Three-factor Between-group Analysis of Variance

Implementation in R

Bruce Dudek

2025-03-31

Contents

1 Three way factorial design. A completely randomized, "between subjects" design

This document is designed for students in the APSY510/511 introductory statistics sequence at the University at Albany. However, it should have much more general utility for other students and researchers learning to use R. A major strength of the document is presentation of extensive methods/logic of employing analytical/orthogonal contrasts both as an approach to obtaining the omnibus analysis and for direct evaluation.

Designs that are called "Completely Randomized Factorial Designs", presume assignment of cases to groups independently. They are sometimes called "between subjects" designs in the psychological sciences. For the design covered in this document, three independent variables are factorially arranged so that each level of each factor is found in combination with each level of the other factors.

The challenge of analyzing a 3 way factorial design is not in the specification and analysis of the omnibus model (`aov`, `lm`, afex`,`granova‘), it is in the follow up analyses.

With the use of the **phia** and **emmeans** packages we can obtain Simple 2-way interactions, Simple Simple Main Effects, Simple Main Effects, and orthogonally partitioned contrasts (and non-orthogonal ones) on all sources of variance, *e.g.*, 3-, and 2-way interaction contrasts, main effect contrasts, simple main effect contrasts.

The bulk of the presentation here is a traditional NHST framework.

This document still needs:

1. easier ways to compute effect sizes, especially for contrasts
2. bayesian and robust methods
3. more on post hoc tests

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. For some of the terminology, block diagram depecitons of a 3-way factorial are included for clarity. The relevant terminology/notation includes:

- *A fixed effects AxBxC design*: Three independent variables, perhaps called A, B, and C. Fully "between groups" design where participants/cases are measured only once on the outcome variable under the combined conditions of factors A, B, and C. All independent variables are fixed rather than random.
- *Levels*: The groups defined by each variable are called levels. E.g., in the primary example used in this document the "Grade" factor (factor C) has two "levels": fifth grade and twelfth grade.
- *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 3x3x2 factorial design (AxBxC) has 18 total combinations or "cells"

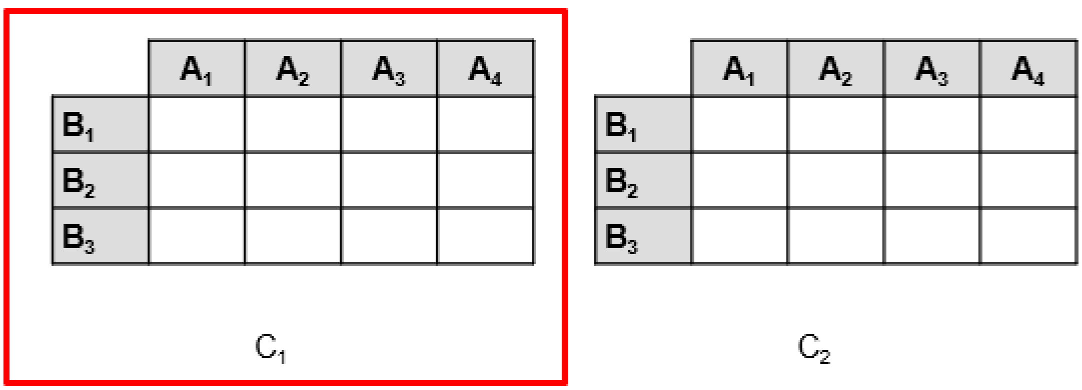
- *Cell means vs Marginal Means, Omnibus Effects*: Cells are defined by the three way combination of factors. The 3x3x2 design used in this document has 18 cells. But we can imaging "collapsing" on one or more IVs to produce marginal sets of means. For example collapsing on factor C produces an AxB matrix of means (9 means) and that would lead to assessment of the omnibus AxB interaction term. The omnibus terms in a 3 way factorial are the three main effects, the three two way interactions and the three way interaction, plus the pooled within group source (residual).

- *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" I find that this term is not generically used outside of the psychology experimental design textbook world or in the R ecosystem. But there is no alternative terminology so the "simple" phraseology strikes me as having a useful role even though a phrase like "simple, simple" may seem a bit silly. These can include:

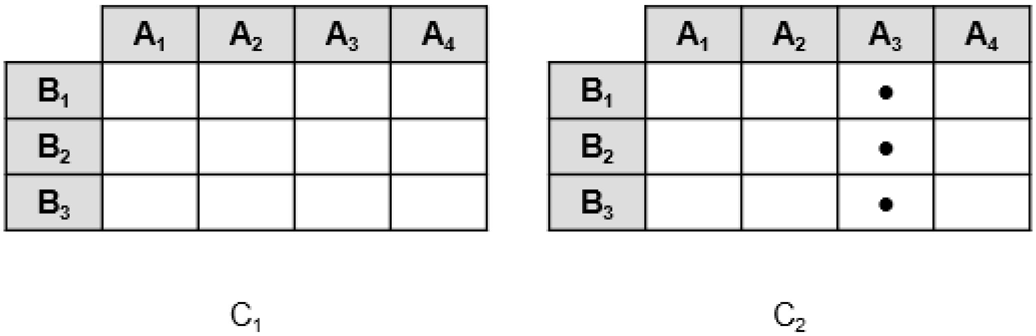
- *Simple interactions*: It is possible to ask a two way interaction question (e.g., AxB), not in an omnibus sense, but at a level of a third IV. For example - AxB at C1

Here is a block diagram depiction of a 3-way design that is a 4x2x2, thus 24 total cells. The red rectangle outlines the part of the study where only participants treated with C1 are. We could imagine a 2-way interaction of A and B there. Thus it would be the simple 2-way interaction of A*B @ C1

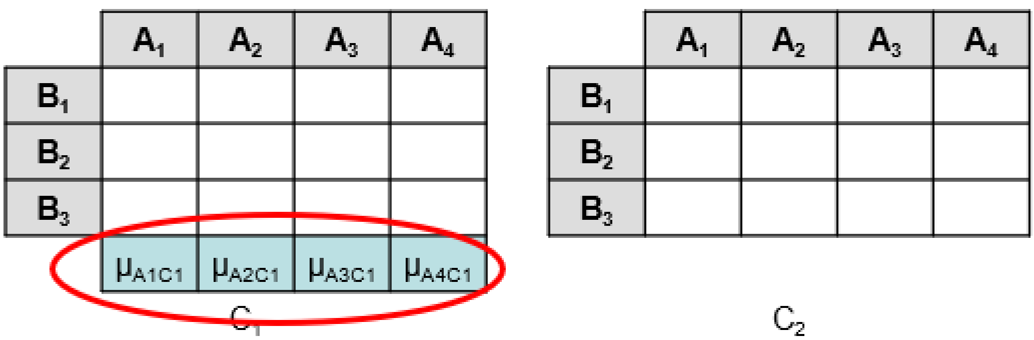


- *Simple Simple Main effects*: The term "main" implies the effect of one factor, but the word "simple" implies that the effect is examined at some level of one or more other factors. The textbook language of "simple-simple" implies that the effect of the one factor of interest (e.g., A) is examined at combined levels of **two** other factors. For example - B at A3,C2

Here is a depiction of a simple simple main effect of B at A3,C2. The effect involves the variation of the three means whose cells are indicated by the black dots - it is a one-dimensional (1-way) question about factor A, located in a specific section of the full block diagram.



- *Simple Main effects*: effects of one IV at levels of another factor, but **collapsed** on the third. E.g., A at B1 or C 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. The word "collapse" indicates that the grid of means is averaged over a third factor. Depicted with the block diagram, we can see the simple main effect of factor A is examined collapsed over B (the marginals), but only in the C1-treated participants. Thus A @ C1 implies collapsing over factor B



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

- A three way interaction is a test of whether simple 2 way interactions differ across levels of a 3rd variable, e.g., AxB at C1 is not the same as AxB at C2
- A simple 2 way interaction is a test of whether simple simple main effects differ.
- An omnibus 2 way interaction tests whether simple main effects differ.
- A 3 way interaction contrast tests whether simple 2way interaction contrasts differ across levels of a third factor.
- 2 way interaction contrasts test whether simple main effect contrasts differ.

When designs such as this are fully factorial, between-groups designs, the error term for all of the sources above would be MSwithin.

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-driven *a priori* hypotheses and from the recognized value of full rank models in ANOVA work.

2 R setup

Quite a few packages are used in this document. In many code chunks, where a the package origin of a function is not described or obvious, I use the `pkgname::functionname` syntax when a function is called. For example `psych::describeBy` calls the `describeBy function from the **psych** package. This format is not used when functions come from base system packages. One can always ask for help on a function` (?functionname‘) to establish the package of its origin if I’ve been inconsistent in this syntactical phrasing.

```
library(afex)
library(bcdstats) #available from bcdudek on Github
library(car)
library(effectsize)
library(emmeans)
library(ggplot2)
library(ggthemes)
library(ggrain)
library(gt)
library(knitr)
library(lmtest)
library(nortest)
library(phia)
library(plyr)
library(psych)
library(Rmisc)
library(sciplot)
library(sjstats)
library(tibble)
```

3 Data Definition

The data are from Keppel and Wickens, pg 466 data set, a 3x3x2 design it is the "keppel_3way_pg466.csv" file. The dependent variable is number of words recalled from a memorized list. Three independent variables were involved and factorially arranged. Participants were either fifth or twelfth graders. They were assigned to one of three Feedback conditions: control (none), praise, or negative. The words were one of three types: "LF_LE", "HF_LE", or "HF_HE". The factorial arrangement of these three factors thus produced a total of eighteen cells in the design.

The data set was explicitly chosen because the effects in the model are subtle and a simple story does not emerge. But it does serve as a useful illustration of how to extract a very wide range of analyses from this 3way factorial design.

In most illustrations in this document, the emphasis is more on how to perform the myriad analyses rather than on interpretation of the outcome of this textbook data set.

Here, the data are loaded into a data frame.

```
bg.3way <- read.csv("keppel_3way_pg466.csv",header=T, stringsAsFactors=TRUE)
```

Examine the contents of a few lines of the data frame.

```
head(bg.3way)
```

```
  snum feedback wordtype grade numrecall
1    1     none    LF_LE fifth         7
2    2     none    LF_LE fifth         7
3    3     none    LF_LE fifth         9
4    4     none    LF_LE fifth        10
5    5     none    LF_LE fifth         9
6    6     none    HF_LE fifth         7
```

Also check the structure of the data frame to verify that the IV's are factors.

```
str(bg.3way)
```

```
'data.frame':   90 obs. of  5 variables:
 $ snum     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ feedback : Factor w/ 3 levels "neg","none","pos": 2 2 2 2 2 2 2 2 2 2 ...
 $ wordtype : Factor w/ 3 levels "HF_HE","HF_LE",..: 3 3 3 3 3 2 2 2 2 2 ...
 $ grade    : Factor w/ 2 levels "fifth","twelfth": 1 1 1 1 1 1 1 1 1 1 ...
 $ numrecall: int  7 7 9 10 9 7 8 9 10 10 ...
```

3.1 Rework some data defninition characteristics

Some housekeeping with data definition that helps with formatting output/graphs is accomplished here.

- change values of feedback variable to whole words to facilitate better graph reading
- `recode` function in **car** is useful

```
bg.3way$feedback <- car::recode(bg.3way$feedback, "'none'='None'; 'pos'='Positive'; 'neg'='N
```

Next, I reorder the levels of two of the factors so that they are not the default alphabetical that R uses. The new orders match how we worked with these variables in other software. These orders are useful in graph labeling and in construction of contrast vectors.

```
bg.3way$feedback <- factor(bg.3way$feedback,
                    levels=c("None","Positive","Negative"))
bg.3way$wordtype <- factor(bg.3way$wordtype,
                    levels=c("LF_LE","HF_LE","HF_HE"))
```

Check that the orders are as requested:

```
levels(bg.3way$feedback)
```

```
[1] "None"     "Positive" "Negative"
```

```
levels(bg.3way$wordtype)
```

```
[1] "LF_LE" "HF_LE" "HF_HE"
```

Convert the snum variable from a numeric to a factor for better compatibility with **afex** functions.

```
bg.3way$snum <- as.factor(bg.3way$snum)
```

In order to make writing code easier for the graphing functions, the data frame is attached here. Prior to the ANOVA work it will be detached.

```
attach(bg.3way)
```

Look at contents of the final version of the data frame

```r
psych::headTail(bg.3way,8,8)
```

```
    snum feedback wordtype   grade numrecall
1      1     None    LF_LE   fifth         7
2      2     None    LF_LE   fifth         7
3      3     None    LF_LE   fifth         9
4      4     None    LF_LE   fifth        10
5      5     None    LF_LE   fifth         9
6      6     None    HF_LE   fifth         7
7      7     None    HF_LE   fifth         8
8      8     None    HF_LE   fifth         9
... <NA>     <NA>     <NA>    <NA>       ...
83    83 Negative    HF_LE twelfth         6
84    84 Negative    HF_LE twelfth         9
85    85 Negative    HF_LE twelfth         9
86    86 Negative    HF_HE twelfth         6
87    87 Negative    HF_HE twelfth         5
88    88 Negative    HF_HE twelfth         7
89    89 Negative    HF_HE twelfth         8
90    90 Negative    HF_HE twelfth         9
```

4 Exploratory Data Analysis

Numeric and graphical summaries are obtained in this section.

