

Two-factor_repeated

AxBxs

Bruce Dudek

2023-04-17

Contents

1	Introduction	2
2	The R Environment	3
3	The data	3
4	Data setup	4
5	Exploratory Data Analysis and Visualizations	4
5.1	Boxplots and Bar Graphs	5
6	Basic Omnibus ANOVA	8
6.1	Using <code>aov</code>	8
6.2	Base ANOVA using <code>afex</code>	11
6.3	Base ANOVA using <code>rstatix</code>	14
6.4	Conclusions regarding base ANOVA implementation	15
7	Begin working with <code>emmeans</code> for followup analyses	15
7.1	Main effect contrasts and interaction contrasts	15
8	Simple Main effects and contrasts	18
8.1	Simple Main Effects	18
8.2	Simple main effect contrasts	21
8.2.1	Simple main effect contrasts using <code>emmeans</code> on an <code>afex</code> model	21
8.2.2	Simple main effect contrasts on an <code>aov</code> model	22
8.3	Other Approaches to simple effects with <code>phia</code> , <code>rstatix</code> and <code>emmeans</code>	24
8.4	Manually subset the data to obtain a type of Simple Main Effect approach	24
8.4.1	First, use tidyverse methods for efficiency	24
8.4.2	Subset the data frame for SME using base R methods	25
8.5	Recommendations regarding contrasts and simple main effects	30
8.6	Pairwise multiple comparison follow up analyses	31
9	Now some mixed models analysis	31
10	Reproducibility	37
	References	38

1 Introduction

This tutorial document attempts to outline multiple methods for analysis of a two-factor, factorial within subjects design (repeated measures) - textbooks call this an $A \times B \times s$ design (A & B are IVs, and s is “subject” or case). It is intended for students in the APSY510/511 statistics sequence at the University at Albany, but can have general utility for researchers and students with some training in repeated measures analyses.

The reader would find it useful to first examine the tutorial for a single factor repeated measures design ($A \times s$), where much of the logic of the analytical flow and the logic of R code for contrasts is laid out.

In order to use this tutorial effectively, the reader should be alert to terminology specific to the style of ANOVA analysis that is traditional in the textbooks such as Kirk, Winer, Keppel, and Maxwell/Delaney/Kelley. The relevant terminology/notation includes:

An $A \times B \times s$ design: Two independent variables, perhaps called A, and B. The “s” represents “subject” on which the repeated measurements are taken. With this traditional analysis it is assumed that each case (subject) has measurements of the dv taken at the same time or under the same conditions. When the repeated measure is measured at different time points for different subjects, linear mixed models are appropriate. The design is a fully “within subjects” design where participants/cases are measured repeatedly under the combined conditions of factors A, and B.

Levels: The groups defined by each variable are called levels. E.g., in the primary example used in this document the “valence” factor (factor A) has three “levels”: negative, positive, and neutral.

Factorial: The design is called a factorial since each level of each factor is found in all possible combinations with all levels of all other factors. So, for example a 3×4 factorial design ($A \times B$) has 12 total combinations or “cells”

Cell means vs Marginal Means, Omnibus Effects: Cells are defined by the three way combination of factors. The 3×4 design used in this document has 12 cells. But we can imagine “collapsing” on one or more IVs to produce marginal sets of means. For example collapsing on factor A produces a B matrix of means (4 means) and that would lead to assessment of the omnibus main effect of B. The omnibus terms in a 2 way factorial are the two main effects and the two way $A \times B$ interaction.

Simple Effects When effects of one or more factors are examined at only specific levels of one or more other factor, then those effects are “simple” In this two factor design these are:

Simple Main effects: effects of one IV at levels of another factor, but collapsed on the third. E.g., A at B1 or B at A3. These are 1-way questions about the effect of a single IV but the location of the questions is specified by a level of a second variable.

An important part of the grasp of the logic is the understanding of these following concepts:

1. An omnibus 2 way interaction tests whether simple main effects differ.
2. 2 way interaction contrasts test whether simple main effect contrasts differ.

In addition, many sections of this tutorial rely on the idea of creating orthogonal sets of contrasts. This philosophy derives from the alignment of that approach with theory-drive *a priori* hypotheses and from the recognized value of full rank models in ANOVA work.

The general philosophy in approaching factorial designs is to follow omnibus analyses with examination of a mix of contrasts and simple effects. In such a design, this means that simple effects (and contrasts) can be effects of either repeated factor. Such contrasts can be performed on interactions, simple main effects, or main effects. These kinds of effects are best approached with a strategy that involves use of error terms tailored to the specific effect - usually called “specific error terms”. These types of follow up analyses are not easily accomplished with the tools available in R. Some approaches outlined below have led to questionable error term choices. Others are appropriate. The best practices approach using specific error terms was strongly recommended by Boik (1981) and Rogan (Rogan, Keselman, & Mendoza, 1979). More recent writers/textbooks have urged this approach and it has been adopted in major commercial software. These are challenging in R and a primary reason why results from R will often not match those that users

may be accustomed to from SPSS (MANOVA and UNIANOVA), and SAS. An analogous issue arises when attempting to test *post hoc* questions with pairwise comparison methods - error term choices are challenging in R. This document lays out strategies for several approaches - no single integrated/comprehensive set of functions provides results that make sense at the time of writing of this document.

The document is structured into major sections on Exploratory Data Analysis, Omnibus ANOVA analysis (many ways), then a focus on **afex** for omnibus analysis, follow up analyses such as contrasts and post hoc testing, and finally a cursory section on linear mixed effects modeling.

The short summary of this tutorial is that it concludes that R has strengths for EDA, omnibus analysis and linear mixed effects models on this two-factor repeated measures design, but approaches to contrasts and multiple comparisons are often vexing and problematic.

2 The R Environment

Several packages are required for the work in this document. They are loaded here, but comments/reminders are placed in some code chunks so that it is clear which package some of the functions come from. Either functions are specified with the `pkgname::functionname` code style, or the package is mentioned in accompanying text.

```
library(tidyverse,quietly=TRUE, warn.conflicts=FALSE)
library(afex,quietly=TRUE, warn.conflicts=FALSE)
library(car,quietly=TRUE, warn.conflicts=FALSE)
library(emmeans,quietly=TRUE, warn.conflicts=FALSE)
library(ez,quietly=TRUE, warn.conflicts=FALSE)
library(ggplot2,quietly=TRUE, warn.conflicts=FALSE)
library(gt,quietly=TRUE, warn.conflicts=FALSE)
library(knitr,quietly=TRUE, warn.conflicts=FALSE)
library(lme4,quietly=TRUE, warn.conflicts=FALSE)
library(lmerTest,quietly=TRUE, warn.conflicts=FALSE)
library(nlme,quietly=TRUE, warn.conflicts=FALSE)
library(psych,quietly=TRUE, warn.conflicts=FALSE)
library(rstatix,quietly=TRUE, warn.conflicts=FALSE)
library(phia,quietly=TRUE, warn.conflicts=FALSE)
library(Rmisc,quietly=TRUE, warn.conflicts=FALSE)
```

3 The data

The data set used in this document comes from the Keppel Experimental Design textbook (Keppel & Wickens, 2004).

Description (taken from the textbook description, with some modification):

Suppose a researcher is interested in comparing the speed with which words differing in emotional value are learned. An $A \times B \times S$ design is used, thus both IVs are repeated measures factors. Each of $n=8$ college-student subjects studies a list of 60 different words. The sixty words are classified according to twelve separate groupings. Twenty words are judged to be of negative emotional valence (level A1), 20 positive (level A2), and 20 neutral (A3). Within each of these categories, 5 words are presented once (level B1), 5 words twice (level B2), 5 three times (level B3), and 5 four times (level B4).

The assignment of specific words to different presentation conditions (factor B) is determined randomly for each subject. The list of 60 words, which is also randomly ordered for each subject, is presented at the rate of two seconds per word, and after the study period, the subjects are given 5 minutes to recall the words in any order.

Each subject has received each type of word under every condition of repetition (factor B), so both independent variables are within-subject factors. They are crossed to make a 3x4 factorial arrangement of the within subject factors. We can call it an AxBxS design because since each subject receives the words in each of the twelve categories, A and B are also crossed with subject as well as with each other.

Factor A, the emotional valence factor is called VALENCE in our analyses. Since A3 was a control condition, it might make sense to use a reverse Helmert set for contrasts on that factor.

Factor B might be called and EXPOSURE factor and the levels (1,2,3,4) are equally spaced along a continuum. Orthogonal trend contrasts would be appropriate for this factor.

Since each subject experiences five words under each of the twelve conditions, the DV score for each of the twelve conditions can range from zero to five.

Some nuances of the data set appear to reveal that it is a synthetic data set for textbook purposes, but it is a reasonable design to try to tear apart with traditional ANOVA techniques, modern approaches to contrasts, and some attempts at linear mixed modeling.

4 Data setup

The data are in a long-format .csv file. The first and last few lines of the data frame are shown.

```
abs1.df <- read.csv("axbxs_k4long_3x4.csv", stringsAsFactors=T)
#colnames(abs1.df) <- c("snum", "index1", "recall", "valence", "exposure")
psych::headTail(abs1.df)
```

```
##      snum index1 recall  valence exposure
## 1      1   a1b1      1 negative      one
## 2      1   a1b2      1 negative      two
## 3      1   a1b3      2 negative     three
## 4      1   a1b4      2 negative     four
## ...  ...   <NA>    ...      <NA>    <NA>
## 93     8   a3b1      2  neutral      one
## 94     8   a3b2      2  neutral      two
## 95     8   a3b3      3  neutral     three
## 96     8   a3b4      4  neutral     four
```

In order to perform the ANOVA analyses, we need the case number variable to be of class ‘factor’. The independent variables (valence and exposure) are already factors since we specified “stringsAsFactors=T” in the read.csv function.

```
abs1.df$snum <- as.factor(abs1.df$snum)
```

Next, the order of the IV levels are changed from the default alphabetical, for graphing purposes.

```
abs1.df$valence <- ordered(abs1.df$valence,
                          levels=c("negative", "positive", "neutral"))
levels(abs1.df$valence)
abs1.df$exposure <- ordered(abs1.df$exposure,
                          levels=c("one", "two", "three", "four"))
levels(abs1.df$exposure)
```

```
## [1] "negative" "positive" "neutral"
## [1] "one"     "two"     "three"   "four"
```

