
	###   coefficients to get "confidence" intervals.

	# Collapse design into one factor (just treating each condition as its own condition,
	#     without considering main effects, interactions, etc.)
	data$Condition <- factor( do.call( paste0, lapply( IVs, function(IV){ data[,IV] } ) ) )

	# Create the temporary model, which may not converge, it doesn't matter
	lmerformula <- paste( "LMEscale ~ 0 + Condition + ", paste( "(1|", grouping.vars, ")", collapse=" +
" ) )
	junkmodel <- lmer( lmerformula, data )

	# Pull out dummy variables from the model matrix per https://rpubs.com/Reinhold/22193
	mydummies <- list()
	for ( i in 1:dim( model.matrix(junkmodel) )[2] ) {
		data[,paste0("c",i)] <- model.matrix(junkmodel)[,i]
	}

	# Make random effect terms using the dummy variables rather than the big 'Condition'
	#     variable. Per https://rpubs.com/Reinhold/22193, this ensures that random
	#     correlations between the random effects will not be used.
	#     We also specify no random intercepts; because the data are scaled, every
	#     subject/item/whatever should already have a mean of 0, so the random intercept
	#     is meaningless and in fact often cannot be fit anyway.
	ran <- paste( "0 +", paste( "c", 1:dim( model.matrix(junkmodel) )[2], collapse="+", sep="" ) )

	# Now fit the good model. Because there is no fixed-effect intercept, it will estimate
	#     a coefficient for each condition, rather than estimating comparisons between
```

```
      #     conditions.
      lmerformula <- paste( "LMEscale ~ 0 + Condition + ", paste( "(", ran, "||", grouping.vars, ")",
collapse=" + " ) )
      model <- lmer( lmerformula, data )

      # A function that gets the fixed-effect estimates from a model; this will be used
      #     for bootstrapping
      getcoef <- function(.){ getME(., "beta" ) }

      # Load the 'boot' package
      require(boot)

      # Print a message so we know the function is going
      message( "Bootstrapping LME-scaled values, may be very slow..." )

      # Figures out the number of bootstrap replicates to use (unless the user
      #     already specified how many)
      if( is.null(nsim) ) {
            nsim <- ifelse( boot.type=="percentile", 2000, 200 )
      }

      # Sets some variables that will be needed depending on whether we do a percentile
      #     or normal bootstrap
      if (boot.type=="percentile" ) {
            ci_idx <- 4:5
            ci.type <- "percent"
      } else {
            ci_idx <- 2:3
            ci.type <- "normal"
      }

      # Bootstrap the model. This line is what takes time
      bigboo <- bootMer( model, getcoef, nsim=nsim )

      # Extracts the requested CIs from the bootstrap samples
      CIs <- do.call(rbind, lapply( 1:length(fixef(model)), function(x){ boot.ci( bigboo, index=x,
type=substr(ci.type, 1, 4), conf=conf )[[ci.type]][ci_idx] } ) )
```

```
    # Gives human-friendly row names and column names
    rownames(CIs) <- substr( names( fixef(model) ), nchar("Condition")+1, nchar(names(fixef(model))) )
    colnames(CIs) <- c("Lower", "Upper")

    # Returns the CIs
    return(CIs)
}
```

By way of example, the following command may be used to read the data from the semantic priming experiment discussed above (Supplementary File 2) into R (after having saved the file to a local drive, and after having run the code in Listing 2 to have the `LMEMinterval()` function in R's memory for the current session) and then calculate the LMEM-based confidence intervals (this code will take several minutes to run, and may issue several failure-to-converge warning messages which may be ignored):

```
simdata <- read.csv( file.choose() )
( CIs <- LMEMinterval( RT ~ Condition + (1|Subject) +
(1|Item), simdata ) )
###               Lower    Upper
### Related   470.1059 495.5972
### Unelated  499.1634 522.6883
```

(Because of the randomness involved in bootstrapping, the results will be slightly different each time the function is called.)

The confidence level may be changed using the `conf` parameter, e.g., `conf=.99` will calculate a 99% rather than a 95% interval. The number of bootstrap replicates used for calculating the bootstrap interval may be changed using the `nsim` parameter. The number of bootstrap replicates is the main determinant of how long the function call will take. The default is to use 2000 replicates for a percentile bootstrap. To make the function run faster, you can request a normal bootstrap instead of a percentile bootstrap, which by default uses only 200 replicates:

```
( CIs <- LMEMinterval( RT ~ Condition + (1|Subject) +
(1|Item), boot.type="normal", simdata ) )
```

The function can straightforwardly be used for other datasets as well. For instance, the following example demonstrates the use of the function with the `lexdec` dataset that is included in the `{languageR}` package.

Listing 3. An additional example of the usage of LMEMinterval().