4.1 Numerical summaries of the dependent variable, by group.

First, use the psych package to look at descriptives. In order to obtain nicer looking tables, it required some relabeling and splitting of the object produced by `describeBy` into two separate tables - with nicerformatting by using `gt`.

```r
des1 <- describeBy(numrecall, list(wordtype,feedback,grade),mat=T,type=2, data=bg.3way)
row.names(des1) <- NULL
colnames(des1)[2] <- "Wordtype"
colnames(des1)[3] <- "Feedback"
colnames(des1)[4] <- "Grade"
colnames(des1)[10] <- "Trimmed_mean"
gt::gt(des1[,c(2:4,5:9)])
```

Wordtype	Feedback	Grade	vars	n	mean	sd	median
LF_LE	None	fifth	1	5	8.4	1.341641	9
HF_LE	None	fifth	1	5	8.8	1.303840	9
HF_HE	None	fifth	1	5	8.0	1.414214	7
LF_LE	Positive	fifth	1	5	7.8	1.303840	8
HF_LE	Positive	fifth	1	5	8.0	1.224745	8
HF_HE	Positive	fifth	1	5	4.4	1.341641	5
LF_LE	Negative	fifth	1	5	8.0	1.414214	8
HF_LE	Negative	fifth	1	5	7.6	1.341641	7
HF_HE	Negative	fifth	1	5	3.8	1.303840	4
LF_LE	None	twelfth	1	5	8.4	1.140175	8
HF_LE	None	twelfth	1	5	8.8	1.483240	9
HF_HE	None	twelfth	1	5	7.8	1.303840	8
LF_LE	Positive	twelfth	1	5	8.0	1.581139	8
HF_LE	Positive	twelfth	1	5	8.2	1.643168	9
HF_HE	Positive	twelfth	1	5	7.4	1.341641	8
LF_LE	Negative	twelfth	1	5	8.4	1.341641	9
HF_LE	Negative	twelfth	1	5	8.0	1.224745	8
HF_HE	Negative	twelfth	1	5	7.0	1.581139	7

```r
gt::gt(des1[,c(2:4,10:17)])
```

4.2 Draw a few graphs to examine the cell means.

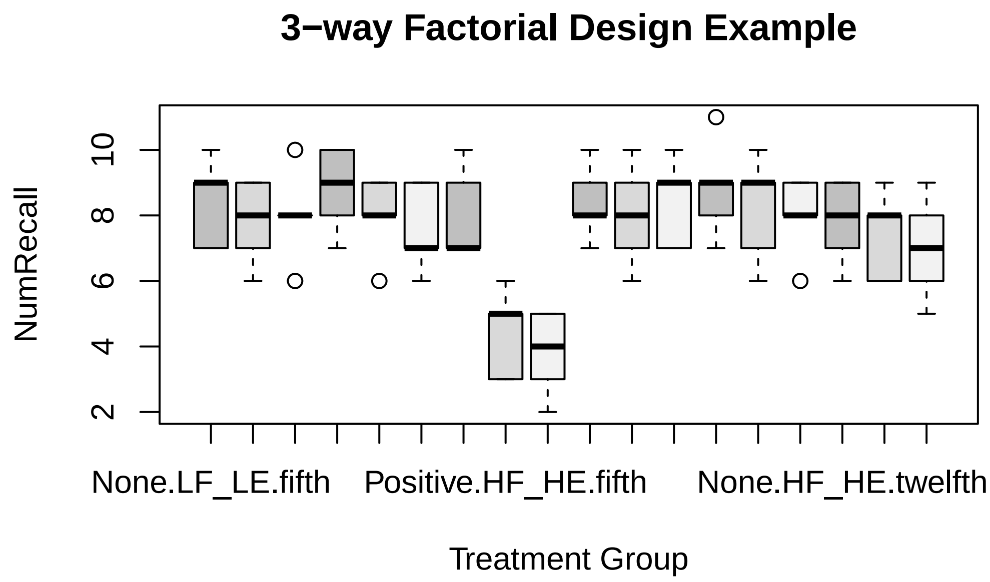
Graphical exploration of several types

4.2.1 First, a clustered box plot

Use the standard `boxplot` function from base R. Notice how the labels are not always fully visible. It is possible to use other functions to angle the labels so they can be seen - not worked out here. With the small N data set from a textbook, some of the plots are not particularly helpful with this data set, but I show them anyway since the primary goal here is a template for use with more realistic and larger data sets. Consequently, the boxplots have odd features in some groups.

Wordtype	Feedback	Grade	Trimmed_mean	mad	min	max	range	skew	kurtosi
LF_LE	None	fifth	8.4	1.4826	7	10	3	-0.1656347	-2.407407
HF_LE	None	fifth	8.8	1.4826	7	10	3	-0.5413871	-1.487889
HF_HE	None	fifth	8.0	0.0000	7	10	3	0.8838835	-1.750000
LF_LE	Positive	fifth	7.8	1.4826	6	9	3	-0.5413871	-1.487889
HF_LE	Positive	fifth	8.0	1.4826	6	9	3	-1.3608276	2.000000
HF_HE	Positive	fifth	4.4	1.4826	3	6	3	-0.1656347	-2.407407
LF_LE	Negative	fifth	8.0	0.0000	6	10	4	0.0000000	2.000000
HF_LE	Negative	fifth	7.6	1.4826	6	9	3	0.1656347	-2.407407
HF_HE	Negative	fifth	3.8	1.4826	2	5	3	-0.5413871	-1.487889
LF_LE	None	twelfth	8.4	1.4826	7	10	3	0.4047960	-0.177514
HF_LE	None	twelfth	8.8	1.4826	7	11	4	0.5516181	0.867768
HF_HE	None	twelfth	7.8	1.4826	6	9	3	-0.5413871	-1.487889
LF_LE	Positive	twelfth	8.0	1.4826	6	10	4	0.0000000	-1.200000
HF_LE	Positive	twelfth	8.2	1.4826	6	10	4	-0.5184205	-1.687242
HF_HE	Positive	twelfth	7.4	1.4826	6	9	3	-0.1656347	-2.407407
LF_LE	Negative	twelfth	8.4	1.4826	7	10	3	-0.1656347	-2.407407
HF_LE	Negative	twelfth	8.0	1.4826	6	9	3	-1.3608276	2.000000
HF_HE	Negative	twelfth	7.0	1.4826	5	9	4	0.0000000	-1.200000

```r
boxplot(numrecall~feedback*wordtype*grade, data=bg.3way,col=c("gray75","gray85","gray95"),
        main="3-way Factorial Design Example",
        xlab="Treatment Group",
        ylab="NumRecall")
```



A different approach to clustered boxplots can be implemented with **ggplot2**.

`ggplot` requires a summary of the data that is efficiently provided by the `summarySE` function. The column in the table labeled with the DV name is actually a column of cell means.

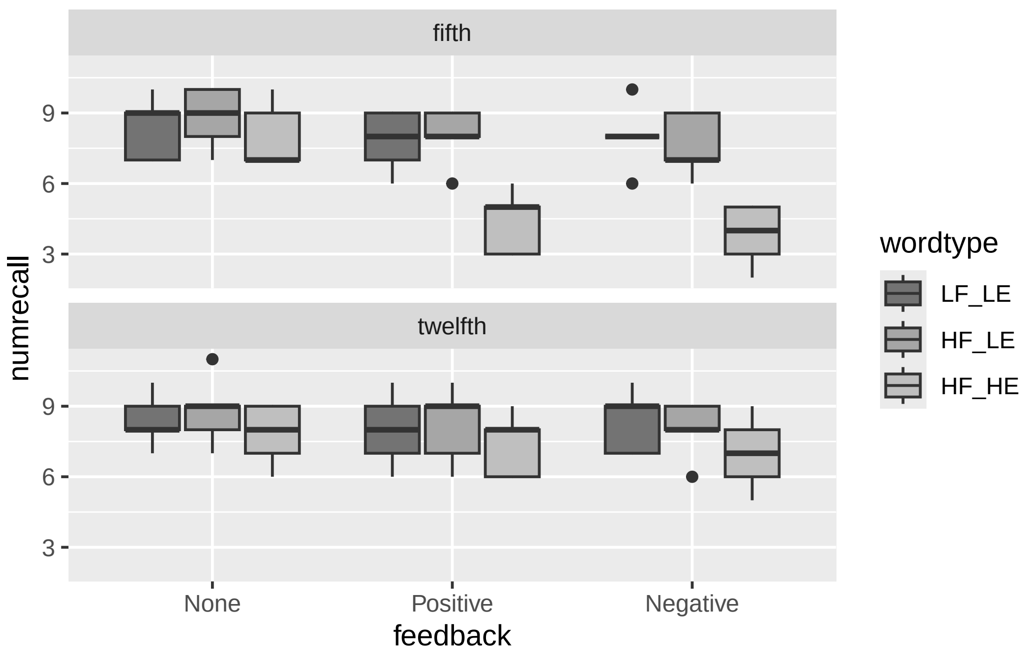
```
myData <- Rmisc::summarySE(data=bg.3way,measurevar="numrecall", groupvars=c("feedback", "wor
myData
```

```
   feedback wordtype   grade N numrecall       sd        se       ci
1      None    LF_LE   fifth 5       8.4 1.341641 0.6000000 1.665867
2      None    LF_LE twelfth 5       8.4 1.140175 0.5099020 1.415715
3      None    HF_LE   fifth 5       8.8 1.303840 0.5830952 1.618932
4      None    HF_LE twelfth 5       8.8 1.483240 0.6633250 1.841685
5      None    HF_HE   fifth 5       8.0 1.414214 0.6324555 1.755978
6      None    HF_HE twelfth 5       7.8 1.303840 0.5830952 1.618932
7  Positive    LF_LE   fifth 5       7.8 1.303840 0.5830952 1.618932
8  Positive    LF_LE twelfth 5       8.0 1.581139 0.7071068 1.963243
9  Positive    HF_LE   fifth 5       8.0 1.224745 0.5477226 1.520722
10 Positive    HF_LE twelfth 5       8.2 1.643168 0.7348469 2.040262
11 Positive    HF_HE   fifth 5       4.4 1.341641 0.6000000 1.665867
12 Positive    HF_HE twelfth 5       7.4 1.341641 0.6000000 1.665867
13 Negative    LF_LE   fifth 5       8.0 1.414214 0.6324555 1.755978
14 Negative    LF_LE twelfth 5       8.4 1.341641 0.6000000 1.665867
```

```
15 Negative    HF_LE   fifth 5       7.6 1.341641 0.6000000 1.665867
16 Negative    HF_LE twelfth 5       8.0 1.224745 0.5477226 1.520722
17 Negative    HF_HE   fifth 5       3.8 1.303840 0.5830952 1.618932
18 Negative    HF_HE twelfth 5       7.0 1.581139 0.7071068 1.963243
```

And now `ggplot` can create the clustered boxplot graph from that summary object. Since we have three IVs, we need to use the `facet_wrap` argument to split the plot into sections for the two levels of the third IV, grade.

```
p <- ggplot(data=bg.3way, aes(x=feedback, y=numrecall, fill=wordtype)) +
  geom_boxplot() +
  scale_fill_manual(values = c("grey45", "grey65", "grey75")) +
    facet_wrap(~grade,ncol=1)

p
```



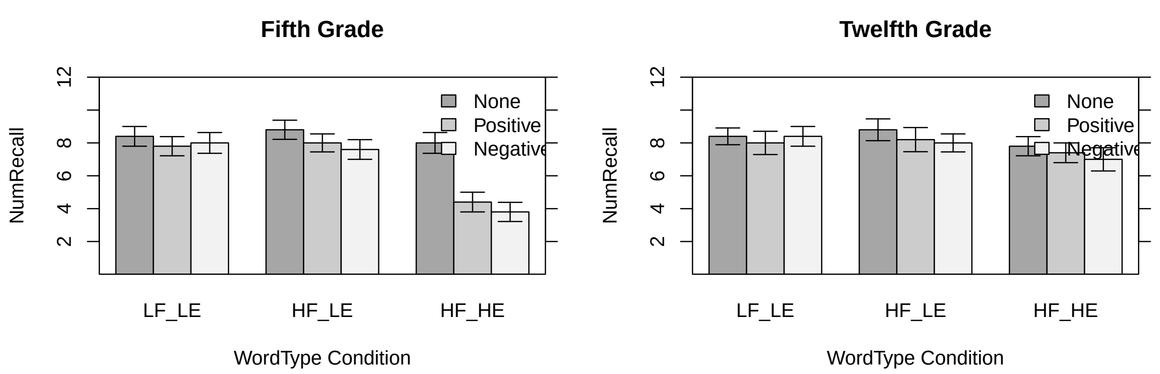
4.2.2 Bar graphs +/- std error bars

Several approaches to the standard bar graph are available. First, use the `bargraph.CI` function from the **sciplot** package. The key to producing side by side panels is to use the `screen` function.

```
# bar graphs +/- std error bars
split.screen(figs=c(1,2))
```

```
[1] 1 2
```

```
screen(1)
sciplot::bargraph.CI(wordtype,numrecall,group=feedback,lc=TRUE, uc=TRUE,legend=T,
            cex.leg=1,bty="n",col=c("gray65","gray80","gray95"),ylim=c(0.01,12),
            ylab="NumRecall",main="Fifth Grade",xlab="WordType Condition",
            data=subset(bg.3way,grade == "fifth"))
box()
axis(4,labels=F)
screen(2)
bargraph.CI(wordtype,numrecall,group=feedback,lc=TRUE, uc=TRUE,legend=T,
            cex.leg=1,bty="n",col=c("gray65","gray80","gray95"),ylim=c(0.01,12),
            ylab="NumRecall",main="Twelfth Grade",xlab="WordType Condition",
            data=subset(bg.3way,grade == "twelfth"))
box()
axis(4,labels=F)
```




```
close.screen(all = TRUE)
```

4.2.3 Bar Graphs with ggplot2

With **ggplot2** we have to produce an object that summarizes the data frame first - extracting means, std errors, and CI's. I did this with an **Rmisc** function.