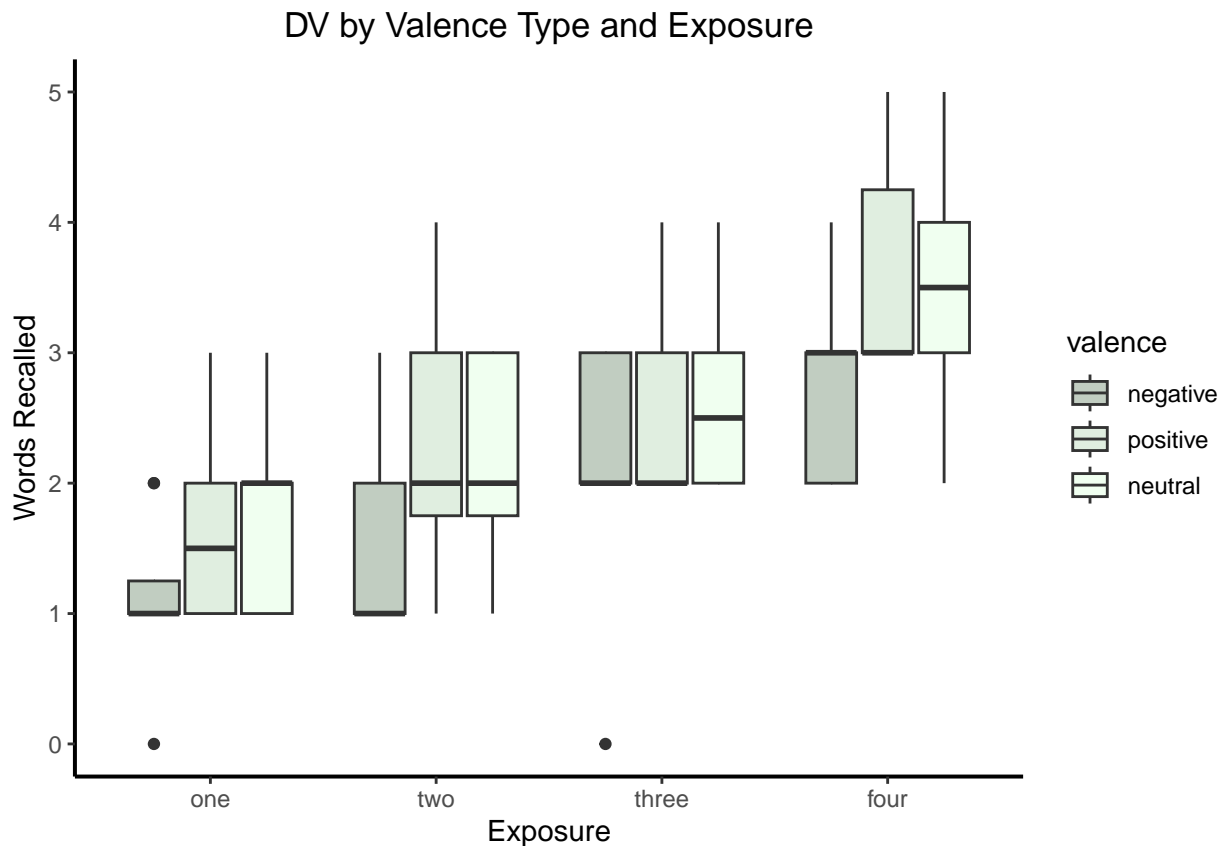
5 Exploratory Data Analysis and Visualizations

Multiple descriptive statistics capabilities and graphics are possible.

5.1 Boxplots and Bar Graphs

Boxplots with only $n=8$ are not terribly helpful, but this provides a template.

```
p1box <- ggplot(data=abs1.df, aes(x=exposure, y=recall, fill=valence)) +  
  geom_boxplot()  
p2box <- p1box +  
  labs(title="DV by Valence Type and Exposure", x="Exposure", y = "Words Recalled") +  
  theme_classic() +  
  theme(panel.grid.major.x = element_blank(),  
        panel.grid.major.y = element_blank(),  
        panel.grid.minor.x = element_blank(),  
        panel.grid.minor.y = element_blank(),  
        panel.background = element_blank(),  
        axis.line.y = element_line(colour="black", linewidth=.7),  
        axis.line.x = element_line(colour="black", linewidth=.7),  
        plot.title = element_text(hjust=.5))  
  ) +  
  scale_fill_manual(values=c('honeydew3','honeydew2', 'honeydew1'))  
p2box
```



A traditional bar graph can be created with `ggplot` tools. The choice of error bars is a topic of some discussion in the literature. Here, I chose to use an approach from Morey (2008) who recommended adjusted confidence intervals, where the adjustment recognizes the repeated measures nature of the design.

The general approach is to extract the descriptive info from the data frame, summarizing each condition and providing means, sem's, CI's etc. The `summarySEwithin` function provides the Morey-adjusted CI's. This summary information is then passed to `ggplot` for the bar graph.

Note that the column names of the summary data frame are re-specified since what comes out of `summarySEwithin` can be confusing.

```
summ <- Rmisc::summarySEwithin(data=abs1.df, idvar="snum",
                              measurevar="recall", withinvars=c("valence", "exposure"))
colnames(summ) <- c("valence", "exposure", "N", "mean", "sd", "sem", "CI" )
summ
```

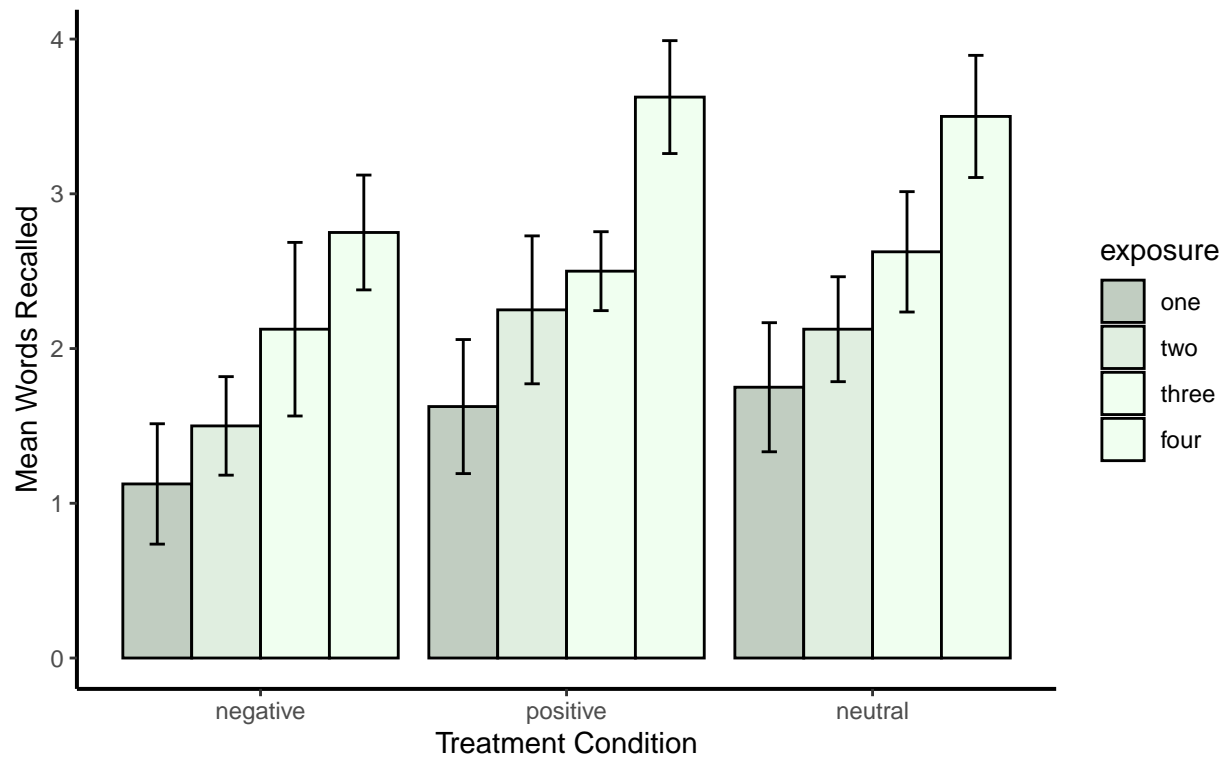
```
##   valence exposure N  mean      sd      sem      CI
## 1 negative     one  8 1.125 0.4652421 0.1644879 0.3889521
## 2 negative     two  8 1.500 0.3808171 0.1346392 0.3183710
## 3 negative    three  8 2.125 0.6709817 0.2372279 0.5609547
## 4 negative    four  8 2.750 0.4438127 0.1569115 0.3710367
## 5 positive     one  8 1.625 0.5180717 0.1831660 0.4331188
## 6 positive     two  8 2.250 0.5716991 0.2021261 0.4779524
## 7 positive    three  8 2.500 0.3050797 0.1078619 0.2550530
## 8 positive     four  8 3.625 0.4364358 0.1543033 0.3648694
## 9 neutral      one  8 1.750 0.4989166 0.1763936 0.4171047
##10 neutral      two  8 2.125 0.4055887 0.1433972 0.3390806
##11 neutral     three  8 2.625 0.4652421 0.1644879 0.3889521
##12 neutral      four  8 3.500 0.4721692 0.1669370 0.3947433
```

This graphing code can serve as a template for $A \times B$ s designs with different numbers of levels than the 3×4 we have here.

```
# a traditional bar graph with Morey adjusted error bars as CI's
p1bar <- ggplot(data=summ, aes(x=valence, y=mean, fill=exposure)) +
  geom_bar(stat="identity", color="black",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=mean-CI, ymax=mean+CI), width=.2,
               position=position_dodge(.9))

#p1bar
p2bar <- p1bar +
  xlab("Treatment Condition") +
  ylab("Mean Words Recalled") +
  ggtitle("Two Factor Repeated Design\n Means plus adjusted 95%CI") +
# labs(title="DV by Valence Type and Exposure \n plus adjusted 95% CI's", x="Exposure", y = "Words Rec
  theme_classic() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.background = element_blank(),
        axis.line.y = element_line(colour="black", linewidth=.7),
        axis.line.x = element_line(colour="black", linewidth=.7),
        plot.title = element_text(hjust=.5)
  ) +
  scale_fill_manual(values=c('honeydew3', 'honeydew2', 'honeydew1', 'honeydew'))
p2bar
```