```
library( languageR )
# Create a variable which will be fully within-subject
and within-item
lexdec$WhichHalf <- factor( ifelse( lexdec$Trial>106,
"Second", "First" ) )
( CIs <- LMEMinterval( RT ~ WhichHalf + (1|Subject) +
(1|Word), lexdec ) )
```

3. Testing the method

It is worthwhile to test whether mixed effect intervals are in fact more accurate summaries of the data pattern than other types of within-subject interval. In what follows, I use several real datasets (a convenience sample including data from published psycholinguistics papers, datasets available online, and data from unpublished experiments) to compare mixed effect intervals, within-subject and within-item Cousineau-Morey intervals in terms of how closely they match the *p*-values obtained from a direct comparison between the conditions of interest.

3a. Evaluation criteria

A metric is needed to determine how accurate an interval is. One appropriate metric is whether the overlap between two conditions' intervals accurately corresponds to a *p*-value. As Cumming and Finch (2005) note, as long as several assumptions about the distribution of the data hold, two independent groups are usually different at the 5% significance level if their respective 95% confidence intervals overlap by just about 58% (specifically, $|\sqrt{2} - 2|$; see footnote 5) of the average margin of error (the margin of error is half of the confidence interval—for a symmetrical confidence interval, the margin of error is the distance from the parameter estimate to the lower or upper bound of the interval—and the average margin of error is the average of two condition's respective margins of error).[4] This applies to any significance level; for instance, if the *p*-value for

[4] This 58% overlap rule is related to the standard error of the difference between means. While the standard error for a single sample mean with *N* observations is $\frac{SD}{\sqrt{N}}$, the standard error of a difference between sample means uses a different formula (see, e.g., Pfister & Jancyk, 2013, equation 4). In a case where the two sample means have the same *N* and same variance, the formula for the standard error of the difference can be algebraically reduced to $\frac{SD}{\sqrt{N}} \times \sqrt{2}$. If the margin of error for the confidence interval of a sample mean is multiplied by $\sqrt{2}$, then the difference between means is significant if the confidence interval does not include the other mean, rather being significant if the two confidence intervals overlap by less than 58% of the average margin of error as above. To illustrate, consider a hypothetical case of two means of unpaired samples where each sample has the same *N* and same variance. The margin of error (MoE) can be calculated based on the standard error of each mean (and in this example the average MoE will equal the MoE for either mean, since their variances and *N*s are the same) or on the standard error of the difference. Assuming $\overline{x_1}$ is the lower of the two sample means and $\overline{x_2}$ is the higher, then the critical MoE of the difference (i.e., the margin of error such that the difference between means will be just barely significant at the chosen alpha level) is the one that is equal to or just below the size of the difference between means, or in other words, the difference between means is equal to the MoE. Keeping in mind that the MoE of the difference is, in this example, equal to the MoE of either mean times $\sqrt{2}$, this means that the critical margin of error is as follows:

$$\overline{x_2} - \overline{x_1} = MoE \times \sqrt{2}$$

Which can be solved for the MoE:

the difference between two groups is .2, then the two groups' 80% confidence intervals should overlap by just about 58% of the average margin of error, if the right assumptions are met. The same rough heuristic applies to two paired groups and their Cousineau-Morey intervals (a demonstration of this, using simulations, is given in Appendix 1). Therefore, we can test the accuracy of mixed effect intervals by calculating, for any given pairwise comparison in a crossed random effect dataset, what the p-value of that comparison is, and what is the largest interval that overlaps by just about 58% of the margin of error. For example, if the *p*-value for the difference between two conditions in a mixed-effect model is .003, and the 99.7% mixed effect based intervals of the two condition means overlap by 58% of the average margin of error, then the intervals are consistent with the *p*-value; on the other hand, if the 99.7% intervals overlap by much less or much more than that, then the intervals are inconsistent with the actual *p*-value.

3b. Sample