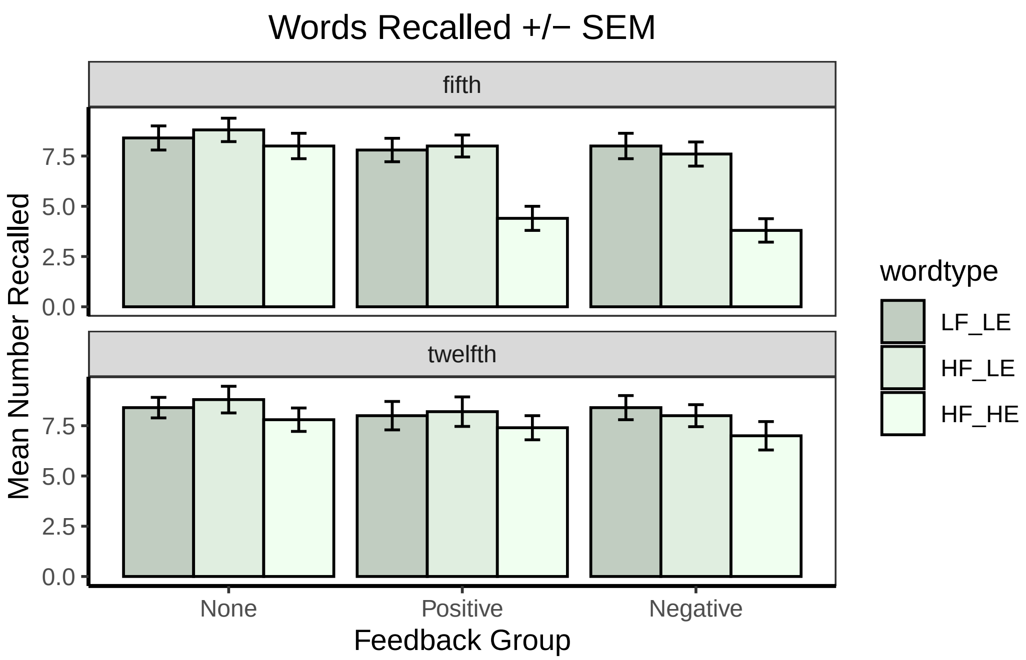
```
# first, summarize the data set to produce a new data frame that ggplot can use
# just repeating what was done above
myData <- Rmisc::summarySE(data=bg.3way,measurevar="numrecall", groupvars=c("feedback", "word
# look at the new data frame that contains the summary statistics
myData
```

```
   feedback wordtype   grade N numrecall       sd        se       ci
1      None    LF_LE   fifth 5       8.4 1.341641 0.6000000 1.665867
2      None    LF_LE twelfth 5       8.4 1.140175 0.5099020 1.415715
3      None    HF_LE   fifth 5       8.8 1.303840 0.5830952 1.618932
4      None    HF_LE twelfth 5       8.8 1.483240 0.6633250 1.841685
5      None    HF_HE   fifth 5       8.0 1.414214 0.6324555 1.755978
6      None    HF_HE twelfth 5       7.8 1.303840 0.5830952 1.618932
7  Positive    LF_LE   fifth 5       7.8 1.303840 0.5830952 1.618932
8  Positive    LF_LE twelfth 5       8.0 1.581139 0.7071068 1.963243
9  Positive    HF_LE   fifth 5       8.0 1.224745 0.5477226 1.520722
10 Positive    HF_LE twelfth 5       8.2 1.643168 0.7348469 2.040262
11 Positive    HF_HE   fifth 5       4.4 1.341641 0.6000000 1.665867
12 Positive    HF_HE twelfth 5       7.4 1.341641 0.6000000 1.665867
13 Negative    LF_LE   fifth 5       8.0 1.414214 0.6324555 1.755978
14 Negative    LF_LE twelfth 5       8.4 1.341641 0.6000000 1.665867
15 Negative    HF_LE   fifth 5       7.6 1.341641 0.6000000 1.665867
16 Negative    HF_LE twelfth 5       8.0 1.224745 0.5477226 1.520722
17 Negative    HF_HE   fifth 5       3.8 1.303840 0.5830952 1.618932
18 Negative    HF_HE twelfth 5       7.0 1.581139 0.7071068 1.963243
```

Now, use the myData object to draw the base plot (p1), and then add style control elements.

```
#library(ggplot2)
#library(ggthemes)
# Now create the Default bar plot
p1 <- ggplot(myData, aes(x=feedback, y=numrecall, fill=wordtype)) +
  geom_bar(stat="identity", color="black",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=numrecall-se, ymax=numrecall+se), width=.2,
                position=position_dodge(.9)) +
  facet_wrap(~grade,ncol=1)
```

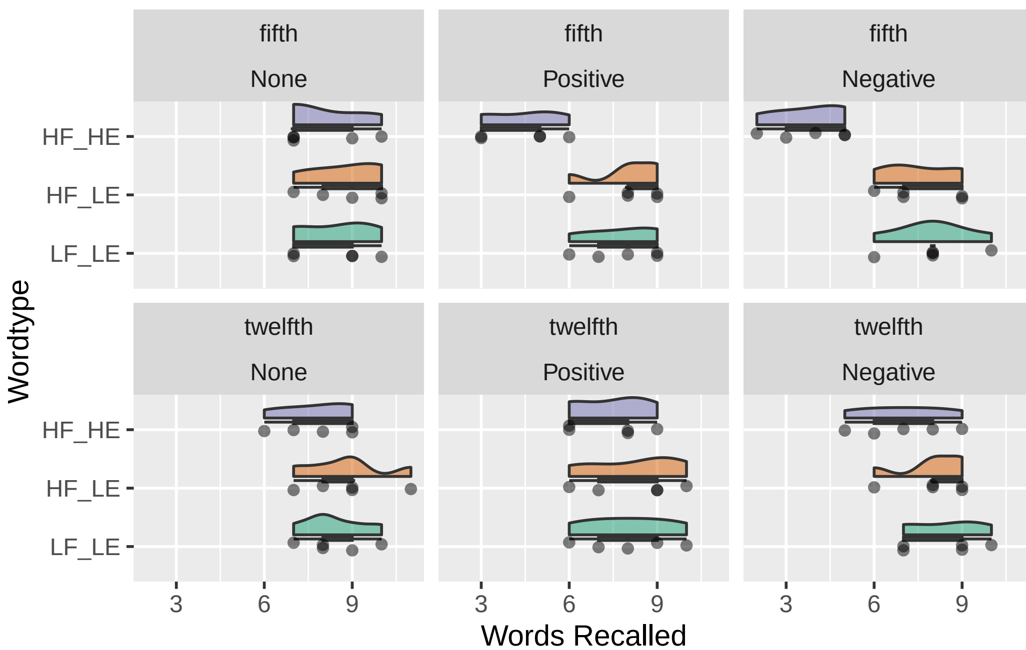
```
p2 <- p1  +labs(title="Words Recalled +/- SEM", x="Feedback Group", y = "Mean Number Recalled
  theme_bw() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        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'))
print(p2)
```



The textbook-derived data set used for the initial examples in this document has woefully inadequate sample sizes to be a good data set for illustrating several of these graphical approaches, including the boxplots and kernel density plots. The same is true of the favored "raincloud" plot, but the code is provided as a template anyway.

```
ggplot2::ggplot(bg.3way, aes(x=wordtype, y=numrecall, fill = wordtype)) +
  ggrain::geom_rain(alpha = .5,
            point.args.pos = rlang::list2(position =
```

```
             position_jitter(width = 0.07, height = 0)),
           boxplot.args.pos = list(
             width = 0.05, position = position_nudge(x = 0.13)),
           violin.args.pos = list(
             side = "r",
             width = 0.7, position = position_nudge(x = 0.2))) +
  theme_gray() +
  scale_fill_brewer(palette = 'Dark2') +
  guides(fill = 'none', color = 'none') +
  coord_flip()+
  facet_wrap(~grade + feedback, ncol=3) +
  xlab("Wordtype") + ylab("Words Recalled")
```



5 Perform the basic/omnibus 3 way ANOVA

First, detach the attached data frame since `aov` permits specification of the data frame name.

```
detach(bg.3way)
```

5.1 Use the `aov` function for the base omnibus analysis

Fitting the core model with `aov` follows a similar logic to that used for 2-way designs. The model formula could have been written the following way:

```
# code not run
fit_base.aov <- aov(numrecall~feedback + wordtype + grade +
                              feedback:wordtype +
                              feedback:grade +
                              wordtype:grade +
                              feedback:wordtype:grade,
                              data=bg.3way)
```

The model syntax shown above, naming each effect in the model, is inefficient when we want a full factorial analysis. It is more efficient to use the asterisk operator which tells `aov` to create all higher order terms from the main effects listed.

```
fit_base.aov <- aov(numrecall~feedback*wordtype*grade,data=bg.3way)
```

5.1.1 a different way to obtain tables of means and std errors

`model.tables` is an efficient function, but care must be taken in unbalanced designs to be certain of whether it produces weighted or unweighted marginal means (I believe that it gives weighted marginal means). In our example this doesn't matter since sample sizes are balanced.

```
model.tables(fit_base.aov,"means",se=T)
```

```
Tables of means
Grand mean

7.6
```

```
 feedback
feedback
    None Positive Negative
   8.367    7.300    7.133

 wordtype
wordtype
LF_LE HF_LE HF_HE
8.167 8.233 6.400

 grade
grade
  fifth twelfth
    7.2     8.0

 feedback:wordtype
          wordtype
feedback   LF_LE HF_LE HF_HE
  None     8.4   8.8   7.9
  Positive 7.9   8.1   5.9
  Negative 8.2   7.8   5.4

 feedback:grade
          grade
feedback   fifth twelfth
  None     8.400 8.333
  Positive 6.733 7.867
  Negative 6.467 7.800

 wordtype:grade
        grade
wordtype fifth twelfth
   LF_LE 8.067 8.267
   HF_LE 8.133 8.333
   HF_HE 5.400 7.400

 feedback:wordtype:grade
, , grade = fifth

          wordtype
feedback   LF_LE HF_LE HF_HE
  None     8.4   8.8   8.0
  Positive 7.8   8.0   4.4
```

```
  Negative 8.0   7.6   3.8

, , grade = twelfth

          wordtype
feedback   LF_LE HF_LE HF_HE
  None     8.4   8.8   7.8
  Positive 8.0   8.2   7.4
  Negative 8.4   8.0   7.0


Standard errors for differences of means
        feedback wordtype  grade feedback:wordtype feedback:grade
          0.3549   0.3549 0.2897            0.6146         0.5018
replic.       30       30     45                10             15
        wordtype:grade feedback:wordtype:grade
                0.5018                  0.8692
replic.             15                       5
```

5.1.2 Obtain ANOVA summary tables

The `summary` and `anova` functions produce the same summary table (Type I SS).

```r
summary(fit_base.aov)
```

```
                        Df Sum Sq Mean Sq F value   Pr(>F)    
feedback                 2  26.87   13.43   7.112  0.00152 ** 
wordtype                 2  64.87   32.43  17.171 7.99e-07 ***
grade                    1  14.40   14.40   7.624  0.00730 ** 
feedback:wordtype        4  14.67    3.67   1.941  0.11283    
feedback:grade           2   8.60    4.30   2.276  0.10999    
wordtype:grade           2  16.20    8.10   4.288  0.01740 *  
feedback:wordtype:grade  4  10.00    2.50   1.324  0.26946    
Residuals               72 136.00    1.89                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit_base.aov)
```

```
Analysis of Variance Table
```

```
Response: numrecall
                        Df  Sum Sq Mean Sq F value    Pr(>F)    
feedback                 2  26.867  13.433  7.1118  0.001519 ** 
wordtype                 2  64.867  32.433 17.1706 7.993e-07 ***
grade                    1  14.400  14.400  7.6235  0.007304 ** 
feedback:wordtype        4  14.667   3.667  1.9412  0.112829    
feedback:grade           2   8.600   4.300  2.2765  0.109987    
wordtype:grade           2  16.200   8.100  4.2882  0.017397 *  
feedback:wordtype:grade  4  10.000   2.500  1.3235  0.269461    
Residuals               72 136.000   1.889                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.1.3 Refit the model using `lm`

It is worth recalling that since `aov` is a wrapper for `lm`, we can we refit the model, using the `lm` function. This produces the same anova summary table as when we used `aov` above, so the residuals are the same. This approach will become useful when we examine analysis of contrasts, in unequal N designs.