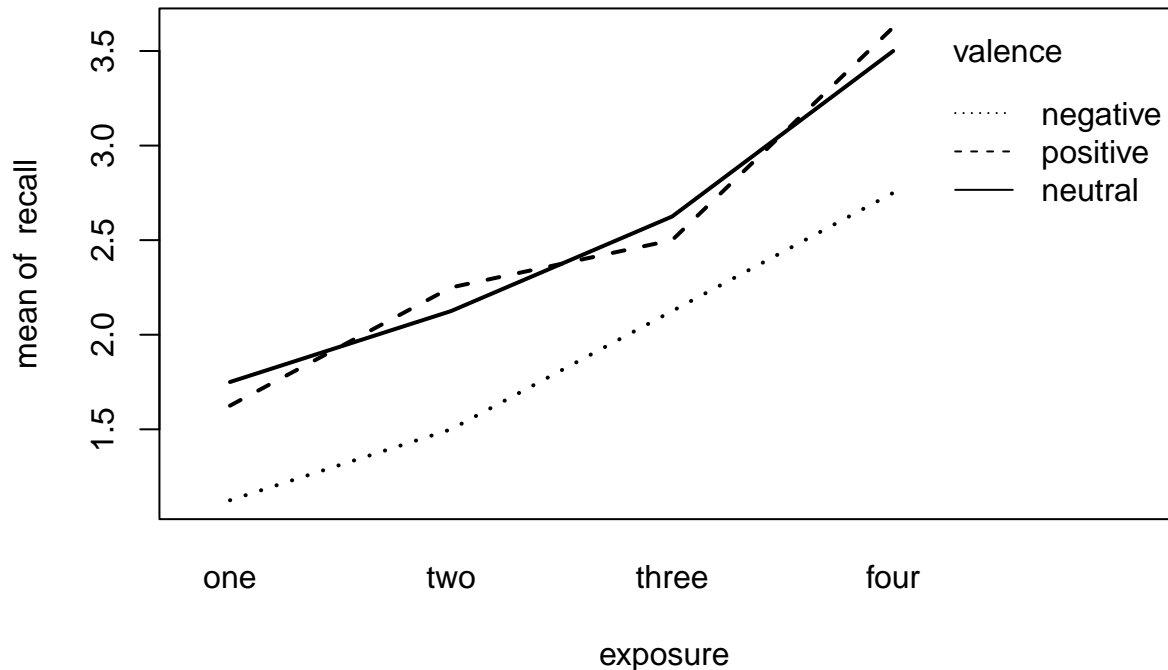
Two Factor Repeated Design Means plus adjusted 95%CI



The data set could also be depicted with a line graph since the exposure variable is levels spaced along a continuum. This also implies that orthogonal polynomials would be useful for the exposure variable.

This graph clearly reveals the lack of a valence by exposure interaction, but two main effects being present.

```
# a quick way to do this is the interaction.plot function  
# a quick way to visualize, but not a publication quality graph  
attach(abs1.df)  
interaction.plot(x.factor=exposure,  
                trace.factor=valence,  
                response=recall,  
                fixed=T,  
                lwd=2) # the so called interaction Plot
```



```
detach(abs1.df)
```

6 Basic Omnibus ANOVA

R has several capabilities for omnibus repeated measures analysis in this $A \times B \times s$ design.

6.1 Using aov

We begin a survey of the many ways to obtain the omnibus analysis by using the core `aov` function. In this type of analysis, `aov` requires that the data frame be in long format, as is the case with the data frame imported above.

In this analysis, we find that both main effects reach traditional levels of significance with alpha set at .05, but the interaction effect is too small to produce a null hypothesis rejection. This would imply that follow up analyses examining simple main effects and interaction contrasts would be unwarranted, but we have attempted to obtain them in later sections below anyway, since this document is a template for general analysis.

```
#perform the basic omnibus 2 factor anova
fit1.aov <- aov(recall ~ (valence*exposure) +
                Error(snum/(valence*exposure)),
                abs1.df)
summary(fit1.aov)
```

```
##
## Error: snum
##           Df Sum Sq Mean Sq F value Pr(>F)
```



```

## Residuals 7 39.17 5.595
##
## Error: snum:valence
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence   2  8.333  4.167  10.94 0.00138 **
## Residuals 14  5.333  0.381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## exposure   3 42.08 14.028  44.75 2.67e-09 ***
## Residuals 21  6.58  0.313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:valence:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence:exposure 6 0.667 0.1111  0.824 0.558
## Residuals       42 5.667 0.1349

```

Next we can try to create contrasts for each factor, valence first and then orthogonal polynomials for exposure.

```

# now create contrasts for factor a (valence)
# careful..... the order of the levels is alphabetical, so if you don't reorder.....
contrasts.v <- matrix(c(-1, -1, 2,
                       -1, 1, 0),ncol=2)
contrasts(abs1.df$valence) <- contrasts.v
contrasts(abs1.df$valence)
# and orthogonal polynomial contrasts for b (exposure)
contrasts.e <- contr.poly(4,scores=c(1,2,3,4)) # 4 levels with the values listed here
contrasts(abs1.df$exposure) <- contrasts.e
contrasts(abs1.df$exposure)

```

```

##           [,1] [,2]
## negative  -1   -1
## positive  -1    1
## neutral   2    0
##           .L .Q .C
## one    -0.6708204 0.5 -0.2236068
## two    -0.2236068 -0.5 0.6708204
## three   0.2236068 -0.5 -0.6708204
## four    0.6708204 0.5 0.2236068

```

The split function permits partitioning of main effects and interactions into single df contrasts, as specified above. HOWEVER, THERE IS AN ISSUE..... THIS SPLIT FUNCTION DOES NOT UTILIZE “SPECIFIC” ERROR TERMS, SO EACH CONTRAST IS TESTED BY ITS OMNIBUS ERROR TERM, AND MAY BE SUBJECT TO INFLATED ALPHA LEVELS PRODUCED BY NON-SPHERICITY.

Attempts to use specific error terms are described in later sections.

```

summary(fit1.aov, split=list("valence"=list(ac1=1,ac2=2),
                            "exposure"=list(lin=1,quad=2,cubic=3),
                            "valence:exposure"=list(ac1.L=1,ac2.L=2,
                                                    ac1.Q=3,ac2.Q=4,
                                                    ac1.C=5,ac2.C=6)))

```

```

##
## Error: snum
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  7  39.17   5.595
##
## Error: snum:valence
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence      2  8.333   4.167  10.938 0.00138 **
##   valence: ac1  1  6.250   6.250  16.406 0.00119 **
##   valence: ac2  1  2.083   2.083   5.469 0.03471 *
## Residuals     14  5.333   0.381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## exposure      3  42.08   14.03  44.747 2.67e-09 ***
##   exposure: lin  1  40.83   40.83 130.253 1.83e-10 ***
##   exposure: quad  1   1.04    1.04   3.323  0.0826 .
##   exposure: cubic  1   0.21    0.21   0.665  0.4241
## Residuals     21   6.58    0.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:valence:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence:exposure      6  0.667  0.1111  0.824 0.5582
##   valence:exposure: ac1.L  1  0.012  0.0125  0.093 0.7623
##   valence:exposure: ac2.L  1  0.104  0.1042  0.772 0.3846
##   valence:exposure: ac1.Q  1  0.063  0.0625  0.463 0.4999
##   valence:exposure: ac2.Q  1  0.021  0.0208  0.154 0.6963
##   valence:exposure: ac1.C  1  0.050  0.0500  0.371 0.5460
##   valence:exposure: ac2.C  1  0.417  0.4167  3.088 0.0861 .
## Residuals             42  5.667  0.1349
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

We can extract marginal and cell means from the aov fit object.

```
print(model.tables(fit1.aov, "means"), digits=3) #report the means of each level
```

```

## Tables of means
## Grand mean
##
## 2.291667
##
## valence
## valence
## negative positive neutral
## 1.875 2.500 2.500
##
## exposure
## exposure
## one two three four
## 1.50 1.96 2.42 3.29

```

```
##
## valence:exposure
##      exposure
## valence  one two three four
## negative 1.12 1.50 2.12 2.75
## positive 1.62 2.25 2.50 3.62
## neutral  1.75 2.12 2.62 3.50
```

It is also possible to extract deviation effects for each source of variance from the aov fit object.

```
print(model.tables(fit1.aov,"effects", n=TRUE),digits=3) # report deviation effects for each level
```

```
## Tables of effects
##
## valence
## valence
## negative positive neutral
## -0.417 0.208 0.208
##
## exposure
## exposure
## one two three four
## -0.792 -0.333 0.125 1.000
##
## valence:exposure
##      exposure
## valence  one two three four
## negative 0.0417 -0.0417 0.1250 -0.1250
## positive -0.0833 0.0833 -0.1250 0.1250
## neutral  0.0417 -0.0417 0.0000 0.0000
```

6.2 Base ANOVA using afex

The **afex** package provides many useful tools in addition to the base omnibus ANOVA capabilities that it has. For the repeated measures designs, this includes traditional df adjustments for non-sphericity as well as tests of sphericity.

The first analysis here simply reproduces the omnibus ANOVA create above, with aov. It also produces the sphericity tests and epsilon correction factors.

```
# control sphericity correction in anova_table: none, GG, HF
fit1.afex <- aov_car(recall ~ (valence*exposure)
+ Error(snum/(valence*exposure)),
anova_table = list(correction = "none"),
return="univariate",
data=abs1.df)
```

```
## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1
```

```
fit1.afex

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##      Sum Sq num Df Error SS den Df F value Pr(>F)
## (Intercept) 504.17 1 39.167 7 90.1064 3.013e-05 ***
## valence 8.33 2 5.333 14 10.9375 0.001378 **
```

```

## exposure          42.08      3    6.583      21 44.7468 2.674e-09 ***
## valence:exposure   0.67      6    5.667      42  0.8235  0.558192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##              Test statistic p-value
## valence          0.75000 0.42188
## exposure         0.69581 0.84124
## valence:exposure 0.02936 0.75148
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##              GG eps Pr(>F[GG])
## valence          0.8000  0.003349 **
## exposure         0.7901  9.841e-08 ***
## valence:exposure 0.5910  0.510493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              HF eps  Pr(>F[HF])
## valence          1.000000 1.378329e-03
## exposure         1.221274 2.674235e-09
## valence:exposure 1.272351 5.581922e-01

```

A repetition of the base analysis is accomplished here with a slightly differently formatted output that is the suitable format for using `emmeans` later.