$$\frac{\overline{x_2} - \overline{x_1}}{\sqrt{2}} = MoE$$

On the other hand, when using the MoE of each mean, the critical MoE is the one such that the confidence intervals overlap by .58 times the average MoE:

$$(\overline{x_1} + MoE) - (\overline{x_2} - MoE) = .58 \times MoE$$

Which can be solved for the MoE:

$$\frac{\overline{x_2} - \overline{x_1}}{1.42} = MoE$$

As $\sqrt{2} \approx 1.42$, this demonstrates the motivation behind using 58% overlap as an approximate value to judge statistical significance—it corresponds to the $\sqrt{2}$ used for calculating a confidence interval of a difference. Indeed, instead of estimating whether two confidence intervals overlap by 58% the average MoE, another approach to using confidence intervals to roughly visualize statistically significant differences is to multiply the MoEs by $\sqrt{2}$, creating difference intervals that imply significance (except for the caveats mentioned above, if the sample variances or sample sizes differ) when the interval for one mean does not include the estimate of the other mean.

A convenience sample of 11 psycholinguistic studies with subject and item random effects, a continuous dependent variable, and at least one nominal independent variable was selected. This independent variable was repeated within subjects and items. Some of these studies also involved between-subjects or between-items variables; in these cases, these were treated as separate experiments and LMEM-based intervals for each group were calculated separately (for example, if a study included two conditions that were measured within both subjects and items, but also had a Group variable that was between-subjects, then this was treated as two separate datasets, one for each group). This was done to avoid the complications in creating appropriate intervals for a design involving both within-subjects and between-subjects comparisons. Across these studies, 162 pairwise comparisons were available to compare the performance of different types of intervals. The samples are summarized in Table 1; links to full citations are available in Supplementary File 4.

Source	**Publication status**	**Number of subjects**	**Number of items**	**Number of conditions**	**Number of pairwise comparisons**
Politzer-Ahles & Fiorentino (2013)	published	28	48	4	6
Politzer-Ahles & Husband (2016)	unpublished	48	48	4	6
Husband & Politzer-Ahles (2016)	unpublished	48	48	6	15
Baayen (2008)	online dataset (CRAN)	21	79	2	1
Politzer-Ahles (ms)	unpublished	40	24	4	6
Politzer-Ahles & Zhang (in press)	published	25	12	4	6
Politzer-Ahles	published	25	12	4	6

& Zhang (in press)					
Politzer-Ahles & Zhang (in press)	published	22	12	4	6
Politzer-Ahles & Zhang (in press)	published	22	12	4	6
Politzer-Ahles & Connell (dataset)	unpublished	25	12	4	6
Fruchter et al. (2015)	published	103	158	2	1
Matushanskaya et al. (dataset)	online dataset (OSF)	24	40	6	15
Matushanskaya et al. (dataset)	online dataset (OSF)	24	40	6	15
Gibson & Wu (2012)	Published	37	15	2	1
Mädebach et al. (dataset)	online dataset (OSF)	24	32	12	66

Table 1. Summary of the studies used to in the present test. Links to datasets are available in Supplementary File 4. Some studies were broken down into multiple smaller studies because they included between-subjects and/or between-items manipulations; see text.

3c. Calculations

Supplementary File 3 includes R code that compares the actual *p*-value to the *p*-value suggested by the interval, for all possible pairwise contrasts within any given dataset. (This code is only included for demonstration purposes; for calculating mixed-effect intervals under normal circumstances, the reader is advised to use the

`LMEMinterval()` function included in Listing 2.) Because the exact *p*-values for mixed effect model contrasts are not known, this script calculates several *p*-value approximations:

1. percentile bootstrap p-values (2 times the percentage of bootstrap beta values for that contrast which fall on the wrong side of zero; for example, if the estimate for a difference is 3, and forty of 2000 bootstrap replicates give an estimate below zero, then the bootstrap *p*-value is $2 \times \frac{40}{2000} = .04$)
2. *p*-values based on the Satterthwaite approximation for mixed effect model degrees of freedom (Schaalje et al., 2002)
3. *p*-values based on treating the *t* statistic as a *z* statistic (because the *t* distribution approaches the normal distribution when the number of degrees of freedom is high)

It then reports the corresponding interval-based *p*-values (i.e., the confidence levels of the widest intervals that overlap by just 58% of the margin of error) for three types of interval:

4. within-subject Cousineau-Morey intervals