```
# do the 3 way anova with the lm function
fit.1lm <- lm(numrecall~feedback*wordtype*grade,data=bg.3way)
#summary(fit.1lm)
anova(fit.1lm)
```

```
Analysis of Variance Table

Response: numrecall
                        Df  Sum Sq Mean Sq F value    Pr(>F)    
feedback                 2  26.867  13.433  7.1118  0.001519 ** 
wordtype                 2  64.867  32.433 17.1706 7.993e-07 ***
grade                    1  14.400  14.400  7.6235  0.007304 ** 
feedback:wordtype        4  14.667   3.667  1.9412  0.112829    
feedback:grade           2   8.600   4.300  2.2765  0.109987    
wordtype:grade           2  16.200   8.100  4.2882  0.017397 *  
feedback:wordtype:grade  4  10.000   2.500  1.3235  0.269461    
Residuals               72 136.000   1.889                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Effect	df	MSE	F	ges	p.value
feedback	2, 72	1.89	7.11 **	.127	.002
wordtype	2, 72	1.89	17.17 ***	.259	<.001
grade	1, 72	1.89	7.62 **	.078	.007
feedback:wordtype	4, 72	1.89	1.94	.073	.113
feedback:grade	2, 72	1.89	2.28	.046	.110
wordtype:grade	2, 72	1.89	4.29 *	.087	.017
feedback:wordtype:grade	4, 72	1.89	1.32	.054	.269

5.2 Use the afex package for the omnibus ANOVA

An alternative that provides some advantages is the suite of ANOVA tools from **afex**. `aov_car` gives type III SS and tests (by default), plus the ges effect size statistic.

Note that the "grade" factor is specified as an "observed" variable. This impacts the method of calculating the generalized effect sizes.

```
fit_base.afex <- aov_car(numrecall~feedback*wordtype*grade + Error(1|snum), type=3, observed=
gt::gt(nice(fit_base.afex))
```

6 Effect sizes for the Omnibus Analysis

The `anova_stats` function is a convenient way to obtain a multiplicity of effect sizes on the omnibus effects. However the downside of using it is that it will not work on an **afex** object. I show it here with the original `aov`fit. The issue (downside) is that the aov fit is likely based on Type I SS. Below this section, the **effectsize** package is use on the **afex** fit which employed Type III SS. In the equal N situation with the current data set there is no distinction, but for unbalanced designs, the **effectsize/afex** approach may be better.

```
# the gt function permits nicer formatting of the table
# anova_stats(fit_base.aov)
gt(anova_stats(fit_base.aov)[1:7,1:6],rownames_to_stub = T)
```

Using the **effectsize** package, we can compute eta squareds (and partial eta squareds) using the **afex** fit with type III SS. Another nice feature of the **effectsize** functions is that confidence intervals are provided for the effect size estimates.

	etasq	partial.etasq	omegasq	partial.omegasq	epsilonsq	cohens.f
feedback	0.092	0.165	0.079	0.120	0.079	0.444
wordtype	0.222	0.323	0.208	0.264	0.209	0.691
grade	0.049	0.096	0.043	0.069	0.043	0.325
feedback:wordtype	0.050	0.097	0.024	0.040	0.024	0.328
feedback:grade	0.029	0.059	0.016	0.028	0.017	0.251
wordtype:grade	0.056	0.106	0.042	0.068	0.043	0.345
feedback:wordtype:grade	0.034	0.068	0.008	0.014	0.008	0.271

Parameter	Cohens_f_partial	CI	CI_low	CI_high
feedback	0.4444649	0.95	0.17538133	0.6726748
wordtype	0.6906235	0.95	0.41823382	0.9372685
grade	0.3253957	0.95	0.08662186	0.5613204
feedback:wordtype	0.3283948	0.95	0.00000000	0.5140729
feedback:grade	0.2514663	0.95	0.00000000	0.4649734
wordtype:grade	0.3451342	0.95	0.05487871	0.5662212
feedback:wordtype:grade	0.2711631	0.95	0.00000000	0.4454608

```
gt::gt(effectsize::cohens_f(fit_base.afex, partial=TRUE, ci=.95, alternative="two"))
```

```
# or other effect sizes
#gt::gt(effectsize::cohens_f(fit_base.afex, partial=TRUE, ci=.95, alternative="two"))
#gt::gt(effectsize::omega_squared(fit_base.afex, partial=TRUE, ci=.95, alternative="two"))
```

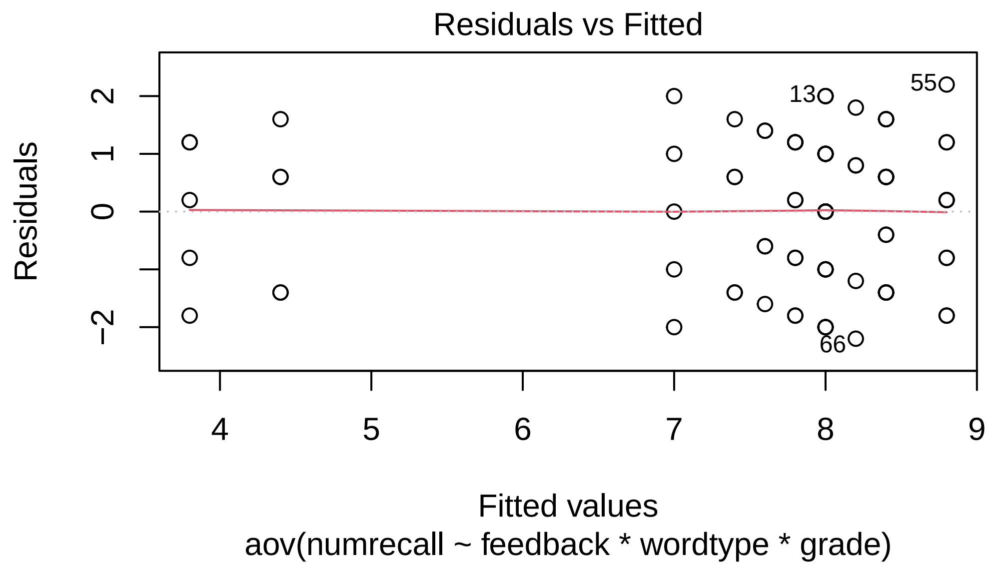
7 Evaluate Assumptions

The normality and homoscedasticity/homogeneity assumptions are evaluated in the standard manner we covered for other linear models and ANOVA objects.

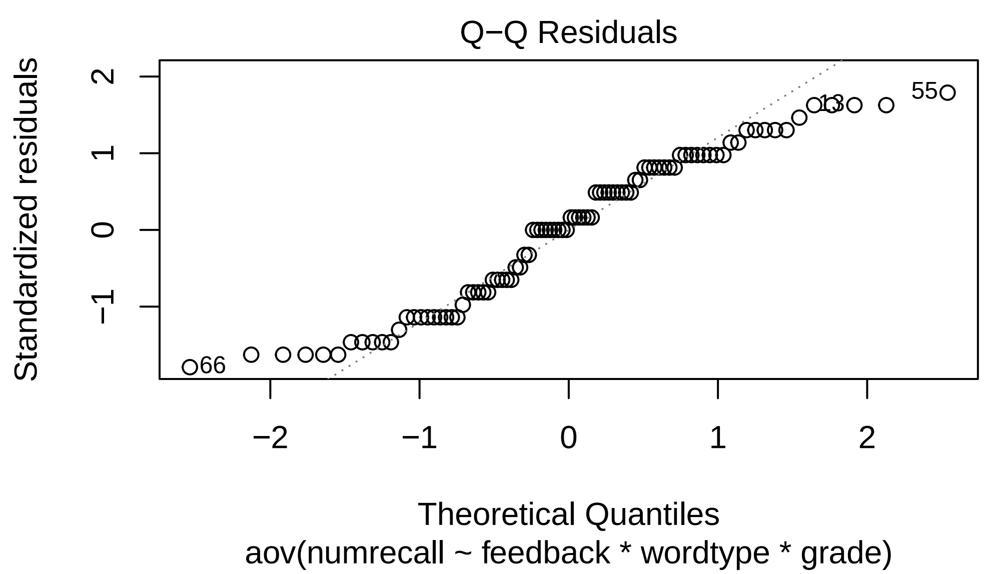
7.1 First, produce the diagnostic plots for linear models

The standard two plots for evaluating homoscedasticity and normality of the residuals are available in the same manner as for other linear model objects that we have seen previously.

```
plot(fit_base.aov,which=1)
```

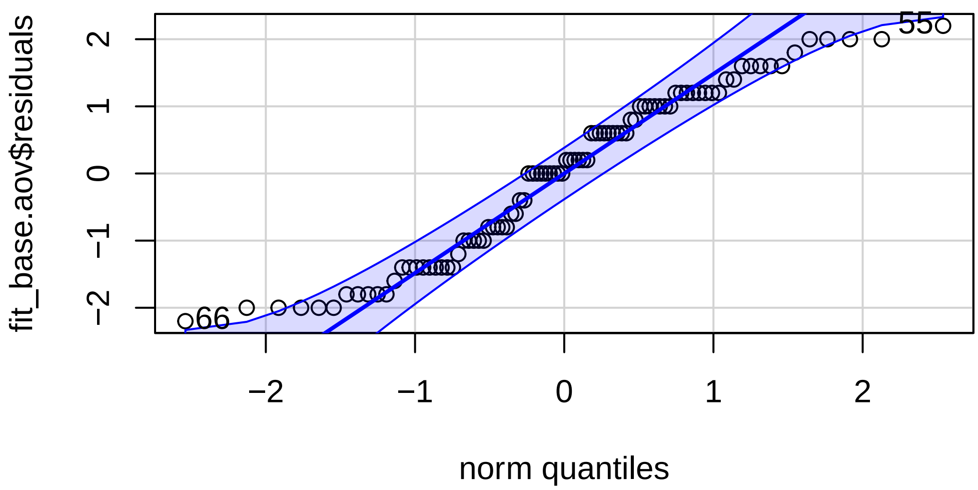



```
plot(fit_base.aov,which=2)
```



The `qqPlot` function from the **car** package produces a nice normal QQ plot.