```

fit1a.afex <- aov_car(recall ~ (valence*exposure)
+ Error(snum/(valence*exposure)),
  anova_table = list(correction = "none"),
  #return="univariate",
  data=abs1.df)
summary(fit1a.afex)

```

```

## Warning in summary.Anova.mlm(object$Anova, multivariate = FALSE): HF eps > 1
## treated as 1
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##              Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept)   504.17     1  39.167     7 90.1064 3.013e-05 ***
## valence         8.33     2   5.333    14 10.9375 0.001378 **
## exposure       42.08     3   6.583    21 44.7468 2.674e-09 ***
## valence:exposure 0.67     6   5.667    42  0.8235 0.558192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##

```

```
##              Test statistic p-value
## valence          0.75000 0.42188
## exposure         0.69581 0.84124
## valence:exposure 0.02936 0.75148
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##              GG eps Pr(>F[GG])
## valence          0.8000  0.003349 **
## exposure         0.7901  9.841e-08 ***
## valence:exposure 0.5910  0.510493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              HF eps  Pr(>F[HF])
## valence          1.000000 1.378329e-03
## exposure         1.221274 2.674235e-09
## valence:exposure 1.272351 5.581922e-01
```

It is possible to format table more neatly, still applying no sphericity correction.

```
gt(afex::nice(fit1a.afex, es=c("pes", "ges"), correction = "none"))
```

Effect	df	MSE	F	ges	pes	p.value
valence	2, 14	0.38	10.94 **	.128	.610	.001
exposure	3, 21	0.31	44.75 ***	.426	.865	<.001
valence:exposure	6, 42	0.13	0.82	.012	.105	.558

Next, the base analysis is repeated, this time employing the Greenhouse-Geisser correction. This uses the epsilon value for the correction rather than the more conservative lower-bound correction that is often taught in textbooks.

```
# control sphericity correction in anova_table: none, GG, HF
fit1b.afex <- aov_car(recall ~ (valence*exposure)
  + Error(snum/(valence*exposure)),
  anova_table = list(correction = "GG"),
  #return="univariate",
  data=abs1.df)
summary(fit1b.afex)
```

```
## Warning in summary.Anova.mlm(object$Anova, multivariate = FALSE): HF eps > 1
## treated as 1
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##              Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept)    504.17     1  39.167     7 90.1064 3.013e-05 ***
## valence         8.33      2   5.333    14 10.9375 0.001378 **
## exposure       42.08      3   6.583    21 44.7468 2.674e-09 ***
## valence:exposure 0.67      6   5.667    42  0.8235 0.558192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## valence           0.75000 0.42188
## exposure          0.69581 0.84124
## valence:exposure  0.02936 0.75148
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## valence           0.8000  0.003349 **
## exposure          0.7901  9.841e-08 ***
## valence:exposure  0.5910  0.510493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps  Pr(>F[HF])
## valence           1.000000 1.378329e-03
## exposure          1.221274 2.674235e-09
## valence:exposure  1.272351 5.581922e-01
```

An alternative is to produce a more nicely formatted table that uses the original, non-corrected, ANOVA but adds in the Greenhouse Geisser correction and requests two types of effect sizes (generalizes effect size and partial eta squared).

```
gt(afex::nice(fit1a.afex, es=c("pes", "ges"), correction = "GG"))
```

Effect	df	MSE	F	ges	pes	p.value
valence	1.60, 11.20	0.48	10.94 **	.128	.610	.003
exposure	2.37, 16.59	0.40	44.75 ***	.426	.865	<.001
valence:exposure	3.55, 24.82	0.23	0.82	.102	.105	.510

6.3 Base ANOVA using rstatix

A relatively new package, **rstatix** appears to offer a flexible approach. The `get_anova` function permits direct control of p value adjustment with GG, HF or none.

A tutorial is available:

<https://www.datanovia.com/en/lessons/repeated-measures-anova-in-r/#computation-1>

This first analysis requests the omnibus analysis with an extensive table that includes sphericity tests and both HF and GG corrections, as well as the uncorrected summary table.

```
fit1.rstatix <- anova_test(data=abs1.df,
                          dv=recall,
                          wid=snun,
                          within=c(valence,exposure))
fit1.rstatix
```

```
## ANOVA Table (type III tests)
##
```

```
## $ANOVA
##           Effect DFn DFd      F      p p<.05  ges
## 1          valence  2  14 10.937 1.00e-03 * 0.128
## 2          exposure  3  21 44.747 2.67e-09 * 0.426
## 3 valence:exposure  6  42  0.824 5.58e-01    0.012
##
## $`Mauchly's Test for Sphericity`
##           Effect      W      p p<.05
## 1          valence 0.750 0.422
## 2          exposure 0.696 0.841
## 3 valence:exposure 0.029 0.751
##
## $`Sphericity Corrections`
##           Effect  GGe      DF[GG]  p[GG] p[GG]<.05  HFe      DF[HF]
## 1          valence 0.800   1.6, 11.2 3.00e-03 * 1.000      2, 14
## 2          exposure 0.790 2.37, 16.59 9.84e-08 * 1.221 3.66, 25.65
## 3 valence:exposure 0.591 3.55, 24.82 5.10e-01    1.272 7.63, 53.44
##           p[HF] p[HF]<.05
## 1 1.00e-03 *
## 2 2.67e-09 *
## 3 5.58e-01
```

Use of the `get_anova_table` produces a shorter output with only the ANOVA table and permits GG or HF corrections.

```
get_anova_table(fit1.rstatix,correction="GG")
```

```
## ANOVA Table (type III tests)
##
##           Effect  DFn  DFd      F      p p<.05  ges
## 1          valence 1.60 11.20 10.937 3.00e-03 * 0.128
## 2          exposure 2.37 16.59 44.747 9.84e-08 * 0.426
## 3 valence:exposure 3.55 24.82  0.824 5.10e-01    0.012
```

6.4 Conclusions regarding base ANOVA implementation

The `afex` and `rstatix` approaches provide good capability for the omnibus analysis as well as the sphericity tests and correction methods. `afex` may be preferred because of its compatibility with the `emmeans` functions.

7 Begin working with `emmeans` for followup analyses

This section outlines follow up analyses including main effect and interaction contrasts as well as attempts at simple main effects and their contrasts.

7.1 Main effect contrasts and interaction contrasts

First, create grids of means (and other descriptive info) for each main effect and the full factorial arrangement of the 12 cells. We use the `afex` object that did not employ the “univariate” argument.

```
fit1.valence.emm <- emmeans::emmeans(fit1a.afex, specs=~valence)
fit1.valence.emm
```

```
## valence emmean  SE df lower.CL upper.CL
## negative  1.88 0.241  7    1.31    2.44
## positive  2.50 0.271  7    1.86    3.14
```

```
## neutral 2.50 0.259 7 1.89 3.11
##
## Results are averaged over the levels of: exposure
## Confidence level used: 0.95
```

```
fit1.exposure.emm <- emmeans::emmeans(fit1a.afex, specs=~exposure)
fit1.exposure.emm
```

```
## exposure emmean SE df lower.CL upper.CL
## one 1.50 0.199 7 1.03 1.97
## two 1.96 0.298 7 1.25 2.66
## three 2.42 0.258 7 1.81 3.03
## four 3.29 0.278 7 2.63 3.95
```

```
##
## Results are averaged over the levels of: valence
## Confidence level used: 0.95
```

```
fit1.cells.emm <- emmeans::emmeans(fit1a.afex, specs=~valence*exposure)
fit1.cells.emm
```

```
## valence exposure emmean SE df lower.CL upper.CL
## negative one 1.12 0.227 7 0.589 1.66
## positive one 1.62 0.263 7 1.003 2.25
## neutral one 1.75 0.250 7 1.159 2.34
## negative two 1.50 0.267 7 0.868 2.13
## positive two 2.25 0.366 7 1.385 3.12
## neutral two 2.12 0.295 7 1.427 2.82
## negative three 2.12 0.350 7 1.296 2.95
## positive three 2.50 0.267 7 1.868 3.13
## neutral three 2.62 0.263 7 2.003 3.25
## negative four 2.75 0.250 7 2.159 3.34
## positive four 3.62 0.324 7 2.859 4.39
## neutral four 3.50 0.327 7 2.726 4.27
```

```
##
## Confidence level used: 0.95
```

Establish an orthogonal contrast set for the two factors and test those main effect contrasts. First, the valence factor where a reverse helmert set makes sense with the neutral condition being a type of control condition that is pitted against the two treatments in the first contrast. The df for the t-tests reveal that the error terms must be specific to the contrast. I have manually worked this out and find that to be the case. This use of specific error terms when the `test` function is applied to “afex” objects is a capability new to `emmeans` in recent revisions. It is a welcome change from prior versions that used the flawed omnibus error term.

```
lincombs.valence <- contrast(fit1.valence.emm,
  list(valence_c1=c(-1,-1, 2),
       valence_c2=c(-1, 1, 0)
  ))
test(lincombs.valence, adjust="none")
```

```
## contrast estimate SE df t.ratio p.value
## valence_c1 0.625 0.299 7 2.092 0.0748
## valence_c2 0.625 0.134 7 4.677 0.0023
##
## Results are averaged over the levels of: exposure
```

Next, perform trend analysis on the exposure levels. Once again specific error terms are used.


```
lincombs.exposure <- contrast(fit1.exposure.emm,"poly")
test(lincombs.exposure, adjust="none")
```

```
## contrast estimate SE df t.ratio p.value
## linear 5.833 0.508 7 11.486 <.0001
## quadratic 0.417 0.216 7 1.930 0.0950
## cubic 0.417 0.541 7 0.770 0.4664
##
## Results are averaged over the levels of: valence
```

Next, create contrasts for interaction contrasts - a cumbersome approach....

```
# first, create vectors for each of the main effect contrast sets, in matrix form
mat.valence <- as.matrix(contrasts(abs1.df$valence))
mat.valence
```

```
##      [,1] [,2]
## negative -1 -1
## positive -1  1
## neutral  2  0
```

```
mat.exp <- as.matrix(contrasts(abs1.df$exposure))
mat.exp
```

```
##      .L .Q .C
## one -0.6708204 0.5 -0.2236068
## two -0.2236068 -0.5 0.6708204
## three 0.2236068 -0.5 -0.6708204
## four 0.6708204 0.5 0.2236068
```

```
# perform vector multiplication to obtain a matrix of coefficients which
# are then placed in a vector that is used in the contrast function
# for emmeans
```

```
intc1 <- mat.valence[,1]%*%t((mat.exp[,1]))
intc1 <- as.vector(intc1)
intc1
```

```
## [1] 0.6708204 0.6708204 -1.3416408 0.2236068 0.2236068 -0.4472136
## [7] -0.2236068 -0.2236068 0.4472136 -0.6708204 -0.6708204 1.3416408
```

```
intc2 <- mat.valence[,1]%*%t((mat.exp[,2]))
intc2 <- as.vector(intc2)
```

```
#intc2
```

```
intc3 <- mat.valence[,1]%*%t((mat.exp[,3]))
intc3 <- as.vector(intc3)
```

```
#intc3
```

```
intc4 <- mat.valence[,2]%*%t((mat.exp[,1]))
intc4 <- as.vector(intc4)
```

```
#intc4
```

```
intc5 <- mat.valence[,2]%*%t((mat.exp[,2]))
intc5 <- as.vector(intc5)
```

```
#intc5
```

```
intc6 <- mat.valence[,2]%*%t((mat.exp[,3]))
intc6 <- as.vector(intc6)
```

```
#intc6
```

Within the contrast function we can label the six vectors to be more directly interpretable.