5. within-item Cousineau-Morey intervals
6. mixed effect intervals

The crucial comparison is that between the "real" *p*-value (the bootstrap-based *p*-value of the actual difference between conditions) and the *p*-value suggested by how much the mixed-effect intervals overlap (e.g., if the 85% intervals for two conditions overlap by 58%, then this suggests a *p*-value of about 1-.85=.15 for the direct comparison between these two conditions). If mixed effect intervals are indeed more accurate, they should show a closer correspondence to the *p*-values.

3d. Results

Supplementary File 4 shows these various *p*-values from all within-subject-and-item contrasts in the sample. On average, the *p*-values implied by the LMEM-based intervals deviated from the actual bootstrap *p*-values by .017 (SD=.026); these are shown in Column M of Supplementary File 4. On the other hand, *p*-values implied by within-subject Cousineau-Morey intervals differed from actual *p*-values by an average of .035 (SD=.059; Column N), and *p*-values implied by within-item Cousineau-Morey intervals by .050 (SD=.080; Column O). For instance, to take one representative example: for the fourth comparison in Politzer-Ahles & Fiorentino 2013 (the fifth row of Supplementary File 4), the bootstrap *p*-value for the difference between these conditions in a mixed-effect model directly comparing them is .352. The *p*-value suggested by LMEM-based intervals is .36125 (i.e., it is the 63.875% intervals that overlap by just about 58% of the average margin of error), which deviates from the real *p*-value by .00925. On the other hand, the *p*-value suggested by within-subjects Cousineau-Morey intervals is .3065 (deviating from the real *p*-value by .0455), and that suggested by within-items Cousineau-Morey intervals is .301 (deviating by .051). Figure 3 shows the distribution of these differences for the entire sample of 162 pairwise comparisons; it can be seen that the LMEM-based intervals have the most observations clustered near 0 (i.e., very little difference between the real *p*-value and the LMEM-based one), whereas within-subject and within-item intervals have longer positive tails (indicating more values that are substantially different from the real *p*-values). Figure 4 shows the correlations between the real *p*-values and the *p*-values suggested by LMEM-based intervals (r=.994), within-subject intervals (r=.973), and within-item intervals (r=.952).

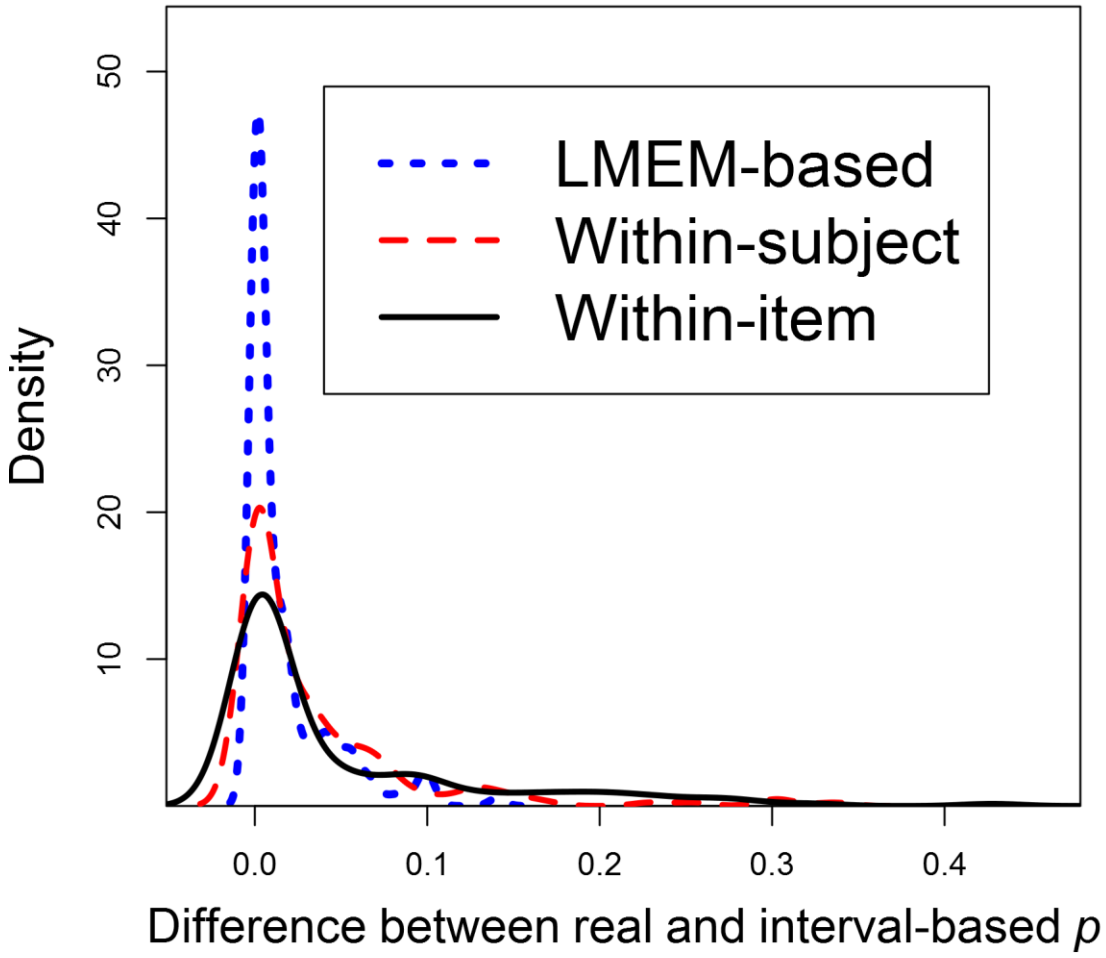
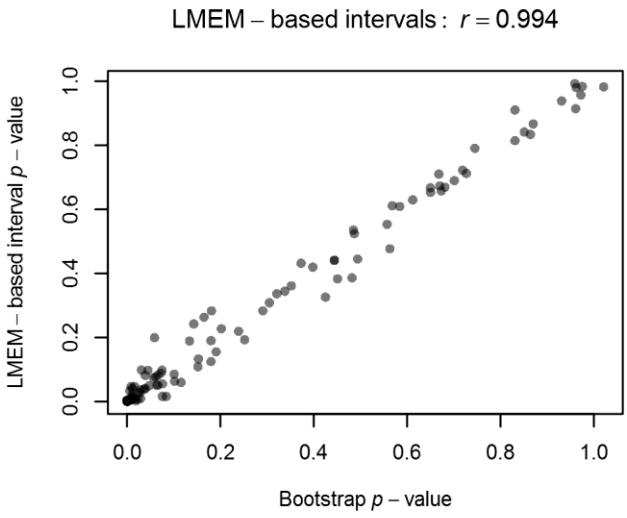
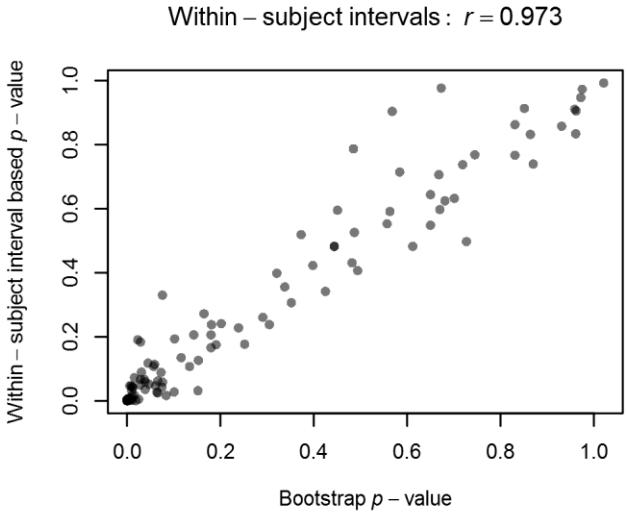


Figure 3. Smoothed kernel density of the differences between (a) the real p*-values and the LMEM-interval-based* p*-values (dotted blue line); (b) the real* p*-values and the within-subject-interval-based values (dashed red line); and (c) the real* p*-values and the within-item-interval-based values (solid black line).*


LMEM – based intervals : r = 0.994
LMEM – based interval p – value
Bootstrap p – value
0.0
0.2
0.4
0.6
0.8
1.0


Within – subject intervals : r = 0.973
Within – subject interval based p – value
Bootstrap p – value
0.0
0.2
0.4
0.6
0.8
1.0

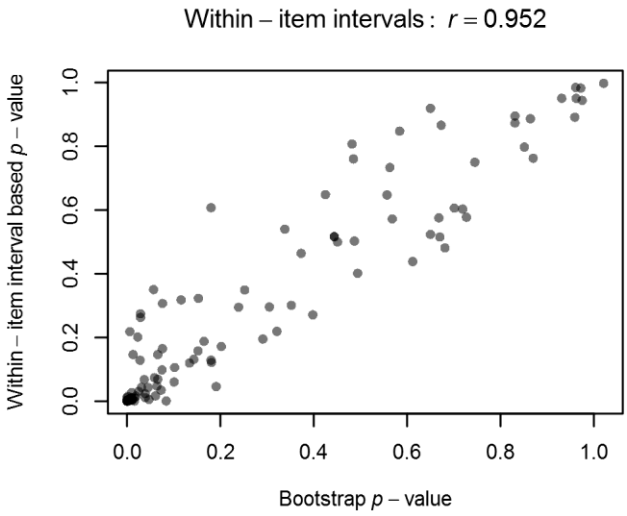

Within – item intervals : r = 0.952
Within – item interval based p – value
Bootstrap p – value
0.0
0.2
0.4
0.6
0.8
1.0

Figure 4. Scatterplots showing the correlation between real p*-values and* p*-values suggested by each type of interval: LMEM-based intervals (top), within-subject intervals (middle), and within-item intervals (bottom).*

A mixed effects model regressing the deviation (i.e., the amount from which the interval-based *p*-value differed from the real bootstrap *p*-value) on interval type[5] found a marginal effect of interval type ($\chi^2(2)=4.69$, $p=.096$), with mixed effects intervals marginally more accurate than by-subject Cousineau-Morey intervals ($b=0.03$, SE=0.02, $t=1.86$) and significantly more accurate than by-item Cousineau-Morey intervals ($b=0.05$, SE=0.02, $t=2.24$).

This advantage for mixed-effect intervals relative to within-subject and within-item intervals is also visible in the simulated data shown in Figure 2B. The 99% within-subject intervals (solid blue lines) in that figure suggest that the difference between the two conditions is not significant at the $\alpha=.01$ level, as they overlap by far more than 58% of the margin of error. The within-item intervals (solid black lines), on the other hand, suggest that the difference is significant at that level, as these intervals do not overlap at all. The mixed-effect model intervals, however, which are wider than the within-item

[5]The model (in the syntax of the `lme4` package in R) was as follows:

```
diff ~ IntervalType + ( IntervalType | Study/Experiment )
```

`diff` represents the difference between the interval-based *p*-value (i.e., the *p*-value based on what confidence level was needed to make intervals that overlapped by 58%) and the real *p*-value based on bootstrapping the pairwise contrast. `IntervalType` represents whether mixed effect intervals, by-subject Cousineau-Morey intervals, or by-item Cousineau-Morey intervals were used. `Study` corresponds to a paper that one or more experiments were put together into, and `Experiment` corresponds to a single dataset within the study (most studies analyzed here consisted of one experiment, but some studies consisted of multiple experiments with the same design, or of an experiment including between-subject or between-item factors which I used to split the dataset into sub-experiments with fully crossed subjects and items). `IntervalType` was dummy coded with mixed-effect intervals as the baseline, and the main effect of `IntervalType` was tested via log-likelihood test comparing the above model with a model without that fixed effect (i.e., a model with only a fixed intercept and the random effects).

intervals but narrower than the within-subject intervals, overlap by just about half the margin of error. This is consistent with the results of the direct statistical comparison, which gives a test statistic of t = -2.17 (see above), which corresponds to a p-value close to .01, which is consistent with the fact that it is the 99% mixed-effect intervals which overlap by close to 58% the margin of error. It should be noted, however, that this is a trend, rather than a guarantee that LMEM-based intervals will align more closely with p-values; in fact, as can be seen in Supplementary File 4, there are some contrasts for which the p-values suggested by within-subject or within-item intervals align more closely with the real p-values, although these contrasts are in the minority.

4. Conclusion

The foregoing discussion presented a method for extending within-subject intervals to research designs with crossed random effects, and demonstrated that, in a sample of real datasets, the extent to which these intervals overlap (which is a common heuristic that practitioners use, often erroneously, to judge whether two means are different) corresponds more closely to the inferential statistics than previous within-subject and within-item types of intervals do. While these are not confidence intervals (i.e., they do not license conclusions about how often they will include a given parameter on repeated sampling), and they do not justify inferences about plausible parameter values (a limitation shared by *bona fide* confidence intervals), they do offer a quick and roughly accurate means to visualize significant pairwise differences between a large number of conditions.

The present method is only discussed for relatively simple designs here. While it could be extended to multi-factor designs and to designs including non-categorical (i.e., ordinal or continuous) outcomes, these more challenging issues are beyond the scope of the present introduction. Furthermore, a unique challenge for this method is that random effects may also have different structures than those presented here. Random effects can be nested rather than crossed, or an experiment may have a combination of crossed and nested random effects. Likewise, some fixed effects may have corresponding random slopes and others (e.g., between-subject effects) may not. Here I have only considered designs that are fully within-subjects and within-items (when faced with datasets with between-subject or between-item factors, I split them up into multiple datasets and calculated mixed effect intervals separately for each sub-dataset). The method could be used for designs that are not like this (for example, where a given subject contributes data to each condition, but a given item only appears in one condition), but the code given here would need to be adapted.

Because the method involves scaling, it is subject to the same concerns discussed by Franz & Loftus (2012), namely, the scaling may sometimes cause variance to propagate between conditions and may hide violations of the assumption of circularity. To my knowledge, every type of within-subject interval faces this problem; the only way around it is to plot true confidence intervals of differences (as noted by Franz and Loftus), rather than within-subject or LMEM-based intervals of individual conditions.

Visually evaluating whether two intervals overlap by 58% of their average margin of error can sometimes be difficult. Practitioners who are interested could instead multiply the margins of error (i.e., half the interval) by $\sqrt{2}$. As described above in Footnote 5 (see also Pfister & Jancyk, 2013), this creates a difference-adjusted interval

such that the difference between two means is likely to be significant at the given alpha level if one mean's interval does not contain the other mean. Just as for any other attempts to evaluate differences using two means' intervals, inferences made from this comparison are only rough estimates, and will be invalid if the two means have very different standard deviations or very different numbers of observations (Cumming & Finch, 2005).

A practical concern is that calculating bootstrap confidence intervals takes a long time, particularly for models with many conditions. The datasets analyzed here took anywhere from several minutes to several hours to calculate percentile bootstrap confidence intervals. Other methods of bootstrapping may be faster (for instance, normal bootstrap confidence intervals can be estimated with far fewer replicates than percentile bootstrap confidence intervals), or intervals can be calculated using a confidence interval formula and estimating the degrees of freedom, but this method still is substantially slower than other within-subject interval methods. Some researchers may not want to spend this much extra time to obtain intervals that are only slightly more accurate, especially given that these intervals, just like other within-subject intervals, are still only a rough heuristic and will never be true confidence intervals. Nevertheless, for researchers who have sufficient time and want to present the most accurate heuristic data summary possible, while keeping in mind the limitations, this method may be a useful extension of other within-subject interval methods, as it does not require aggregating across items or other random effects.