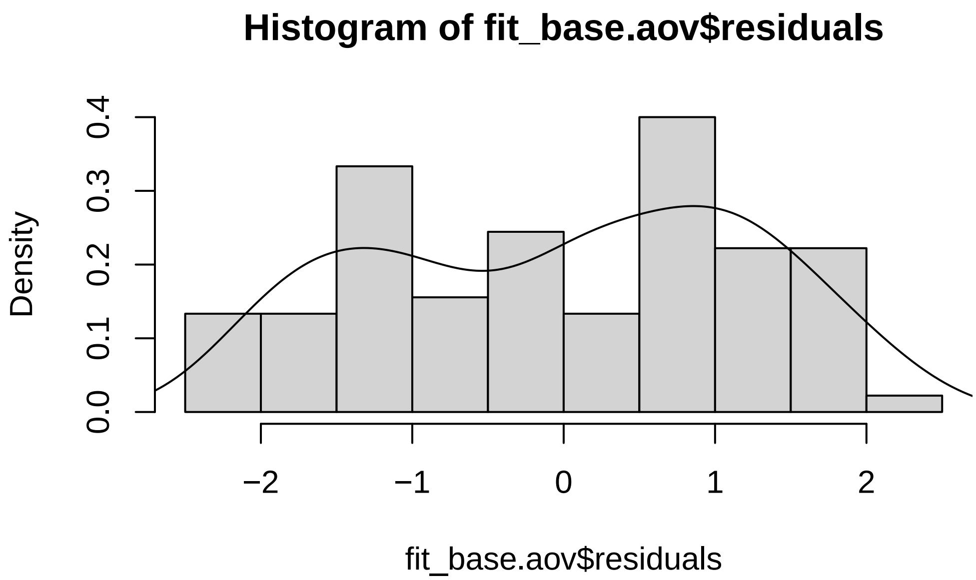
```
car::qqPlot(fit_base.aov$residuals)
```




```
[1] 66 55
```

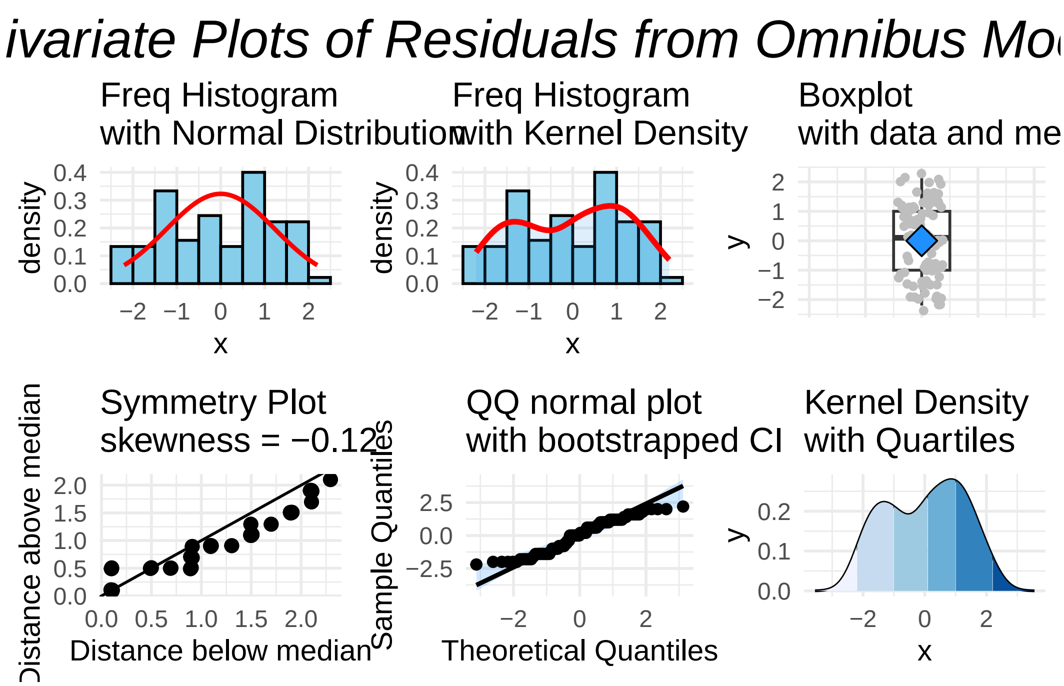
There is clearly some non-normality in the distribution of these residuals so let's examine a frequency histogram with a kernel density overlaid.

```
hist(fit_base.aov$residuals, breaks=10, prob=TRUE)
lines(density(fit_base.aov$residuals))
```



Or use the `explore` function from the **bcdstats** package on the `aov` fit object from above.

```
bcdstats::explore(residuals(fit_base.aov),varname="Residuals from Omnibus Model")
```



```
   vars  n mean   sd median trimmed  mad  min max range  skew kurtosis   se
X1    1 90    0 1.24    0.1    0.01 1.63 -2.2 2.2   4.4 -0.12     -1.2 0.13
```

7.2 Inferential tests of the Normality and Homogeneity of Variance assumptions

We can see some non-normality of the residuals from the graphical displays, so inferential tests might be in order. The Anderson-Darling test rejects the null hypothesis that residuals are normally distributed.

```
nortest::ad.test(residuals(fit_base.aov))
```

```

	Anderson-Darling normality test

data:  residuals(fit_base.aov)
A = 1.5133, p-value = 0.0006257
```

A Shapiro-Wilk test reaches the same conclusion.

```
shapiro.test(residuals(fit_base.aov))
```

```

	Shapiro-Wilk normality test

data:  residuals(fit_base.aov)
W = 0.94663, p-value = 0.001042
```

The Levene test for homogeneity of variance is obtained using the function from the **car** package. Very little variation in cell variances/stdevs was seen and this test was not significant.

```
car::leveneTest(fit_base.aov)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group 17  0.0897      1
      72
```

The Breusch Pagan test of homoscedasticity leads to a similar outcome, as does the non-constant variance test.

```
lmtest::bptest(fit_base.aov)
```

```

	studentized Breusch-Pagan test

data:  fit_base.aov
BP = 4.1174, df = 17, p-value = 0.9994
```

```
car::ncvTest(fit.1lm)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.008655078, Df = 1, p = 0.92588
```

Since the normality assumption appears to have been violated, this data set should be submitted to alternative approaches. DV scale transformations, bootstrapping, other robust analyses or Bayesian inference might be considered. These alternatives are not yet incorporated into this tutorial.

8 Orthogonal Contrasts on the Omnibus Effects

We will obtain partitioning of omnibus effects into their single df contrasts here and later use the **emmeans** and **phia** packages both for contrasts and simple effects.

8.1 Work with the `split` argument and the `summary.lm` function to obtain contrasts on omnibus effects.

The Feedback and Wordtype factors both have three levels and are now decomposed into a pair of orthogonal contrasts each. For Feedback, the first contrast compares the negative condition to the average of the other two and the second contrasts compares none and positive. For realistic applications of these methods, the choice of explicit contrasts should be made on * a priori* grounds. For this tutorial, with a textbook example, the choice was based on the fact that for feedback, group 3 was the control condition, so a 1-, -1, 2 contrast seemed appropriate. and the second one is the orthogonal one to that contrast.

```
contrasts.feedback <- matrix(c(-1,-1,2,-1,1,0),ncol=2)
contrasts(bg.3way$feedback) <- contrasts.feedback
contrasts(bg.3way$feedback)
```

```
         [,1] [,2]
None       -1   -1
Positive   -1    1
Negative    2    0
```

For the Wordtype variable, the first contrast compares LF_LE to the other two, and the second compares HF-LE to HF_HE. I chose this set somewhat arbitrarily since we don't have explicit information on theory that would drive contrasts for this factor in this textbook illustration.

```
contrasts.wordtype <- matrix(c(2,-1,-1,0,-1,1),ncol=2)
contrasts(bg.3way$wordtype) <- contrasts.wordtype
contrasts(bg.3way$wordtype)
```

```
      [,1] [,2]
LF_LE    2    0
HF_LE   -1   -1
HF_HE   -1    1
```

It is worth recalling that the two-level factor, "grade" can already be conceptualized as a contrast since it has only 1 df. However, the default coding for "grade" is still in place and that is dummy coding.

```
contrasts(bg.3way$grade)
```

```
        twelfth
fifth         0
twelfth       1
```

It is best to change that to "sum to zero" coding as well so that interpretation of regression coefficients and effects is straight forward.

```
contrasts.grade <- matrix(c(1,-1),ncol=1)
```

```
contrasts(bg.3way$grade) <- contrasts.grade
contrasts(bg.3way$grade)
```

```
        [,1]
fifth      1
twelfth   -1
```

I will redo the basic omnibus ANOVA here and the use the `split` function to obtain tests of contrasts.

```
fit.1aov <- aov(numrecall~feedback*wordtype*grade,data=bg.3way)
#summary(fit.1aov)
anova(fit.1aov)
```

```
Analysis of Variance Table

Response: numrecall
                        Df  Sum Sq Mean Sq F value    Pr(>F)    
feedback                 2  26.867  13.433  7.1118  0.001519 ** 
wordtype                 2  64.867  32.433 17.1706 7.993e-07 ***
grade                    1  14.400  14.400  7.6235  0.007304 ** 
feedback:wordtype        4  14.667   3.667  1.9412  0.112829    
feedback:grade           2   8.600   4.300  2.2765  0.109987    
wordtype:grade           2  16.200   8.100  4.2882  0.017397 *  
feedback:wordtype:grade  4  10.000   2.500  1.3235  0.269461    
Residuals               72 136.000   1.889                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The initial application of this approach includes partitioning of both the Feedback and Wordtype factors. Grade is only two levels, so requires no further decomposition. Notice that `summary` now produces all eleven single df contrasts among the twelve cells, breaking down main effects and interactions involving Feedback and Wordtype.

```
summary(fit.1aov,  split=list(feedback=list(feedback.ac1=1, feedback.ac2=2),
                              wordtype=list(wordtype.ac1=1, wordtype.ac2=2)))
```

```
                                                   Df Sum Sq Mean Sq F value
feedback                                            2  26.87   13.43   7.112
  feedback: feedback.ac1                            1   9.80    9.80   5.188
  feedback: feedback.ac2                            1  17.07   17.07   9.035
wordtype                                            2  64.87   32.43  17.171
  wordtype: wordtype.ac1                            1  14.45   14.45   7.650
  wordtype: wordtype.ac2                            1  50.42   50.42  26.691
grade                                               1  14.40   14.40   7.624
feedback:wordtype                                   4  14.67    3.67   1.941
  feedback:wordtype: feedback.ac1.wordtype.ac1      1   5.63    5.63   2.978
  feedback:wordtype: feedback.ac2.wordtype.ac1      1   2.41    2.41   1.275
```

```
  feedback:wordtype: feedback.ac1.wordtype.ac2        1   2.41    2.41   1.275
  feedback:wordtype: feedback.ac2.wordtype.ac2        1   4.22    4.22   2.237
feedback:grade                                        2   8.60    4.30   2.276
  feedback:grade: feedback.ac1                        1   3.20    3.20   1.694
  feedback:grade: feedback.ac2                        1   5.40    5.40   2.859
wordtype:grade                                        2  16.20    8.10   4.288
  wordtype:grade: wordtype.ac1                        1   4.05    4.05   2.144
  wordtype:grade: wordtype.ac2                        1  12.15   12.15   6.432
feedback:wordtype:grade                               4  10.00    2.50   1.324
  feedback:wordtype:grade: feedback.ac1.wordtype.ac1  1   0.63    0.63   0.331
  feedback:wordtype:grade: feedback.ac2.wordtype.ac1  1   1.88    1.88   0.993
  feedback:wordtype:grade: feedback.ac1.wordtype.ac2  1   1.88    1.88   0.993
  feedback:wordtype:grade: feedback.ac2.wordtype.ac2  1   5.62    5.62   2.978
Residuals                                            72 136.00    1.89
                                                       Pr(>F)
feedback                                              0.00152 **
  feedback: feedback.ac1                              0.02571 *
  feedback: feedback.ac2                              0.00364 **
wordtype                                             7.99e-07 ***
  wordtype: wordtype.ac1                              0.00721 **
  wordtype: wordtype.ac2                             2.05e-06 ***
grade                                                 0.00730 **
feedback:wordtype                                     0.11283
  feedback:wordtype: feedback.ac1.wordtype.ac1        0.08870 .
  feedback:wordtype: feedback.ac2.wordtype.ac1        0.26258
  feedback:wordtype: feedback.ac1.wordtype.ac2        0.26258
  feedback:wordtype: feedback.ac2.wordtype.ac2        0.13913
feedback:grade                                        0.10999
  feedback:grade: feedback.ac1                        0.19721
  feedback:grade: feedback.ac2                        0.09520 .
wordtype:grade                                        0.01740 *
  wordtype:grade: wordtype.ac1                        0.14747
  wordtype:grade: wordtype.ac2                        0.01338 *
feedback:wordtype:grade                               0.26946
  feedback:wordtype:grade: feedback.ac1.wordtype.ac1  0.56693
  feedback:wordtype:grade: feedback.ac2.wordtype.ac1  0.32243
  feedback:wordtype:grade: feedback.ac1.wordtype.ac2  0.32243
  feedback:wordtype:grade: feedback.ac2.wordtype.ac2  0.08870 .
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we only pass one variable at a time to the `split` function, the resulting ANOVA table yields interaction comparisons instead of interaction contrasts.

```r
summary(fit.1aov,  split=list(feedback=list(feedback.ac1=1, feedback.ac2=2)))
```

```
                                        Df Sum Sq Mean Sq F value   Pr(>F)    
feedback                                 2  26.87   13.43   7.112  0.00152 ** 
  feedback: feedback.ac1                 1   9.80    9.80   5.188  0.02571 *  
  feedback: feedback.ac2                 1  17.07   17.07   9.035  0.00364 ** 
wordtype                                 2  64.87   32.43  17.171 7.99e-07 ***
grade                                    1  14.40   14.40   7.624  0.00730 ** 
feedback:wordtype                        4  14.67    3.67   1.941  0.11283    
  feedback:wordtype: feedback.ac1        2   8.03    4.02   2.126  0.12669    
  feedback:wordtype: feedback.ac2        2   6.63    3.32   1.756  0.18007    
feedback:grade                           2   8.60    4.30   2.276  0.10999    
  feedback:grade: feedback.ac1           1   3.20    3.20   1.694  0.19721    
  feedback:grade: feedback.ac2           1   5.40    5.40   2.859  0.09520 .  
wordtype:grade                           2  16.20    8.10   4.288  0.01740 *  
feedback:wordtype:grade                  4  10.00    2.50   1.324  0.26946    
  feedback:wordtype:grade: feedback.ac1  2   2.50    1.25   0.662  0.51905    
  feedback:wordtype:grade: feedback.ac2  2   7.50    3.75   1.985  0.14479    
Residuals                               72 136.00    1.89                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Similarly, with only Wordtype partitioned.......