```
lincombs.int <- contrast(fit1.cells.emm,
  list(val_c1xlin=intc1,
        val_c1xquad=intc2,
        val_c1xcubic=intc3,
        val_c2xlin=intc4,
        val_c2xquad=intc5,
        val_c2xcubic=intc6
  ))
```

Test the interaction contrasts with the `test` function. Note the capability for p value adjustment for multiple comparisons (no adjustment performed in this illustration). Once again, error terms for each of the six contrast are specific to that contrast, appropriately so.

```
test(lincombs.int, adjust="none")

## contrast      estimate      SE df t.ratio p.value
## val_c1xlin    -0.0559 0.335  7  -0.167  0.8721
## val_c1xquad   0.1250 0.280  7   0.447  0.6682
## val_c1xcubic  -0.1118 0.287  7  -0.390  0.7081
## val_c2xlin    0.1677 0.173  7   0.970  0.3645
## val_c2xquad   0.1250 0.206  7   0.607  0.5630
## val_c2xcubic   0.3354 0.198  7   1.692  0.1344
```

8 Simple Main effects and contrasts

When an interaction exists, it cannot be fully described unless both interaction contrasts (above) and Simple Main effects and their contrasts are examined. In the data set employed in this tutorial, the omnibus analysis failed to find evidence for an interaction so these follow up analyses would not be warranted. However, they are included here since the document can then serve as a template for other data sets where an interaction does exist.

8.1 Simple Main Effects

In a two-factor, factorial design, simple main effects are effects within a column or row of the design matrix. `emmeans` provides capabilities for simple main effects, but as you will see, issues arise in error term specification. The SME contrasts however, do appear to use correct error terms. This section (and the next) work through many different approaches; some work and some don't. A recommendation section is found at the end.

First, I just repeat the base ANOVA using `afex` in order to refresh memory about which object we are beginning with.

```
# again, the basic 3x4 ANOVA
fit1a.afex <- aov_car(recall ~ (valence*exposure)
  + Error(snum/(valence*exposure)),
  anova_table = list(correction = "none"),
  #return="univariate",
  data=abs1.df)

summary(fit1a.afex)

## Warning in summary.Anova.mlm(object$Anova, multivariate = FALSE): HF eps > 1
## treated as 1

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
```

```

##              Sum Sq num Df Error SS den Df F value   Pr(>F)
## (Intercept)    504.17     1   39.167     7 90.1064 3.013e-05 ***
## valence         8.33     2    5.333    14 10.9375 0.001378 **
## exposure       42.08     3    6.583    21 44.7468 2.674e-09 ***
## valence:exposure 0.67     6    5.667    42 0.8235 0.558192
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##              Test statistic p-value
## valence              0.75000 0.42188
## exposure              0.69581 0.84124
## valence:exposure      0.02936 0.75148
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##              GG eps Pr(>F[GG])
## valence          0.8000 0.003349 **
## exposure          0.7901 9.841e-08 ***
## valence:exposure 0.5910 0.510493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              HF eps  Pr(>F[HF])
## valence          1.000000 1.378329e-03
## exposure          1.221274 2.674235e-09
## valence:exposure 1.272351 5.581922e-01

```

Initially, we extract the `emmeans` grids, by levels of exposure, thus simple effects of valence AT levels of exposure.

```

#afex_options(emmeans:"multivariate")
simplesv.fit1a.emm <- emmeans::emmeans(fit1a.afex, "valence", by="exposure")
simplesv.fit1a.emm

```

```

## exposure = one:
## valence emmean SE df lower.CL upper.CL
## negative 1.12 0.227 7 0.589 1.66
## positive 1.62 0.263 7 1.003 2.25
## neutral 1.75 0.250 7 1.159 2.34
##
## exposure = two:
## valence emmean SE df lower.CL upper.CL
## negative 1.50 0.267 7 0.868 2.13
## positive 2.25 0.366 7 1.385 3.12
## neutral 2.12 0.295 7 1.427 2.82
##
## exposure = three:
## valence emmean SE df lower.CL upper.CL
## negative 2.12 0.350 7 1.296 2.95
## positive 2.50 0.267 7 1.868 3.13

```

```
## neutral    2.62 0.263  7    2.003    3.25
##
## exposure = four:
## valence  emmean    SE df lower.CL upper.CL
## negative  2.75 0.250  7    2.159    3.34
## positive  3.62 0.324  7    2.859    4.39
## neutral   3.50 0.327  7    2.726    4.27
##
## Confidence level used: 0.95
```

Immediately, we can see that the df for the error terms is unexpected. There are two possible error terms: valence by subject or valence by subject at exposure “one”. Either of those error terms would be expected to have 2 and 14 df. The denominator df of 7 here is perplexing. Since SS for each effect (and the error term) are not presented we cannot sort out how `emmeans` is accomplishing these tests - **a major problem**. The F values also do not match analyses that I’ve done manually. The numerator df are correct, but that is all.

```
test(contrast(simplesv.fit1a.emm), joint=TRUE, adjust=none)
```

```
## exposure df1 df2 F.ratio p.value note
## one      2   7   5.367 0.0386 d
## two      2   7  10.500 0.0078 d
## three    2   7   1.797 0.2344 d
## four     2   7  13.300 0.0041 d
##
## d: df1 reduced due to linear dependence
```

Next we look at the other set of SME, exposure at levels of valence.

```
simplese.fit1a.emm <- emmeans::emmeans(fit1a.afex, "exposure", by="valence")
simplese.fit1a.emm
```

```
## valence = negative:
## exposure emmean    SE df lower.CL upper.CL
## one      1.12 0.227  7    0.589    1.66
## two      1.50 0.267  7    0.868    2.13
## three    2.12 0.350  7    1.296    2.95
## four     2.75 0.250  7    2.159    3.34
##
## valence = positive:
## exposure emmean    SE df lower.CL upper.CL
## one      1.62 0.263  7    1.003    2.25
## two      2.25 0.366  7    1.385    3.12
## three    2.50 0.267  7    1.868    3.13
## four     3.62 0.324  7    2.859    4.39
##
## valence = neutral:
## exposure emmean    SE df lower.CL upper.CL
## one      1.75 0.250  7    1.159    2.34
## two      2.12 0.295  7    1.427    2.82
## three    2.62 0.263  7    2.003    3.25
## four     3.50 0.327  7    2.726    4.27
##
## Confidence level used: 0.95
```

The same problem arises with confusing df and F values here. So, neither set of simple main effects appears to be calculated correctly. At least the numerator df are correct, but that is about all.

```
test(contrast(simplese.fit1a.emm), joint=TRUE)
```

```
## valence df1 df2 F.ratio p.value note
## negative 3 7 36.333 0.0001 d
## positive 3 7 30.743 0.0002 d
## neutral 3 7 42.354 0.0001 d
##
## d: df1 reduced due to linear dependence
```

There is another way to obtain tests of SME that works well with BG designs. Maybe it will overcome the issue with error terms and error term df seen for the first way just demonstrated. We use the grid of means from all cells, extracted a few code chunks back and then employ the “simple=”each” ” argument.

Unfortunately, this produces the same questionable result.

Conclusion: SME obtained with emmeans in this AxBxs design are problematic.

```
test(contrast(fit1.cells.emm, simple="each"), joint=TRUE, adjust="none")
```

```
## $`simple contrasts for valence`
## exposure df1 df2 F.ratio p.value note
## one      2  7  5.367 0.0386 d
## two      2  7 10.500 0.0078 d
## three    2  7  1.797 0.2344 d
## four     2  7 13.300 0.0041 d
##
## d: df1 reduced due to linear dependence
##
## $`simple contrasts for exposure`
## valence df1 df2 F.ratio p.value note
## negative 3  7 36.333 0.0001 d
## positive 3  7 30.743 0.0002 d
## neutral  3  7 42.354 0.0001 d
##
## d: df1 reduced due to linear dependence
```

8.2 Simple main effect contrasts

Both valence and exposure SME are now broken down into orthogonal contrasts, the same contrasts we applied to them in the main effect and interaction contrasts section.

8.2.1 Simple main effect contrasts using emmeans on an afex model

Since the “omnibus” SME were tested in a problematic way, then we might expect that contrasts on those SME would also be problematic. We will try to use the grid of means separating the set of SME of valence first and build contrasts that are identical to what we used for the main effect contrasts.

Surprisingly, these SME contrasts look correct, and used error terms specific to the contrasts. I double checked these results with manual computations and am confident in trusting them.

```
# simple contrasts? These look ok, I think
lincombs.valence.simplev <- contrast(simplesv.fit1a.emm,
                                   list(valence_c1=c(-1,-1, 2),
                                        valence_c2=c(-1, 1, 0)
                                   ))
test(lincombs.valence.simplev, adjust="none")
```

```
## exposure = one:
```

```
## contrast estimate SE df t.ratio p.value
## valence_c1 0.750 0.526 7 1.426 0.1970
## valence_c2 0.500 0.189 7 2.646 0.0331
##
## exposure = two:
## contrast estimate SE df t.ratio p.value
## valence_c1 0.500 0.267 7 1.871 0.1036
## valence_c2 0.750 0.164 7 4.583 0.0025
##
## exposure = three:
## contrast estimate SE df t.ratio p.value
## valence_c1 0.625 0.420 7 1.488 0.1803
## valence_c2 0.375 0.263 7 1.426 0.1970
##
## exposure = four:
## contrast estimate SE df t.ratio p.value
## valence_c1 0.625 0.324 7 1.930 0.0950
## valence_c2 0.875 0.227 7 3.862 0.0062
```

Similarly, on the other set of SME (valence effects at levels of exposure) the contrasts also look correctly tested (trend) with specific error terms.