Acknowledgements

I would like to thank Drs. Page Piccinini and Rory Turnbull for feedback on this work. Any errors are my own.

References

Baayen, H., Davidson, D., & Bates, D. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language, 59*, 390-412. DOI: 10.1016/j.jml.2007.12.005

Baguley, T. (2012). Calculating and graphing within-subject confidence intervals for ANOVA. *Behavior Research Methods, 44*, 158-175. DOI: 10.3758/s13428-011-0123-7

Barr, D., Levy, R., Scheepers, C., & Tily, H. (2013). Random effects structure for confirmatory hypothesis testing: keep it maximal. *Journal of Memory and Language, 68*, 255-278. DOI: 10.1016/j.jml.2012.11.001

Belia, S, Fidler, F., Williams, J., & Cumming, G. (2005). Researchers misunderstand confidence intervals and standard error bars. *Psychological Methocs, 10*, 389-396. DOI: 10.1037/1082-989X.10.4.389.

Blouin, D., & Rioppele, A. (2005). On confidence intervals for within-subjects designs. *Psychological Methods, 10*, 397-412. DOI: 10.1037/1082-989X.10.4.397

Chang, Y., & Lane, D. (2016). Generalizing across stimuli as well as subjects: a non-mathematical tutorial on mixed-effects models. *The Quantitative Methods for Psychology, 12,* 201-219. DOI: 10.20982/tqmp.12.3.p201

Cousineau, D. (2005). Confidence intervals in within-subject designs: a simpler solution to Loftus and Masson's method. *Tutorials in Quantitative Methods for Psychology, 1*, 42-45. DOI: 10.20982/tqmp.01.1.p042

Cumming, G., (2014). The new statistics: why and how. *Psychological Science, 25*, 7-29. DOI: 10.1177/0956797613504966

Cumming, G., & Finch, S. (2005). Inference by eye: confidence intervals and how to read pictures of data. *American Psychologist, 60,* 170-180. DOI: 10.1037/0003-066X.60.2.170

Davison, A., & Hinkley, D. (1997). *Bootstrap Methods and Their Application*. Cambridge: Cambridge University Press.

Franz, V., & Loftus, G. (2012). Standard errors and confidence intervals in within-subjects designs: generalizing Loftus and Masson (1994) and avoiding the biases of alternative accounts. *Psychonomic Bulletin & Review, 19,* 395-404. DOI: 10.3758/s13423-012-0230-1

Gould, L., McKibben, T., Ekstrand, C., Lorentz, E., & Borowsky, R. (2016). The beat goes on: the effect of rhythm on reading aloud. *Language, Cognition and Neuroscience, 31*, 236-250.

Judd, C., Westfall, J., & Kenny, D. (2012). Treating stimuli as a random factor in social psychology: a new and comprehensive solution to a pervasive but largely ignored

problem. *Journal of Personality and Social Psychology, 103*, 54-69. DOI: 10.1037/a0028347

Loftus, G., & Masson, M. (1994). Using confidence intervals in within-subject designs. *Psychonomic Bulletin and Review, 1*, 476-490. DOI: 10.3758/BF03210951

Morey, R. (2008). Confidence intervals from normalized data: a correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology, 4*, 61-64. DOI: 10.20982/tqmp.04.2.p061

Morey, R., Hoekstra, R., Rouder, M., & Wagenmakers, E. (2016). The fallacy of placing confidence in confidence intervals. *Psychonomic Bulletin & Review, 23*, 103-123. DOI: 10.3758/s13423-015-0947-8

Pfister, R., & Jancyzk, M. (2013). Confidence intervals for two sample means: calculation, interpretation, and a few simple rules. *Advances in Cognitive Psychology, 9*, 74-80. DOI: 10.5709/acp-0133-x

Rouder, J., & Morey, R. (2005). Relational and arelational confidence intervals: a comment on Fidler, Thomason, Cumming, Finch, and Leeman (2004). *Psychological Science, 16*, 77-79). DOI: 10.1111/j.0956-7976.2005.00783.x

Schaalje, G., McBride, J., & Fellingham, G. (2002). Adequacy of approximations to distributions of test statistics in complex mixed linear models. *Journal of Agricultural, Biological, and Environmental Statistics, 7*, 512-524. DOI: 10.1198/108571102726

Appendix 1: Simulations to demonstrate the relationship between confidence interval overlap and *p*-values

Some simple simulations can be used to test the validity of a given type of confidence interval. These in turn will be useful as diagnostics to examine whether LMEM-based intervals yield comparable results.

The CI of paired differences

Let's first consider the simplest case: a test of two paired samples (which is mathematically equivalent to a one-sample test against 0, once the two vectors of paired samples are turned into one vector of differences). In this case a CI is both descriptive and inferential (to adopt Tryon's [2001] terms): it is the case both that 95% of 95% CIs will contain the true mean, *and* that a 95% CI that does not cross zero indicates significance at α=.05. Using simulations to illustrate the descriptive validity of the CI (i.e., that 95% of samples contain the true mean) is straightforward; since the within-subject CIs we discuss below are not true confidence intervals, however, I will only use simulations to illustrate the inferential validity of the CI (i.e., that the CI includes zero α% of the time); for paired samples these are really just two different ways of stating the same problem. To verify the inferential validity of the CI, one simply needs to generate a large number of samples (of any sample size) from a given population where $\mu_{difference}$=0, and count the proportion of samples in which the CI fails to include 0 (i.e., the proportion of false positives). For a 95% CI this should be 5%, and this is indeed what we observe with a large number of samples.

Listing 4. R code to simulate a large number of samples and demonstrate that the proportion of samples in which the CI of the paired differences includes zero corresponds to the p*-value.*