```r
summary(fit.1aov,  split=list(wordtype=list(wordtype.ac1=1, wordtype.ac2=2)))
```

```
                                        Df Sum Sq Mean Sq F value   Pr(>F)    
feedback                                 2  26.87   13.43   7.112  0.00152 ** 
wordtype                                 2  64.87   32.43  17.171 7.99e-07 ***
  wordtype: wordtype.ac1                 1  14.45   14.45   7.650  0.00721 ** 
  wordtype: wordtype.ac2                 1  50.42   50.42  26.691 2.05e-06 ***
grade                                    1  14.40   14.40   7.624  0.00730 ** 
feedback:wordtype                        4  14.67    3.67   1.941  0.11283    
  feedback:wordtype: wordtype.ac1        2   8.03    4.02   2.126  0.12669    
  feedback:wordtype: wordtype.ac2        2   6.63    3.32   1.756  0.18007    
feedback:grade                           2   8.60    4.30   2.276  0.10999    
wordtype:grade                           2  16.20    8.10   4.288  0.01740 *  
  wordtype:grade: wordtype.ac1           1   4.05    4.05   2.144  0.14747    
  wordtype:grade: wordtype.ac2           1  12.15   12.15   6.432  0.01338 *  
```

```
feedback:wordtype:grade                      4  10.00    2.50   1.324  0.26946
  feedback:wordtype:grade: wordtype.ac1  2   2.50    1.25   0.662  0.51905
  feedback:wordtype:grade: wordtype.ac2  2   7.50    3.75   1.985  0.14479
Residuals                               72 136.00    1.89
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

8.1.1 A WORD OF CAUTION

Please recall from work in the 2-way design and the oneway tutorial that use of this `split` function, while fairly simple, produces Type I SS decompositions when sample sizes are unequal. This may not be desirable, so other approaches are necessary (such as `summary.lm`). Others are outlined below using the **emmeans** or **phia** packages.

When we apply the `summary.lm` function to an `aov` object, the resultant table is a table of the regression coefficients for the model. The t-tests here are tantamount to tests of Type III SS and may be more desirable when sample sizes are unequal. In our example, the squares of these t values equal the F tests from the summary tables above where `split` was use. But that was ONLY because sample sizes were equal in this example.

```
summary.lm(fit.1aov)
```

```
Call:
aov(formula = numrecall ~ feedback * wordtype * grade, data = bg.3way)

Residuals:
   Min     1Q Median     3Q    Max 
  -2.2   -1.0    0.1    1.0    2.2 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                7.60000    0.14487  52.460  < 2e-16 ***
feedback1                 -0.23333    0.10244  -2.278  0.02571 *  
feedback2                 -0.53333    0.17743  -3.006  0.00364 ** 
wordtype1                  0.28333    0.10244   2.766  0.00721 ** 
wordtype2                 -0.91667    0.17743  -5.166 2.05e-06 ***
grade1                    -0.40000    0.14487  -2.761  0.00730 ** 
feedback1:wordtype1        0.12500    0.07244   1.726  0.08870 .  
feedback2:wordtype1        0.14167    0.12546   1.129  0.26258    
feedback1:wordtype2       -0.14167    0.12546  -1.129  0.26258    
feedback2:wordtype2       -0.32500    0.21731  -1.496  0.13913    
```

```
feedback1:grade1          -0.13333    0.10244  -1.302  0.19721
feedback2:grade1          -0.30000    0.17743  -1.691  0.09520 .
wordtype1:grade1           0.15000    0.10244   1.464  0.14747
wordtype2:grade1          -0.45000    0.17743  -2.536  0.01338 *
feedback1:wordtype1:grade1  0.04167    0.07244   0.575  0.56693
feedback2:wordtype1:grade1  0.12500    0.12546   0.996  0.32243
feedback1:wordtype2:grade1 -0.12500    0.12546  -0.996  0.32243
feedback2:wordtype2:grade1 -0.37500    0.21731  -1.726  0.08870 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.374 on 72 degrees of freedom
Multiple R-squared:  0.5336,    Adjusted R-squared:  0.4235
F-statistic: 4.846 on 17 and 72 DF,  p-value: 9.884e-07
```

9 Follow up analyses to evaluate contrasts and simple effects

Analysts have a pair of major choices to make about following up the omnibus effects outlined above. One choice is whether to follow up the omnibus analysis with pairwise types of comparisons using a function such as `pairwise.t.test` (or `pairs` in **emmeans**) along with alpha rate adjustment or with post hoc multiple comparison tests versus a followup method that employs contrasts (perhaps orthogonal). A second choice is whether to obtain simple effects and contrast effects with **ghlt**, **emmeans**, or **phia**. This document demonstrates the contrast approach using the latter two packages (**ghlt** may be added on later). Some multiple comparison approaches are integrated into the demonstration of the use of **emmeans**.

Arbitrarily, the next section demonstrates use of **phia** first and then **emmeans** is used in the following section. "phia" stands for "Post Hoc Interaction Analysis", a label that I find unfortunate. I have argued that a strength of employing contrast analysis is that it should emanate from *a priori* hypotheses which then inform the values of the contrast coefficients. Nonetheless, **phia**, and its `testInteractions` function is a very powerful tool for evaluating simple effects and their contrasts, main effect contrasts, and interaction contrasts. That is the primary usage here.

For uses of both **phia** and **emmeans** (in the later section) it is required to have already performed the omnibus full factorial ANOVA. For these omnibus analyses, it is best to have had in place either "effect/deviation" coding or orthogonal contrast coding. But the use of these two packages can be seen as adding contrast and simple effect analyses on to the omnibus analysis. Full orthogonal sets are not even necessary since the contrasts obtained are not being used to create the full omnibus model which might have already been generated using deviation coding schemes.

Two different motivations exist for using either **phia** or **emmeans** in order to evaluate contrasts and simple effects. The first is that they produce inferential tests based on Type III SS decompositions. The second is that they are the only way I know to obtain simple effects, aside from manual computations.

In the particular data set we are working with here, the only higher order effect that was significant was the Wordtype by Grade interaction. This means that we would typically only follow up that effect by looking at simple main effects of one of those factors at levels of the other, as well as their interaction contrasts. We would also follow up with contrasts on the Feedback factor since its main effect was significant and no interactions with Feedback were significant. Of course this logic is a prime example of traditional thinking about NHST methods and use of the word significance which are both under scrutiny now.

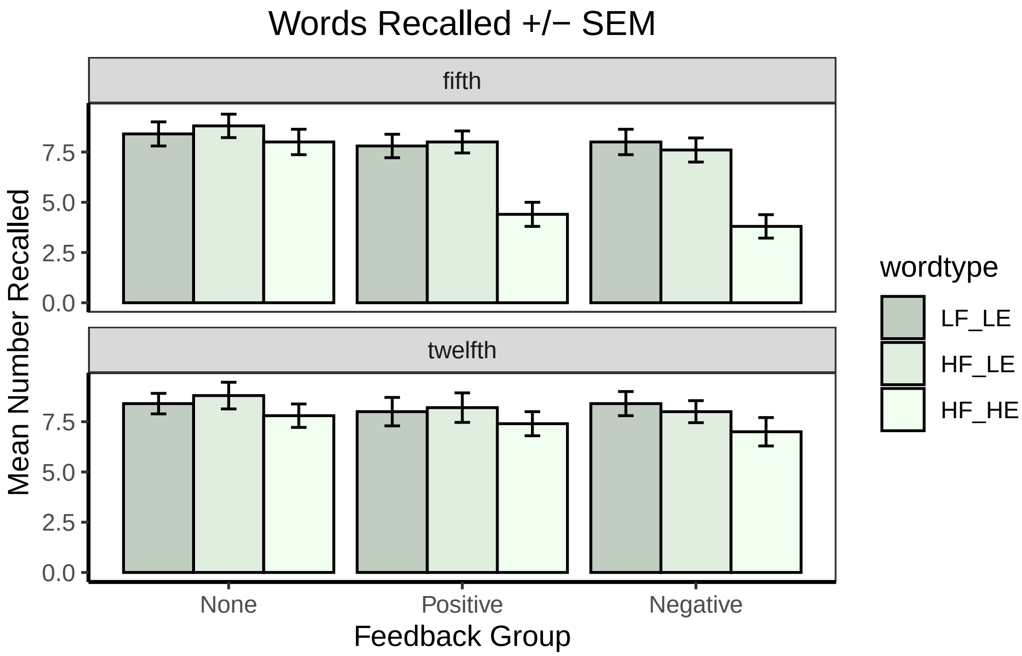
However, the approach taken here is designed to demonstrate how the whole suite of effects can be obtained.

10 Use of the phia package for simple effects and contrasts

Initially this document uses the **phia** package for these evaluations and makes extensive use of the `testInteractions` function from that package. The methods by "holm", and "Hochberg" are possible for p value adjustments with contrasts. **emmeans** is demonstrated in the next section.

It is worth repeating display of the full set of means in a bar graph so that some of these effects can be visualized.

```
# #| code-summary: "Show/Hide Code"
p2
```



10.1 Simple two way interactions

Simple two way interactions can also be obtained (three sets). We don't expect to be interested in any of these since the three way interaction was not significant, but we might have had an *a-priori* hypothesis about one or more of them, so illustration is included for that possibility and for completeness of this template document.

The information in the tables produced by the `testInteractions` function is not organized/labeled in a user-friendly manner. Each of the rows of the tables below represent tests of the simple two-way at that level of the factor in which the effects are examined (e.g.,feedback by grade at levels of grade in the first illustration).

The general strategy in the `testInteractions` function is to use an argument called "fixed" to specify the variable at which effects of other factors are to be examined in simple effects. The "across" argument specifies a variable for which the effect is being requested.

Note that the combined effects of two variables are requested by putting both of them in the "across" argument - the variable @ which these effects are examined is designated by the "fixed" argument.

Reading the table requires some understanding of the idiosyncracies of `testInteractions` formatting. There is one large table (three rows, including a row for residuals) and it may have wrapped into three sections in some Quarto renderings. The columns labeled (for example) feedback1:wordtype1 are indeed, interactions as the colon implies. The numeric appendage (e.g., feedback1) means a contrast. The numeric values (e.g., 6, 3.2, etc) are "estimates" of the underlying interaction contrast. But since we are not examining contrasts here we can ignore these columns. They are used to compile the F test of each of the simple two way interactions.

The F tests are tests of the two different simple two way interactions at the two levels of grade. At least the question of level of grade is clear in the table - rows. Note that the numerator of the two F tests both have 4 df, as they should since the effects are interactions of wordtype and feedback, both 3 level factors. Also note that the df and SS for the Residual term matches that for the omnibus analysis - each of these simple two-way interactions is tested by the omnibus MSwg term as expected for an appropriate analysis rather than fully separating the analysis into two parts.

To be clear, the F value of 3.0882 tests the simple 2 way interaction of feedback and wordtype at fifth graders. And the F value of .1765 tests the simple two way interaction of feedback and wordtype at twelfth graders.

```
# feedback*wordtype at levels of grade
testInteractions(fit.1aov,fixed=c("grade"), across=c("feedback", "wordtype"),adjust="none")
```

```
F Test:
P-value adjustment method: none
          feedback1:wordtype1 feedback2:wordtype1 feedback1:wordtype2
  fifth                     6                 3.2                -3.2
twelfth                     3                 0.2                -0.2
Residuals
          feedback2:wordtype2    SE1     SE2    SE3    SE4 Df  Sum of Sq      F
  fifth                  -2.8  3.688   2.129 2.1292 1.2293  4    23.3333 3.0882
twelfth                   0.2  3.688   2.129 2.1292 1.2293  4     1.3333 0.1765
Residuals                     72.000 136.000
           Pr(>F)
  fifth   0.02098 *
twelfth   0.94983
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking back at the graph, one can see that in fifth graders, the two lower means in the Positive and Negative HF_HE groups is what is driving the simple two-way there. That pattern is not present in the twelfth graders, although the pattern exists, just much smaller in magnitude.

The analyst would likely never do all of the three possible sets of simple two-way interactions. One one should suffice and it would probably be the one where the two primary IVs were interacted. In our example, the first one fits the way the ggplot bar graph is drawn so that is probably what the choice would be. The reader can see the patterns that generate the two simple two-way interactions in the upper and lower facets of that ggplot bar graph. Note that the test of the simple two-way of wordtype and feedback at fifth grade is significant by traditional NHST methodologies (p <.05), but the other simple two-way is not. This distinction cannot be taken as evidence that the three-way interaction is present - in fact that was tested and found N.S. This apparently illogical outcome is not uncommon with NHST in factorial designs.

The other two possible simple two way interactions are shown here for the sake of illustration of `testInteractions` usage.

A second possible set of simple two way interactions is examined next, wordtype by grade at levels of feedback. The three F tests are the tests of the three different simple two way interactions of wordtype and grade at the three levels of feedback (rows). Also note that their df of 2 is what is expected.