```
lincombs.valence.simplee <- contrast(simplese.fit1a.emm, "poly")
test(lincombs.valence.simplee, adjust="none")
```

```
## valence = negative:
## contrast estimate SE df t.ratio p.value
## linear 5.50 0.598 7 9.203 <.0001
## quadratic 0.25 0.313 7 0.798 0.4512
## cubic -0.25 0.818 7 -0.306 0.7689
##
## valence = positive:
## contrast estimate SE df t.ratio p.value
## linear 6.25 0.901 7 6.934 0.0002
## quadratic 0.50 0.267 7 1.871 0.1036
## cubic 1.25 0.726 7 1.722 0.1287
##
## valence = neutral:
## contrast estimate SE df t.ratio p.value
## linear 5.75 0.526 7 10.929 <.0001
## quadratic 0.50 0.327 7 1.528 0.1705
## cubic 0.25 0.590 7 0.424 0.6845
```

8.2.2 Simple main effect contrasts on an aov model

It is worth exploring whether `emmeans` can work on an `aov` fit object, so first I have repeated the base `aov` analysis from early in this document.

```
fit1.aov <- aov(recall ~ (valence*exposure) +
               Error(snum/(valence*exposure)),
               abs1.df)
summary(fit1.aov)
```

```
##
## Error: snum
## Df Sum Sq Mean Sq F value Pr(>F)
```

```

## Residuals 7 39.17 5.595
##
## Error: snum:valence
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence    2  8.333   4.167  10.94 0.00138 **
## Residuals 14  5.333   0.381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## exposure   3 42.08  14.028  44.75 2.67e-09 ***
## Residuals 21  6.58   0.313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: snum:valence:exposure
##           Df Sum Sq Mean Sq F value Pr(>F)
## valence:exposure 6  0.667  0.1111  0.824  0.558
## Residuals       42  5.667  0.1349

```

Now, testing one set of simples (valence at levels of exposure), we employ the same method used above with the `afex_car` object.

Inexplicably, the df for the error terms are a fractional value, 43.27. It is not at all clear how this arose. **This approach should not be used.**

```

simplesv.fit1a.emm <- emmeans::emmeans(fit1.aov, "valence", by="exposure")

```

```

## Note: re-fitting model with sum-to-zero contrasts

```

```

simplesv.fit1a.emm

```

```

## exposure = one:
## valence emmean SE df lower.CL upper.CL
## negative 1.12 0.291 14.4 0.503 1.75
## positive 1.62 0.291 14.4 1.003 2.25
## neutral 1.75 0.291 14.4 1.128 2.37
##
## exposure = two:
## valence emmean SE df lower.CL upper.CL
## negative 1.50 0.291 14.4 0.878 2.12
## positive 2.25 0.291 14.4 1.628 2.87
## neutral 2.12 0.291 14.4 1.503 2.75
##
## exposure = three:
## valence emmean SE df lower.CL upper.CL
## negative 2.12 0.291 14.4 1.503 2.75
## positive 2.50 0.291 14.4 1.878 3.12
## neutral 2.62 0.291 14.4 2.003 3.25
##
## exposure = four:
## valence emmean SE df lower.CL upper.CL
## negative 2.75 0.291 14.4 2.128 3.37
## positive 3.62 0.291 14.4 3.003 4.25
## neutral 3.50 0.291 14.4 2.878 4.12

```

```
##
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
test(contrast(simplestv.fit1a.emm), joint=TRUE, adjust=none)

## exposure df1 df2 F.ratio p.value note
## one 2 43.27 4.455 0.0174 d
## two 2 43.27 6.576 0.0032 d
## three 2 43.27 2.758 0.0746 d
## four 2 43.27 9.121 0.0005 d
##
## d: df1 reduced due to linear dependence
```

8.3 Other Approaches to simple effects with phia, rstatix and emmeans

An attempt was made at using the **phia** tools on an aov model. This also failed and output is not shown. It appears that **phia** is challenged by this AxBxS design.

```
# fit1.aov <- aov(recall ~ (valence*exposure) +
#               Error(snum/(valence*exposure)),
#               abs1.df)
# summary(fit1.aov)
# testInteractions(fit1.aov, fixed="valence", across="exposure", adjust="none")
# str(fit1.aov)
```

rstatix model objects are also not compatible with **emmeans**.

```
## explore rstatix approach - not supported directly with emmeans
fit1.statix <- anova_test(data=abs1.df,
                        dv=recall,
                        wid=snum,
                        within=c(valence, exposure)
                        )
get_anova_table(fit1.statix)

# simplestv.fit1a.emm <- emmeans::emmeans(fit1.statix, "valence", by="exposure")
# simplestv.fit1a.emm
# test(contrast(simplestv.fit1a.emm, "eff"), joint=TRUE, adjust=none)
```

8.4 Manually subset the data to obtain a type of Simple Main Effect approach

One possibility for approaching SME is to subset the data frame. For example, a SME of exposure at a level of valence could be produced by subsetting the data frame such that only one level of exposure was examined at a time and the full 3x4 base ANOVA is not the model object worked on for SME.

8.4.1 First, use tidyverse methods for efficiency

I have used two tidyverse tools here to produce the set of simple main effects of exposure at levels of valence. One is to use the **group_by** function to essentially split the data set into three components, one for each level of valence. The second is to utilize the pipes operator to efficiently create the flow of analysis. The code should be read:

- begin with the full `abs1.df` data frame
- subset the data frame according to levels of valence, thus four new data sets
- perform a standard **rstatix** anova on each of those four subsets of data, giving four analyses of valence
- this flow of three functions has output sent to a new object, “sme.e” which is then shown.

An advantage of this is that the error terms for the SME of exposure are justifiable; they can be conceived as valence x subject @ a level of exposure, and df are correct. Another advantage is that a “GG” or “HF” correction could be employed.

```
sme.e <- abs1.df %>%
  group_by(valence) %>%
  anova_test(dv=recall, wid=snum, within=exposure) %>%
  get_anova_table(correction="none")
sme.e
```

```
## # A tibble: 3 x 8
##   valence Effect      DFn  DFd    F      p `p<.05` ges
##   <ord>   <chr>   <dbl> <dbl> <dbl> <dbl> <chr>   <dbl>
## 1 negative exposure    3    21  20.2 0.00000216 *    0.415
## 2 positive exposure    3    21  24.7 0.00000044 *    0.441
## 3 neutral  exposure    3    21  29.6 0.000000981 *    0.43
```

The analogous operational sequence is also possible for the other set of simple main effects, valence at levels of exposure.

```
sme.v <- abs1.df %>%
  group_by(exposure) %>%
  anova_test(dv=recall, wid=snum, within=valence) %>%
  get_anova_table(correction="none")
sme.v
```

```
## # A tibble: 4 x 8
##   exposure Effect      DFn  DFd    F      p `p<.05` ges
##   <ord>   <chr>   <dbl> <dbl> <dbl> <dbl> <chr>   <dbl>
## 1 one     valence    2    14  3.42 0.062    ""    0.146
## 2 two     valence    2    14 12.8 0.000699 "*"    0.136
## 3 three   valence    2    14  2.12 0.157    ""    0.068
## 4 four    valence    2    14 10.4 0.002    "*"    0.189
```

If it were possible for `emmeans` to operate on a `rstatix` model then it would be possible to include extraction of SME contrasts in this flow as well. Unfortunately, as we saw above, `emmeans` won’t work on model objects from `rstatix`

8.4.2 Subset the data frame for SME using base R methods

Traditional subsetting can be done with the `subset` function. Here, I have subset on the valence factor, making three data frames on which a 1-way ANOVA of exposure can be performed.

```
# Do subsetting traditional way, making new data frames

subvneg <- subset(abs1.df, valence=="negative")
subvpos <- subset(abs1.df, valence=="positive")
subvneu <- subset(abs1.df, valence=="neutral")
```

Now, we just run an ANOVA on exposure with the standard `afex` tools on each subset of the data. These analyses replicate the results just obtained with the `tidyverse` methods of subsetting and piping.

```
#first at negative
fit_exp.neg <- aov_car(recall ~ exposure
  + Error(snum/exposure),
  anova_table = list(correction = "none"),
  return="univariate",
  data=subvneg)
```

```
## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1
```

```
fit_exp.neg
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept) 112.50     1   13.00     7  60.577 0.0001086 ***
## exposure    12.25     3    4.25    21  20.177 2.159e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## exposure          0.61551 0.73774
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## exposure 0.82336 1.408e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps Pr(>F[HF])
## exposure 1.306918 2.15938e-06
```

```
#second at positive
```

```
fit_exp.pos <- aov_car(recall ~ exposure
                      + Error(snum/exposure),
                      anova_table = list(correction = "none"),
                      return="univariate",
                      data=subvpos)
```

```
## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1
```

```
fit_exp.pos
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept) 200.00     1   16.50     7  84.849 3.667e-05 ***
## exposure    16.75     3    4.75    21  24.684 4.404e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
```

```

##           Test statistic p-value
## exposure      0.44875 0.47433
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## exposure 0.76645 7.561e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps  Pr(>F[HF])
## exposure 1.162602 4.403681e-07

```

```

# third at neutral
fit_exp.neu <- aov_car(recall ~ exposure
                      + Error(snum/exposure),
                      anova_table = list(correction = "none"),
                      return="univariate",
                      data=subvneu)

```

```

## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1

```

```

fit_exp.neu

```

```

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept) 200.00     1  15.00     7 93.333 2.684e-05 ***
## exposure    13.75     3   3.25    21 29.615 9.806e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## exposure      0.81111 0.94584
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## exposure 0.86667 6.027e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps  Pr(>F[HF])
## exposure 1.424242 9.805573e-08

```

The other set of SME (valence at levels of exposure) can be obtained in the analogous manner:

```

subeone <- subset(abs1.df, exposure=="one")
subetwo <- subset(abs1.df, exposure=="two")
subethree <- subset(abs1.df, exposure=="three")
subefour <- subset(abs1.df, exposure=="four")

# first at exposure one
fit_val.one <- aov_car(recall ~ valence
                      + Error(snum/valence),
                      anova_table = list(correction = "none"),
                      return="univariate",
                      data=subeone)

## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1

fit_val.one

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value    Pr(>F)
## (Intercept)  54.00     1  6.6667     7 56.7000 0.0001339 ***
## valence       1.75     2  3.5833    14  3.4186 0.0618044 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## valence      0.7788 0.47236
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## valence 0.81887  0.07538 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps Pr(>F[HF])
## valence 1.035183 0.0618044