```
simulate_diff <- function( within_effect, within_effectsd, between_sd, n ){

      # Figure out what the effect will be for each subject
      subjeffects <- rnorm( n, within_effect, within_effectsd )

      # Figure out what the mean will be for each subject
      subjmeans <- rnorm( n, 10, between_sd )

      # Create a 2-row matrix, where rows are the two conditions and
      #     columns are the n subjects
      subjmeans <- rbind( subjmeans - .5*subjeffects, subjmeans
+ .5*subjeffects )

      # Calculate the pairwise differences
      diff <- mean( subjmeans[2,] - subjmeans[1,] )

      # Calculate the standard 95% CI of the differences
      diffME <- sd( subjmeans[2,] - subjmeans[1,] ) / sqrt(n) * qt(.975, n-
1)

      # Return TRUE if the CI fails to include 0, FALSE otherwise
      return( abs(diff) - diffME > 0 )
}

# Run this function 1,000,000 times and show the proportion of false
positives
```

```
nsim <- 1000000
length( which(results <- unlist( lapply( 1:nsim,
function(x){ simulate_diff( 0, 1, 3, 48) } ) ) ) / nsim
```

Also note that in this case a CI and a *p*-value (from a *t*-test) are mathematically related, and one can be derived from the other. The *p*-value is simply the confidence level of the largest CI that just barely touches zero. This value can be straightforwardly calculated from the formula for the margin of error (i.e., half of the CI) where qt() is the quantile function (which finds the critical *t* value for a given significance level and degrees of freedom; here the significance level is adjusted for a two-tailed test):

$$\frac{SD_x}{\sqrt{n}} \cdot qt\left(1 - {}^{\alpha}/_{2}, n-1\right) = \bar{x}$$

This can be algebraically solved for α (where pt() is the cumulative distribution function, the inverse of the quantile function: for a given *t*-value it returns a significance level):

$$\alpha = 2 \cdot \left(1 - pt\left(\frac{\bar{x}}{{SD_x}/{\sqrt{n}}}, n-1\right)\right)$$

Notice that this is, in fact, the t-test formula. Thus it is straightforwardly demonstrable that a *p*-value corresponds to the confidence level for the widest CI that just touches zero (or,

specifically, 100×(1-*p*) corresponds to that confidence level). This can also be confirmed with any random sample:

Listing 5. R code to simulate a large number of samples and demonstrate the p*-value is equal to the confidence level of the CI that just touches zero.*

```
# Population parameters
n <- 48
within_effect <- 0
within_effectsd <- 1
between_sd <- 3

# Get a sample
subjeffects <- rnorm( n, within_effect, within_effectsd )
subjmeans <- rnorm( n, 10, between_sd )
subjmeans <- rbind( subjmeans - .5*subjeffects, subjmeans
+ .5*subjeffects )

# Get the within-subject differences
diff <- mean( subjmeans[2,] - subjmeans[1,] )


# Find the alpha level that would be needed for a two-tailed CI that
only just touches zero
CI_alpha <- 2*(1 - pt( abs(diff) * sqrt(n) / sd(subjmeans[2,] -
subjmeans[1,]), n-1 ) )

# Show the t-test results and the CI confidence level
t.test( subjmeans[2,], subjmeans[1,], paired=T )
```

```
      paste0( "You need a ", round(100*(1-CI_alpha),3), "% CI to just touch
0" )
```

Two within-subject intervals, for two paired samples

Now we consider a more complicated case: within-subject intervals for the means of the two paired samples, rather than one CI for the difference. As has been described previously (e.g., Cumming & Finch, 2005, among others), a 95% CI of a condition mean is not directly informative about differences between that condition mean and other condition means: i.e., while the endpoints accurately provide a description of where the mean is, directly looking at whether that condition's CI overlaps with another does not provide an inference about whether the two conditions' means differ at α=.05. Instead, as noted by Cumming & Finch (2005), as long as each group is n>10 and the two groups have similar variance, then the two means can be roughly inferred to differ at about α=.05 if the *overlap* between the two confidence intervals is about 58% of the margin of error (MoE; i.e., half the width of the CI).

This .58×MoE rule only directly applies to normal CIs in the case of comparisons between two independent groups. For comparisons between two conditions of within-subject (paired) data, a within-subject interval is needed (Loftus & Masson, 1994). Here I will illustrate this with simulated Cousineau-Morey intervals.[6] Just as we demonstrated inferential validity for a

[6] I do not illustrate Loftus & Masson intervals in this simulation, as these intervals were never meant to be used for direct pairwise comparisons, but rather were meant to give impression of overall effects across

one-sample test (i.e., a paired differences test) above by counting the proportion of CIs that were simulated from a population with $\mu_{difference}=0$ but failed to cross 0, we can conduct a similar demonstration for the two-condition case by drawing a large number of paired samples and counting the proportion of simulations in which the sample Cousineau-Morey CIs overlap by less than 58% of the average margin of error. Again, for a 95% interval and a population in which $\mu_{difference}=0$, this should be approximately 5% of samples, and this is indeed the case in the simulations.

Listing 6. R code to simulate a large number of samples and demonstrate the p*-value is close to the confidence level of the Cousineau-Morey intervals that overlap by just 58% of the average MoE.*

```
simulate_cmci <- function( within_effect, within_effectsd, between_sd, n ){

      # Figure out what the effect will be for each subject
      subjeffects <- rnorm( n, within_effect, within_effectsd )

      # Figure out what the mean will be for each subject
      subjmeans <- rnorm( n, 10, between_sd )

      # Create a 2-row matrix, where rows are the two conditions and
      #     columns are the n subjects
      subjmeans <- rbind( subjmeans - .5*subjeffects, subjmeans
+ .5*subjeffects )
```

multiple conditions. For a simple two-condition case with perfect sphericity, the Loftus & Masson interval will give the same result as a Cousineau-Morey interval.