```
# wordtype*grade at levels of feedback
testInteractions(fit.1aov,fixed=c("feedback"), across=c("wordtype","grade"),adjust="none")
```

```
F Test:
P-value adjustment method: none
          wordtype1 wordtype2    SE1     SE2 Df Sum of Sq      F  Pr(>F)
    None       -0.2       0.2  2.129   1.229  2    0.0667 0.0176 0.98251
Positive        2.8      -2.8  2.129   1.229  2   13.0667 3.4588 0.03679 *
Negative        2.8      -2.8  2.129   1.229  2   13.0667 3.4588 0.03679 *
Residuals                     72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And now the third set:

```
# feedback*grade at levels of wordtype
testInteractions(fit.1aov,fixed=c("wordtype"), across=c("feedback","grade"),adjust="none")
```

```
F Test:
P-value adjustment method: none
          feedback1 feedback2    SE1     SE2 Df Sum of Sq      F  Pr(>F)
LF_LE          -0.6      -0.2  2.129   1.229  2       0.2 0.0529 0.94847
HF_LE          -0.6      -0.2  2.129   1.229  2       0.2 0.0529 0.94847
HF_HE          -3.6      -3.2  2.129   1.229  2      18.2 4.8176 0.01087 *
Residuals                     72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.2 Simple-simple main effects

These are effects of one factor at combined levels of two other factors.

First examine the so-called simple, simple main effects (three different sets). These effects would be assessed after finding significant simple two way interactions. Notice that the `testInteractions` function accepts an argument for p value adjustment (e.g., bonferroni/holm/fdr), but that I've set it to "none" in these illustrations.

The general strategy in the `testInteractions` function is to use an argument called "fixed" to specify the variable at which effects of other factors are to be examined in simple effects. The "across" argument specifies a variable for which the effect is being requested.

The first set here examines effects of wordtype at combinations of the other two factor levels and is one of the sets of simple simple main effects most visible from the ggplot bar graph above.

Here, and for later tables, "values" of effects are differentiated by the contrasts associated with the factors although those are aggregated to obtain the simple effects - this is confirmed since many of them have multiple df. We are ignoring those "effect" values here since we have not addressed the contrasts in place at the time the `aov` fit was done (default is dummy coding but we changed it to orthogonal contrast coding at the beginning of this section). The exact choice of contrast set doesn't influence the 2 df simple simple main effects here.

Notice that each of these simple simple main effects in this first set have 2 df (three means compared, so 2 df). Later we will break the effects down into contrasts. Compare the significance test results to the ggplot bar graph. The only significant simple simple main effects in this table are the wordtype comparisons in the upper panel (fifth grade) for the negative and positive feedback conditions. This pattern of significance fits what the eye sees in the graph.

```
# 1,  effects of wordtype @ combinations of feedback and grade
testInteractions(fit.1aov,fixed=c("feedback","grade"), across="wordtype",adjust="none")
```

```
F Test:
P-value adjustment method: none
                   wordtype1 wordtype2    SE1     SE2 Df Sum of Sq       F
    None :   fifth       0.0      -0.8  1.506   0.869  2     1.600  0.4235
Positive :   fifth       3.2      -3.6  1.506   0.869  2    40.933 10.8353
Negative :   fifth       4.6      -3.8  1.506   0.869  2    53.733 14.2235
    None : twelfth       0.2      -1.0  1.506   0.869  2     2.533  0.6706
Positive : twelfth       0.4      -0.8  1.506   0.869  2     1.733  0.4588
Negative : twelfth       1.8      -1.0  1.506   0.869  2     5.200  1.3765
Residuals                                72.000 136.000
                      Pr(>F)
```

```
    None :   fifth    0.6564
Positive :   fifth 7.696e-05 ***
Negative :   fifth 6.226e-06 ***
    None : twelfth    0.5146
Positive : twelfth    0.6339
Negative : twelfth    0.2590
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We could also look at other sets of simple simple main effects but would probably only look at one (such as the immediately preceding one), followed up by contrasts (later in this doc).

```
# 2,  effects of grade @ combinations of feedback and wordtype
testInteractions(fit.1aov,fixed=c("feedback","wordtype"), across="grade",adjust="none")
```

```
F Test:
P-value adjustment method: none
                 Value     SE  Df Sum of Sq       F    Pr(>F)
    None : LF_LE   0.0  0.869   1       0.0  0.0000 1.0000000
Positive : LF_LE  -0.2  0.869   1       0.1  0.0529 0.8186747
Negative : LF_LE  -0.4  0.869   1       0.4  0.2118 0.6467745
    None : HF_LE   0.0  0.869   1       0.0  0.0000 1.0000000
Positive : HF_LE  -0.2  0.869   1       0.1  0.0529 0.8186747
Negative : HF_LE  -0.4  0.869   1       0.4  0.2118 0.6467745
    None : HF_HE   0.2  0.869   1       0.1  0.0529 0.8186747
Positive : HF_HE  -3.0  0.869   1      22.5 11.9118 0.0009371 ***
Negative : HF_HE  -3.2  0.869   1      25.6 13.5529 0.0004450 ***
Residuals               72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And the third possible set of simple simple main effects:

```
# 3,  effects of feedback @ combinations of grade and wordtype
testInteractions(fit.1aov,fixed=c("grade","wordtype"), across="feedback",adjust="none")
```

```
F Test:
P-value adjustment method: none
                feedback1 feedback2    SE1     SE2 Df Sum of Sq       F
  fifth : LF_LE      -0.2      -0.6  1.506   0.869  2     0.933  0.2471
```

```
twelfth : LF_LE       0.4      -0.4  1.506   0.869  2     0.533  0.1412
  fifth : HF_LE      -1.6      -0.8  1.506   0.869  2     3.733  0.9882
twelfth : HF_LE      -1.0      -0.6  1.506   0.869  2     1.733  0.4588
  fifth : HF_HE      -4.8      -3.6  1.506   0.869  2    51.600 13.6588
twelfth : HF_HE      -1.2      -0.4  1.506   0.869  2     1.600  0.4235
Residuals                          72.000 136.000
                   Pr(>F)
  fifth : LF_LE    0.7818
twelfth : LF_LE    0.8686
  fifth : HF_LE    0.3772
twelfth : HF_LE    0.6339
  fifth : HF_HE 9.354e-06 ***
twelfth : HF_HE    0.6564
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.3 Simple main effects

In studies where the 3way interaction is not significant, the traditional approach would advise the analyst to progress with examination the omnibus 2way interactions. If any are significant then they would partly be characterized by examining simple main effects (there are six possible sets here, two each for each of the three possible 2way interactions).

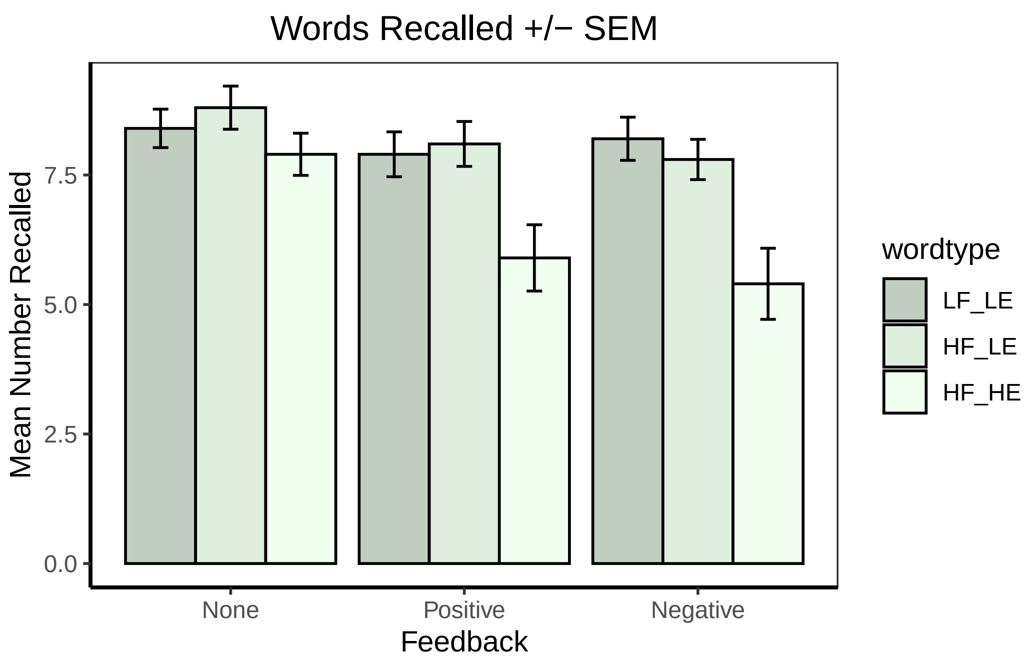
It is not possible to interpret such simple main effects without reference to the tables of collapsed/marginal means such as those produce above following the initial `aov` fit OR by examining the redrawn graph that depicts 2way layouts, collapsed on one of the three factors.

Such a redrawn graph is presented here, to depict the two way layout of wordtype and feedback (a 3x3 structure), since the first of the six sets of SME shown here examines the effect of wordtype at levels of feedback. If this were actual data analysis of a true data set (rather than a textbook/artifical one), then we would only look to follow up the wordtype by grade two way interaction since it was the only one of the three that reached traditional significance levels. But this first graphical illustration permits seeing the more complex 3x3 arrangement, even though the wordtype by feedback interaction was NS. The reader should recognize that the effects of wordtype (seen in the plot below) appear to depend on level of feedback.

From this depiction, we might expect to find an effect of wordtype only in the negative and positive feedback conditions. In reality the 2-way interaction tests whether the wordtype effect is different in the three levels of feedback. It was the general impression that the HF_HE group is lower only in the positive and negative feedback levels. Nontheless while this impression is clear, the difference was not significant with the wordtype by feedback interaction was tested. In this instance the F test result does not match up with the eyeball test - but that is why a test is done. We won't dwell on this seeming paradox since the point of this document is more oriented to providing a template than fully to evaluate a textbook data set (remember, low n means low power).

```
# first, summarize the data set to produce a new data frame that ggplot can use
myData2.wg <- Rmisc::summarySE(data=bg.3way,measurevar="numrecall", groupvars=c("feedback", "
# look at the new data frame that contains the summary statistics
myData2.wg
```

```
#library(ggplot2)
#library(ggthemes)
# Now create the Default bar plot
p3 <- ggplot(myData2.wg, aes(x=feedback, y=numrecall, fill=wordtype)) +
  geom_bar(stat="identity", color="black",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=numrecall-se, ymax=numrecall+se), width=.2,
                position=position_dodge(.9))

p4 <- p3  +labs(title="Words Recalled +/- SEM", x="Feedback", y = "Mean Number Recalled")+
  theme_bw() +
  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'))
p4
```



The following sequence of tests of six sets of simple main effects is presented only to demonstrate how to generate each of them. We would probably only examine number 3 or 4.