# now at exposure two
fit_val.two <- aov_car(recall ~ valence
                      + Error(snum/valence),
                      anova_table = list(correction = "none"),
                      return="univariate",
                      data=subetwo)

## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1

fit_val.two

##

```

```

## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value   Pr(>F)
## (Intercept) 92.042     1  14.9583     7 43.072 0.000315 ***
## valence      2.583     2   1.4167    14 12.765 0.000699 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## valence      0.83045 0.57272
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##           GG eps Pr(>F[GG])
## valence 0.85503  0.001457 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           HF eps  Pr(>F[HF])
## valence 1.104027 0.0006989624

```

```

# now at exposure three
fit_val.three <- aov_car(recall ~ valence
                        + Error(snum/valence),
                        anova_table = list(correction = "none"),
                        return="univariate",
                        data=subethree)

```

```

## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1

```

```
fit_val.three
```

```

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##           Sum Sq num Df Error SS den Df F value   Pr(>F)
## (Intercept) 140.167     1  11.1667     7 87.8657 3.272e-05 ***
## valence      1.083     2   3.5833    14  2.1163   0.1574
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##           Test statistic p-value
## valence      0.96052 0.88617
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity

```

```

##
##          GG eps Pr(>F[GG])
## valence 0.96202    0.1598
##
##          HF eps Pr(>F[HF])
## valence 1.319188  0.1573849
# now at exposure four
fit_val.four <- aov_car(recall ~ valence
                        + Error(snum/valence),
                        anova_table = list(correction = "none"),
                        return="univariate",
                        data=subefour)

## Warning in summary.Anova.mlm(Anova.out, multivariate = FALSE): HF eps > 1
## treated as 1

fit_val.four

##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##          Sum Sq num Df Error SS den Df F value   Pr(>F)
## (Intercept) 260.042     1 12.9583     7 140.473 6.91e-06 ***
## valence      3.583     2  2.4167    14 10.379 0.00172 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Mauchly Tests for Sphericity
##
##          Test statistic p-value
## valence      0.85612 0.62749
##
##
## Greenhouse-Geisser and Huynh-Feldt Corrections
## for Departure from Sphericity
##
##          GG eps Pr(>F[GG])
## valence 0.87422    0.002926 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          HF eps Pr(>F[HF])
## valence 1.14133 0.001719723

```

8.5 Recommendations regarding contrasts and simple main effects

Use of the `afex_car` omnibus model in conjunction with `emmeans` yielded correct results for main effect contrasts, interaction contrasts, and simple main effect contrasts. The only problem was with production of correctly tested Simple Main Effects.

In fact, the only way that reasonable SME analyses were obtained was when the subsetting methods were employed. These can be recommended, but will be inefficient in larger factorial designs.

So, I recommend using `afex_car` for the omnibus model and all contrasts, but switching to subsetting for SME, using either `afex_car` or `rstatix` in an approach that is a series of 1-way repeated measures analyses.

A bit inefficient, and there are still issues in whether the error terms that come out of such subsetting are comprehensively defensible. This is tricky territory....

8.6 Pairwise multiple comparison follow up analyses

Instead of using contrasts to follow up omnibus analyses, pairwise comparison approaches are possible - either multiple comparisons or paired t-test types of comparisons with bonferroni family types of p value adjustment. Some of these can be obtained with `emmeans`. This section is still under construction....

9 Now some mixed models analysis

I have explored several linear mixed effects models strategies and simply provide code here without explanation in this version of this tutorial document.

```
fit1a.lme <- lme(recall~valence*exposure,
                random=~1|snum/valence/exposure,
                method="REML",
                data=abs1.df)

fit1a.lme

## Linear mixed-effects model fit by REML
## Data: abs1.df
## Log-restricted-likelihood: -88.60043
## Fixed: recall ~ valence * exposure
##      (Intercept)      valence1      valence2      exposure.L
##      2.29166667      0.10416667      0.31250000      1.30437299
##      exposure.Q      exposure.C valence1:exposure.L valence2:exposure.L
##      0.20833333      0.09316950      -0.00931695      0.08385255
## valence1:exposure.Q valence2:exposure.Q valence1:exposure.C valence2:exposure.C
##      0.02083333      0.06250000      -0.01863390      0.16770510
##
## Random effects:
## Formula: ~1 | snum
##      (Intercept)
## StdDev:  0.6591843
##
## Formula: ~1 | valence %in% snum
##      (Intercept)
## StdDev:  0.2159329
##
## Formula: ~1 | exposure %in% valence %in% snum
##      (Intercept) Residual
## StdDev:  0.4032037 0.178525
##
## Number of Observations: 96
## Number of Groups:
##              snum      valence %in% snum
##              8      24
## exposure %in% valence %in% snum
##              96

anova(fit1a.lme)

##              numDF denDF  F-value p-value
## (Intercept)      1     63 90.10635 <.0001
```

```
## valence          2    14 10.93750  0.0014
## exposure         3    63 72.14286 <.0001
## valence:exposure 6    63  0.57143  0.7515
```

Now work on changing cov structure. First, the base model with more specificity

```
fit1.lme <- lme(recall~valence*exposure,
               random=list(snum=pdBlocked(list(~1, pdIdent(~valence-1),
                                                         pdIdent(~exposure-1))),
                          method="REML",
                          data=abs1.df)
summary(fit1.lme)
```

```
## Linear mixed-effects model fit by REML
## Data: abs1.df
##      AIC      BIC    logLik
## 203.8817 242.7748 -85.94086
##
## Random effects:
## Composite Structure: Blocked
##
## Block 1: (Intercept)
## Formula: ~1 | snum
##      (Intercept)
## StdDev:  0.6477979
##
## Block 2: valencenegative, valencepositive, valenceneutral
## Formula: ~valence - 1 | snum
## Structure: Multiple of an Identity
##      valencenegative valencepositive valenceneutral
## StdDev:      0.248008      0.248008      0.248008
##
## Block 3: exposureone, exposuretwo, exposurethree, exposurefour
## Formula: ~exposure - 1 | snum
## Structure: Multiple of an Identity
##      exposureone exposuretwo exposurethree exposurefour Residual
## StdDev:  0.2439753  0.2439753  0.2439753  0.2439753  0.3673154
##
## Fixed effects: recall ~ valence * exposure
##              Value Std.Error DF   t-value p-value
## (Intercept)  2.2916667 0.24142007 77  9.492445  0.0000
## valence1     0.1041667 0.04454355 77  2.338535  0.0220
## valence2     0.3125000 0.07715169 77  4.050462  0.0001
## exposure.L   1.3043730 0.11428991 77 11.412845  0.0000
## exposure.Q   0.2083333 0.11428991 77  1.822850  0.0722
## exposure.C   0.0931695 0.11428991 77  0.815203  0.4175
## valence1:exposure.L -0.0093169 0.05301741 77 -0.175734  0.8610
## valence2:exposure.L  0.0838525 0.09182884 77  0.913140  0.3640
## valence1:exposure.Q  0.0208333 0.05301741 77  0.392953  0.6954
## valence2:exposure.Q  0.0625000 0.09182884 77  0.680614  0.4982
## valence1:exposure.C -0.0186339 0.05301741 77 -0.351468  0.7262
## valence2:exposure.C  0.1677051 0.09182884 77  1.826279  0.0717
## Correlation:
##              (Intr) valnc1 valnc2 exps.L exps.Q exps.C v11:.L v12:.L
## valence1      0
```



```

## valence2          0      0
## exposure.L        0      0      0
## exposure.Q         0      0      0      0
## exposure.C         0      0      0      0      0
## valence1:exposure.L 0      0      0      0      0      0
## valence2:exposure.L 0      0      0      0      0      0      0
## valence1:exposure.Q 0      0      0      0      0      0      0      0
## valence2:exposure.Q 0      0      0      0      0      0      0      0
## valence1:exposure.C 0      0      0      0      0      0      0      0
## valence2:exposure.C 0      0      0      0      0      0      0      0
##                   v11:.Q v12:.Q v11:.C
## valence1
## valence2
## exposure.L
## exposure.Q
## exposure.C
## valence1:exposure.L
## valence2:exposure.L
## valence1:exposure.Q
## valence2:exposure.Q 0
## valence1:exposure.C 0      0
## valence2:exposure.C 0      0      0
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.53481880 -0.55059295  0.02043337  0.49914413  1.85957812
##
## Number of Observations: 96
## Number of Groups: 8

```

```
anova(fit1.lme)
```

```

##          numDF denDF  F-value p-value
## (Intercept)      1    77 90.10651 <.0001
## valence          2    77 10.93750 0.0001
## exposure         3    77 44.74679 <.0001
## valence:exposure  6    77  0.82353 0.5552

```

Compound symmetry produces same results as fit1.lme, CS is default.

```

# now assume compound symmetry:
fit2.lme <- lme(recall~valence*exposure,
                random=list(snum=pdBlocked(list(~1, pdCompSymm(~valence-1), pdCompSymm(~exposure-1))),
                method="REML",
                data=abs1.df)
summary(fit2.lme)

```

```

## Linear mixed-effects model fit by REML
## Data: abs1.df
##      AIC      BIC    logLik
## 207.8817 251.6364 -85.94086
##
## Random effects:
## Composite Structure: Blocked
##
## Block 1: (Intercept)

```