```
	# Here we scale the subject means (per Cousineau, 2005)
	scaled <- subjmeans - rbind(colMeans(subjmeans),colMeans(subjmeans))
+ mean(colMeans(subjmeans))

	# Here we get the margin of error of the scaled means, times the
	#	Morey (2008) correction factor
	MEs <- apply( scaled, 1, sd ) / sqrt(n) * qt(.975,n-1) * sqrt(2)

	# Get the mean, CI lower bound, and CI upper bound for each subject
	means <- rowMeans( subjmeans )
	lower <- means - MEs
	upper <- means + MEs

	# Find the average margin of error for the two groups (in a 2-
condition
	#	case, each ME equals the average ME)
	average_margin_of_error <- mean( MEs )

	# Calculate the absolute overlap
	overlap <- ifelse( means[1]<means[2],
			upper[1]-lower[2],
			upper[2]-lower[1]
		)

	# Return TRUE if the overlap is less than 58% the average margin of
error, FALSE otherwise
	return( overlap/average_margin_of_error <= .58 )

}
```

```
# Run this function 100,000 times and show the proportion of false
positives
nsim <- 100000
length( which( results <- unlist( lapply( 1:nsim,
function(x){ simulate_cmci(0, 1, 3, 48) } ) ) ) ) / nsim
```

Just as we did above, we can also draw a relationship between the Cousineau-Morey interval and the p-value by noting that the p-value is related to the confidence level of the interval. In this case, rather than corresponding to the confidence level of the widest interval that just touches zero, now the *p*-value corresponds to the confidence level of the widest pair of intervals that just barely overlaps by 58% of the average margin of error. (In the case with only two conditions, after scaling the data according to the Cousineau-Morey procedure then the margins of error for the two conditions are equal, and thus also equal to the average margin of error.) Once again, this can be algebraically calculated (where x_1 is the lower of the two condition means, and MoE is the margin of error for each condition):

$$\frac{(\overline{x_1} + \ MoE) - (\overline{x_2} - MoE)}{MoE} = \ .58$$

which can be solved for MoE:

$$MoE = \frac{\overline{x_2} - \overline{x_1}}{1.42}$$

And, in turn, solved for α after plugging in the MoE formula (with the correction factor, based on Morey, 2008): $\frac{SD_{x_{scaled}}}{\sqrt{n}} \cdot qt\left(1 - {}^{\alpha}/_{2}, n-1\right) \cdot \sqrt{2}$:

$$\alpha = 2 \cdot (1 - pt\left(\frac{\overline{x_1} - \overline{x_2}}{-1.42} \cdot \frac{\sqrt{n}}{SD_{x_{scaled}} \cdot \sqrt{2}}\right))$$

Finally, as we did above for differences, here we can demonstrate for any simulated sample that this widest pair of Cousineau-Morey intervals that just barely reaches 58% overlap also has a confidence level approximately corresponding to 100×(1-*p*):

Listing 7. R code to calculate a large number of Cousineau-Morey intervals at differnet confidence levels for a simulated dataset, and show that the confidence level of the intervals that overlap by just 58% of the average MoE closely corresponds to the p*-value.*

```
      # Population parameters
      n <- 48
      within_effect <- 0
      within_effectsd <- 1
      between_sd <- 3

      # Get a sample
      subjeffects <- rnorm( n, within_effect, within_effectsd )
      subjmeans <- rnorm( n, 10, between_sd )
      subjmeans <- rbind( subjmeans - .5*subjeffects, subjmeans
+ .5*subjeffects )

      # Get the condition means
```

```
      means <- rowMeans( subjmeans )

      # Scale the data
      scaled <- subjmeans - rbind(colMeans(subjmeans),colMeans(subjmeans))
+ mean(colMeans(subjmeans))

      sds <- apply( scaled, 1, sd )


      # Find the alpha level that would be needed for a two-tailed CI that
only just touches zero
      CI_alpha <- 2*(1 - pt( (-abs(means[1]-means[2])/-1.42) * ( sqrt(n)/
(mean(sds)*sqrt(2)) ), n-1 ) )

      # Show the t-test results and the CI confidence level
      t.test( subjmeans[2,], subjmeans[1,], paired=T )
      paste0( "You need a ", round(100*(1-CI_alpha),3), "% CI to just
barely have 58% overlap" )
```

(Notice that here, because the .58×MoE rule only approximates α=.05 significance level, rather than exactly equaling it, the confidence levels do not exactly match up with the *p* value calculated by the t-test; in fact, they are conservative, slightly overestimating the actual *p* value.)

In summary, for Cousineau-Morey intervals of two paired means, the validity of the interval can be confirmed by using simulations to show that the false alarm rate (for inferences based on the proportion of overlap between two intervals) in a population with a zero effect is close

to the nominal α level, and that the significance levels that can be inferred from the .58×MoE rule closely approximate (although do not exactly equal) the *p*-value that would be yielded from directly testing the difference. We can then apply these same diagnostics to crossed mixed-effect intervals.