```

## Formula: ~1 | snum
## (Intercept)
## StdDev: 0.2515999
##
## Block 2: valencenegative, valencepositive, valenceneutral
## Formula: ~valence - 1 | snum
## Structure: Compound Symmetry
## StdDev Corr
## valencenegative 0.463576
## valencepositive 0.463576 0.714
## valenceneutral 0.463576 0.714 0.714
##
## Block 3: exposureone, exposurertwo, exposurethree, exposurefour
## Formula: ~exposure - 1 | snum
## Structure: Compound Symmetry
## StdDev Corr
## exposureone 0.5123176
## exposurertwo 0.5123176 0.773
## exposurethree 0.5123176 0.773 0.773
## exposurefour 0.5123176 0.773 0.773 0.773
## Residual 0.3673155
##
## Fixed effects: recall ~ valence * exposure
## Value Std.Error DF t-value p-value
## (Intercept) 2.2916667 0.24142022 77 9.492439 0.0000
## valence1 0.1041667 0.04454354 77 2.338536 0.0220
## valence2 0.3125000 0.07715167 77 4.050463 0.0001
## exposure.L 1.3043730 0.11428985 77 11.412851 0.0000
## exposure.Q 0.2083333 0.11428985 77 1.822851 0.0722
## exposure.C 0.0931695 0.11428985 77 0.815204 0.4175
## valence1:exposure.L -0.0093169 0.05301742 77 -0.175734 0.8610
## valence2:exposure.L 0.0838525 0.09182886 77 0.913139 0.3640
## valence1:exposure.Q 0.0208333 0.05301742 77 0.392953 0.6954
## valence2:exposure.Q 0.0625000 0.09182886 77 0.680614 0.4982
## valence1:exposure.C -0.0186339 0.05301742 77 -0.351467 0.7262
## valence2:exposure.C 0.1677051 0.09182886 77 1.826279 0.0717
## Correlation:
## (Intr) valnc1 valnc2 exps.L exps.Q exps.C vl1:.L vl2:.L
## valence1 0
## valence2 0 0
## exposure.L 0 0 0
## exposure.Q 0 0 0 0
## exposure.C 0 0 0 0 0
## valence1:exposure.L 0 0 0 0 0 0
## valence2:exposure.L 0 0 0 0 0 0 0
## valence1:exposure.Q 0 0 0 0 0 0 0 0
## valence2:exposure.Q 0 0 0 0 0 0 0 0
## valence1:exposure.C 0 0 0 0 0 0 0 0
## valence2:exposure.C 0 0 0 0 0 0 0 0
## vl1:.Q vl2:.Q vl1:.C
## valence1
## valence2
## exposure.L
## exposure.Q

```

```
## exposure.C
## valence1:exposure.L
## valence2:exposure.L
## valence1:exposure.Q
## valence2:exposure.Q 0
## valence1:exposure.C 0 0
## valence2:exposure.C 0 0 0
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.53481943 -0.55059281 0.02043341 0.49914393 1.85957820
##
## Number of Observations: 96
## Number of Groups: 8
```

```
anova(fit2.lme)
```

```
##          numDF denDF  F-value p-value
## (Intercept)      1    77 90.10640 <.0001
## valence          2    77 10.93750 0.0001
## exposure         3    77 44.74684 <.0001
## valence:exposure  6    77  0.82353 0.5552
```

Try using lmer.

```
#library(lme4)
#library(lmerTest)
```

```
fit1.lmer <- lmer(recall~valence*exposure + (1|snum) + (1|valence:snum) + (1|exposure:snum), data=abs1.
```

```
summary(fit1.lmer)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: recall ~ valence * exposure + (1 | snum) + (1 | valence:snum) +
## (1 | exposure:snum)
## Data: abs1.df
##
## REML criterion at convergence: 171.9
##
## Scaled residuals:
##      Min      1Q   Median      3Q      Max
## -2.53482 -0.55059 0.02043 0.49914 1.85958
##
## Random effects:
## Groups      Name          Variance Std.Dev.
## exposure:snum (Intercept) 0.05952 0.2440
## valence:snum  (Intercept) 0.06151 0.2480
## snum          (Intercept) 0.41965 0.6478
## Residual                0.13492 0.3673
## Number of obs: 96, groups: exposure:snum, 32; valence:snum, 24; snum, 8
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  2.291667   0.241422  6.999881  9.492 3.01e-05 ***
## valence1     0.104167   0.044543 14.000178  2.339 0.03471 *
## valence2     0.312500   0.077151 14.000178  4.050 0.00119 **
## exposure.L   1.304373   0.114290 20.999966 11.413 1.83e-10 ***
```

```

## exposure.Q          0.208333  0.114290 20.999966  1.823  0.08260 .
## exposure.C          0.093169  0.114290 20.999966  0.815  0.42411
## valence1:exposure.L -0.009317  0.053017 41.999964 -0.176  0.86135
## valence2:exposure.L  0.083853  0.091829 41.999964  0.913  0.36638
## valence1:exposure.Q  0.020833  0.053017 41.999964  0.393  0.69634
## valence2:exposure.Q  0.062500  0.091829 41.999964  0.681  0.49985
## valence1:exposure.C -0.018634  0.053017 41.999964 -0.351  0.72699
## valence2:exposure.C  0.167705  0.091829 41.999964  1.826  0.07492 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) valnc1 valnc2 exps.L exps.Q exps.C vl1:.L vl2:.L vl1:.Q
## valence1  0.000
## valence2  0.000  0.000
## exposure.L 0.000  0.000  0.000
## exposure.Q 0.000  0.000  0.000  0.000
## exposure.C 0.000  0.000  0.000  0.000  0.000
## vlnc1:xps.L 0.000  0.000  0.000  0.000  0.000  0.000
## vlnc2:xps.L 0.000  0.000  0.000  0.000  0.000  0.000  0.000
## vlnc1:xps.Q 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## vlnc2:xps.Q 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## vlnc1:xps.C 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## vlnc2:xps.C 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##          vl2:.Q vl1:.C
## valence1
## valence2
## exposure.L
## exposure.Q
## exposure.C
## vlnc1:xps.L
## vlnc2:xps.L
## vlnc1:xps.Q
## vlnc2:xps.Q
## vlnc1:xps.C 0.000
## vlnc2:xps.C 0.000  0.000

```

```
anova(fit1.lmer)
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## valence      2.9514  1.4757     2    14 10.9376 0.001378 **
## exposure    18.1118  6.0373     3    21 44.7468 2.674e-09 ***
## valence:exposure 0.6667  0.1111     6    42  0.8235 0.558193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(fit1.lmer,ddf="lme4")
```

```

## Analysis of Variance Table
##          npar  Sum Sq Mean Sq F value
## valence      2  2.9514  1.4757 10.9376
## exposure     3 18.1118  6.0373 44.7468
## valence:exposure  6  0.6667  0.1111  0.8235

```

```
anova(fit1.lmer, valence=3)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## valence      2.9514  1.4757     2    14 10.9376  0.001378 **
## exposure    18.1118  6.0373     3    21 44.7468 2.674e-09 ***
## valence:exposure 0.6667  0.1111     6    42  0.8235  0.558193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
if(require(pbkrtest))
anova(fit1.lmer, ddf="Kenward-Roger")
```

```
## Loading required package: pbkrtest
```

```
## Type III Analysis of Variance Table with Kenward-Roger's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## valence      2.9514  1.4757     2    14 10.9376  0.001378 **
## exposure    18.1118  6.0373     3    21 44.7468 2.674e-09 ***
## valence:exposure 0.6667  0.1111     6    42  0.8235  0.558193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit1.lmer, ddf="Satterthwaite")
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## valence      2.9514  1.4757     2    14 10.9376  0.001378 **
## exposure    18.1118  6.0373     3    21 44.7468 2.674e-09 ***
## valence:exposure 0.6667  0.1111     6    42  0.8235  0.558193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10 Reproducibility

|Ver 1.0 Apr 17, 2023

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
```

```

## [1] pbkrtest_0.5.1 Rmisc_1.5.1      plyr_1.8.8      lattice_0.20-45
## [5] phia_0.2-1      rstatix_0.7.1   psych_2.2.9     nlme_3.1-161
## [9] lmerTest_3.1-3  knitr_1.41      gt_0.8.0        ez_4.4-0
## [13] emmeans_1.8.3  car_3.1-1       carData_3.0-5   afex_1.2-0
## [17] lme4_1.1-31    Matrix_1.5-3    forcats_0.5.2   stringr_1.5.0
## [21] dplyr_1.0.10   purrr_1.0.0     readr_2.1.3     tidyr_1.2.1
## [25] tibble_3.1.8   ggplot2_3.4.0   tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
## [1] TH.data_1.1-1      googledrive_2.0.0  minqa_1.2.5
## [4] colorspace_2.0-3  ellipsis_0.3.2     estimability_1.4.1
## [7] fs_1.5.2           rstudioapi_0.14    farver_2.1.1
## [10] fansi_1.0.3        mvtnorm_1.1-3      lubridate_1.9.0
## [13] xml2_1.3.3         codetools_0.2-18   splines_4.2.2
## [16] mnormt_2.1.1       jsonlite_1.8.4     nloptr_2.0.3
## [19] broom_1.0.2        dbplyr_2.2.1       compiler_4.2.2
## [22] httr_1.4.4         backports_1.4.1    assertthat_0.2.1
## [25] fastmap_1.1.0      gargle_1.2.1       cli_3.5.0
## [28] htmltools_0.5.4    tools_4.2.2        coda_0.19-4
## [31] gtable_0.3.1       glue_1.6.2         reshape2_1.4.4
## [34] Rcpp_1.0.9         cellranger_1.1.0   vctrs_0.5.1
## [37] xfun_0.36          rvest_1.0.3        timechange_0.1.1
## [40] lifecycle_1.0.3   googlesheets4_1.0.1 MASS_7.3-58.1
## [43] zoo_1.8-11         scales_1.2.1       hms_1.1.2
## [46] parallel_4.2.2    sandwich_3.0-2     yaml_2.3.6
## [49] stringi_1.7.8     highr_0.10         boot_1.3-28.1
## [52] rlang_1.0.6        pkgconfig_2.0.3    evaluate_0.19
## [55] labeling_0.4.2     tidyselect_1.2.0   magrittr_2.0.3
## [58] R6_2.5.1           generics_0.1.3     multcomp_1.4-20
## [61] DBI_1.1.3          pillar_1.8.1       haven_2.5.1
## [64] withr_2.5.0        mgcv_1.8-41        survival_3.4-0
## [67] abind_1.4-5        modelr_0.1.10      crayon_1.5.2
## [70] utf8_1.2.2         tzdb_0.3.0         rmarkdown_2.19
## [73] grid_4.2.2         readxl_1.4.1       reprex_2.0.2
## [76] digest_0.6.31     xtable_1.8-4       numDeriv_2016.8-1.1
## [79] munsell_0.5.0

```

References

- Boik, R. J. (1981). A priori tests in repeated measures designs: Effects of nonsphericity. *Psychometrika*, *46*, 241–255. Journal Article.
- Keppel, G., & Wickens, T. D. (2004). *Design and analysis : A researcher's handbook* (4th ed., pp. xii, 611 p.). Book, Upper Saddle River, N.J.: Pearson Prentice Hall.
- Morey, R. D. (2008). Confidence intervals from normalized data: A correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology*, *4*, 61–64. Journal Article.
- Rogan, J. C., Keselman, H. J., & Mendoza, J. L. (1979). Analysis of repeated measurements. *British Journal of Mathematical and Statistical Psychology*, *32*, 269–286. Journal Article.