In the first analysis we find that in the both the negative feedback condition the effect of feedback significant, as the graph seems to suggest. But note that most of the difference is due to the position of the HF_HE group. We would pursue that pattern with examination of simple main effect contrasts.

```
# 1-2  effects of wordtype at levels of feedback, and then feedback at levels of wordtype
# note these are collapsed on grade
testInteractions(fit.1aov,fixed="feedback", across="wordtype",adjust="none")
```

```
F Test:
P-value adjustment method: none
          wordtype1 wordtype2    SE1     SE2 Df  Sum of Sq       F    Pr(>F)
    None        0.1      -0.9  1.065   0.615  2     4.067  1.0765 0.3462168
Positive        1.8      -2.2  1.065   0.615  2    29.600  7.8353 0.0008341 ***
Negative        3.2      -2.4  1.065   0.615  2    45.867 12.1412 2.859e-05 ***
Residuals                     72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov,fixed="wordtype", across="feedback",adjust="none")
```

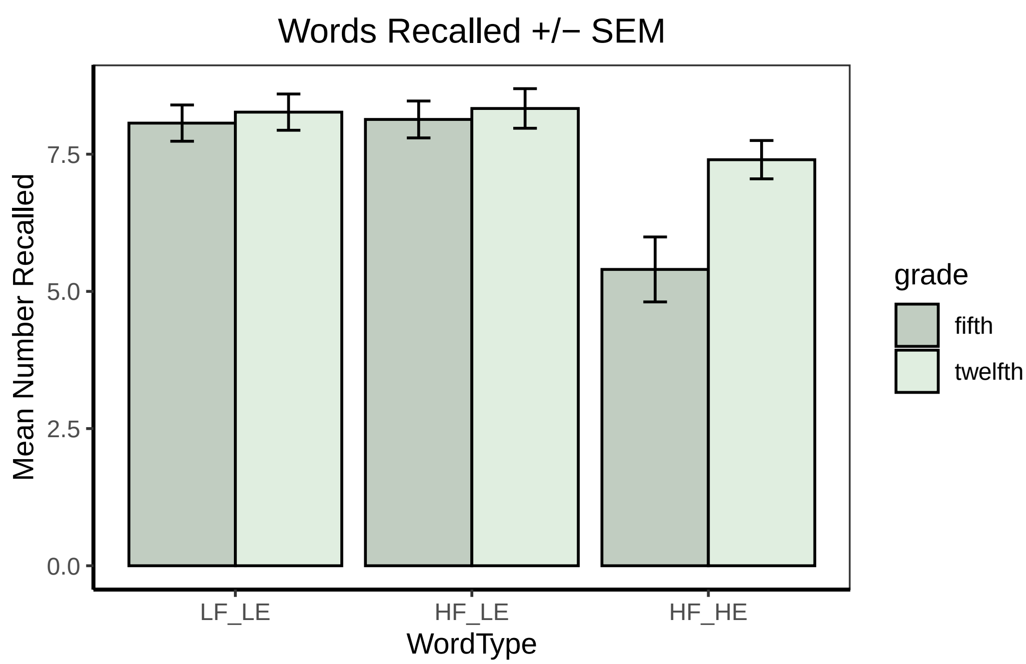
```
F Test:
P-value adjustment method: none
          feedback1 feedback2    SE1     SE2 Df Sum of Sq      F    Pr(>F)
LF_LE           0.1      -0.5  1.065   0.615  2     1.267 0.3353 0.7162383
HF_LE          -1.3      -0.7  1.065   0.615  2     5.267 1.3941 0.2546664
HF_HE          -3.0      -2.0  1.065   0.615  2    35.000 9.2647 0.0002627 ***
Residuals                    72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next we will begin to examine the origins of omnibus wordtype by grade interaction that was significant. In order to interpret the sets of simples possible in this 3x2 grid of means, we should visualize the outcome.

```r
# first, summarize the data set to produce a new data frame that ggplot can use
myData3.wg <- Rmisc::summarySE(data=bg.3way,measurevar="numrecall", groupvars=c("wordtype", "
# look at the new data frame that contains the summary statistics
myData3.wg
```

```r
#library(ggplot2)
#library(ggthemes)
# Now create the Default bar plot
p5 <- ggplot(myData3.wg, aes(x=wordtype, y=numrecall, fill=grade)) +
  geom_bar(stat="identity", color="black",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=numrecall-se, ymax=numrecall+se), width=.2,
                position=position_dodge(.9))

p6 <- p5  +labs(title="Words Recalled +/- SEM", x="WordType", y = "Mean Number Recalled")+
  theme_bw() +
  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'))
p6
```



In the first of the `testInteractions` analyses in this next code chunk (#3), the set of simple main effects of wordtype at levels of grade is produced. From the graph we may expect wordtype to be significant in the 12th graders, but perhaps not in the 5th graders. This is exactly what happened, as seen in the first table. In the second `testInteractions` output here (#4) the effect of grade is significant only in the HF_HE condition, as the visual inspection would have suggested.

```
# 3-4  effects of wordtype at levels of grade, and then grade at levels of feedback
# note these are collapsed on feedback, respectively
testInteractions(fit.1aov,fixed="grade", across="wordtype",adjust="none")
```

```
F Test:
P-value adjustment method: none
          wordtype1 wordtype2    SE1     SE2 Df  Sum of Sq       F    Pr(>F)
  fifth         2.6  -2.73333  0.869   0.502  2     72.933 19.3059 1.937e-07 ***
twelfth         0.8  -0.93333  0.869   0.502  2      8.133  2.1529    0.1236
Residuals                     72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov,fixed="wordtype", across="grade",adjust="none")
```

```
F Test:
P-value adjustment method: none
          Value    SE  Df Sum of Sq       F    Pr(>F)
LF_LE      -0.2  0.502   1       0.3  0.1588 0.6914214
HF_LE      -0.2  0.502   1       0.3  0.1588 0.6914214
HF_HE      -2.0  0.502   1      30.0 15.8824 0.0001597 ***
Residuals       72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The third omnibus 2way can also be followed up with either of its set of simple main effects. This is presented just for the sake of completeness. A redrawn plot of feedback by grade is not provided and the two-way interaction of feedback by grade that would have prompted this analysis was not significant in the omnibus analysis.

```r
# 5-6  effects of feedback at levels of grade, and then grade at levels of feedback
# note these are collapsed on  wordtype
testInteractions(fit.1aov,fixed="grade", across="feedback",adjust="none")
```

```
F Test:
P-value adjustment method: none
          feedback1 feedback2    SE1     SE2 Df  Sum of Sq      F    Pr(>F)
  fifth        -2.2  -1.66667  0.869   0.502  2    32.933 8.7176 0.0004071 ***
twelfth        -0.6  -0.46667  0.869   0.502  2     2.533 0.6706 0.5145724
Residuals                     72.000 136.000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov,fixed="feedback", across="grade",adjust="none")
```

```
F Test:
P-value adjustment method: none
             Value     SE  Df Sum of Sq      F   Pr(>F)
    None   0.06667  0.502   1    0.0333 0.0176 0.894689
Positive  -1.13333  0.502   1    9.6333 5.1000 0.026957 *
Negative  -1.33333  0.502   1   13.3333 7.0588 0.009708 **
Residuals          72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.4 Contrast Analyses in factorial designs using phia

Initially, we will re-create the custom contrasts for the Wordtype and Feedback factors since they each have more than two levels. Note that these are the same contrasts employed above in the omnibus analyses, but phia requires them to be created in a slightly different manner for usage in the `testInteraction` function. The orthogonal sets employed here were arbitrarily chosen. I am not certain why **phia** needs these contrasts established as lists.

```
# first for the feedback factor:
fbc1 <- list(feedback=c(-.5, -.5, 1)) # same as defined above, except using fractions
fbc2 <- list(feedback=c(-1, 1, 0)) # same as defined above, except using fractions
fbc1
```

```
$feedback
[1] -0.5 -0.5  1.0
```

```
fbc2
```

```
$feedback
[1] -1  1  0
```

```
# now for the wordtype factor
wtypec1 <- list(wordtype=c(1, -.5, -.5)) # same as defined above, except using fractions
wtypec2 <- list(wordtype=c(0, -1, 1)) # same as defined above, except using fractions
wtypec1
```

```
$wordtype
[1]  1.0 -0.5 -0.5
```

```
wtypec2
```

```
$wordtype
[1]  0 -1  1
```

10.4.1 Simple Effect Contrasts and Simple Interaction Contrasts

We already obtained 2 and 3-way interaction contrasts above with the summary/split approach and by using `summary.lm` to obtain Type III SS equivalents. Here, we obtain simple interaction contrasts and simple effect contrasts.

In this code, notice that `testInteractions` takes a "custom" argument and that is where we can specify the contrast to be used.

10.4.1.1 Simple 2 way interaction contrasts

First we will partition the simple two-way two way interactions into contrasts. This is shown here largely as a template since the simple 2-way interactions and their contrasts would typically not be evaluated in this data set since the 3-way interaction was not significant by NHST standards. An exception would be if some of the contrasts were *a priori*

Note that the labeling in the table is odd. The two`testInteraction` functions produce the two tables. The first examines the effect involving the first contrast on feedback (called fbc1 in the custom argument), and the second examines the second contrast on feedback (called fbc1). Examining the first of the two tables, each row represents evaluation of a simple two way interaction contrast of the Feedback (first contrast) and Grade at each of the three levels of Wordtype.

The use of the colon symbol does NOT imply an interaction between those two effects as is more typical in R ANOVA/lm notation. In addition, the labeling of the Feedback effect as "feedback1" doesn't refer to the exact contrast that we called fbc1. It is just a generic label for the fact that a contrast on feedback was specified. So in the first table, the first F value of .0794 represents a test of *fbc1 by grade AT LF_LE*. The second, is the same effect, but AT HF_LE and so forth. In the second table, the first row evaluates the *fbc2 by grade interaction AT LF_LE*, with the F value of .0265 EVEN THOUGH the table still uses the "feedback1" label. The confusion can arise because the tables do not employ the fbc1 and fbc2 labels that we created. In each table, there are three tests because there are three levels of Wordtype and the effect is either fbc1 by grade (table 1) or fbc2 by grade (table 2). This pattern plays out in subsequent implementations as well. So, while `testInteractions` is very powerful, the organization of the output and its labels leaves something to be desired.

```
# first, an example of simple 2-way interaction contrasts of feedback with grade at levels o
testInteractions(fit.1aov, fixed="wordtype", custom=fbc1, adjustment="none", across="grade")
```

```
F Test:
P-value adjustment method: none
                   Value     SE  Df Sum of Sq      F Pr(>F)
LF_LE : feedback1   -0.3  1.065   1      0.15 0.0794 0.7789
HF_LE : feedback1   -0.3  1.065   1      0.15 0.0794 0.7789
HF_HE : feedback1   -1.8  1.065   1      5.40 2.8588 0.0952 .
Residuals                72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
testInteractions(fit.1aov, fixed="wordtype", custom=fbc2, adjustment="none", across="grade")
```

```
F Test:
P-value adjustment method: none
                  Value     SE  Df Sum of Sq      F  Pr(>F)
LF_LE : feedback1  -0.2  1.229   1      0.05 0.0265 0.87121
HF_LE : feedback1  -0.2  1.229   1      0.05 0.0265 0.87121
HF_HE : feedback1  -3.2  1.229   1     12.80 6.7765 0.01121 *
Residuals               72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When both factors in a simple 2way interaction have contrasts then the code specification is slightly more complex. With the interaction of wordtype and feedback at levels of grade, grade is specified as "fixed". The "across" argument is not used. Rather, both contrasts are specified simultaneously in the "custom" argument and 'testInteractions understands that the interaction of those two contrasts is to be examined.

Note that the output uses the colon symbol again, but it doesn't mean interaction. The interaction here is between fbc1 and wtypec1 (thus an interaction contrast) at fifth grade.

```
# simple 2-2way interaction contrast of wordtype and grade at levels of feedback
# one requested at a time here, both of the first contrasts (fbc1 and wtypec1) are requested
testInteractions(fit.1aov, fixed="grade", custom=c(fbc1,wtypec1), adjustment="none")
```

```
F Test:
P-value adjustment method: none
                                  Value     SE  Df Sum of Sq      F Pr(>F)
  fifth : feedback1 : wordtype1  1.50  0.922   1      5.00 2.6471 0.1081
twelfth : feedback1 : wordtype1  0.75  0.922   1      1.25 0.6618 0.4186
Residuals                              72.000 136
```

Code for the other six contrasts available in this set is shown here, but results suppressed to save space. If one were to run these additional chunks, the same "feedback1" and "wordtype1" labels would appear, but meaning either fbc1/fbc2 or wtypec1/wtypec2, respectively.

```
testInteractions(fit.1aov,fixed="grade", custom=c(fbc1,wtypec2), adjustment="none")
testInteractions(fit.1aov,fixed="grade", custom=c(fbc2,wtypec1), adjustment="none")
testInteractions(fit.1aov,fixed="grade", custom=c(fbc2,wtypec2), adjustment="none")
```

Sometimes we prefer to examine simple interaction comparisons rather than simple interaction contrasts. That kind of effect is exemplified with this type of code where a contrast is specified for one factor (wordtype here) and the other factor is not broken into contrasts (feedback here, specified in the "across" argument). These will be a 2 df interaction contrasts: either wordtype contrast 1 by feedback at levels of grade or wordtype contrast2 by feedback at levels of grade.

Once again the fifth: and twelfth: notation just specify the levels of grade AT which the simple interaction comparisons are located.

```
# now simple 2-way interaction comparisons of wordtype contrasts and feedback at levels of gr
# note that these effects have 2 df
testInteractions(fit.1aov, fixed="grade", custom=wtypec1, adjustment="none", across="feedbac
```

```
F Test:
P-value adjustment method: none
                    feedback1 feedback2    SE1     SE2 Df Sum of Sq      F
  fifth : wordtype1       3.0       1.6  1.844   1.065  2    9.2667 2.4529
twelfth : wordtype1       1.5       0.1  1.844   1.065  2    1.2667 0.3353
Residuals                               72.000 136.000
                    Pr(>F)
  fifth : wordtype1 0.0932 .
twelfth : wordtype1 0.7162
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
testInteractions(fit.1aov, fixed="grade", custom=wtypec2, adjustment="none", across="feedbac
```

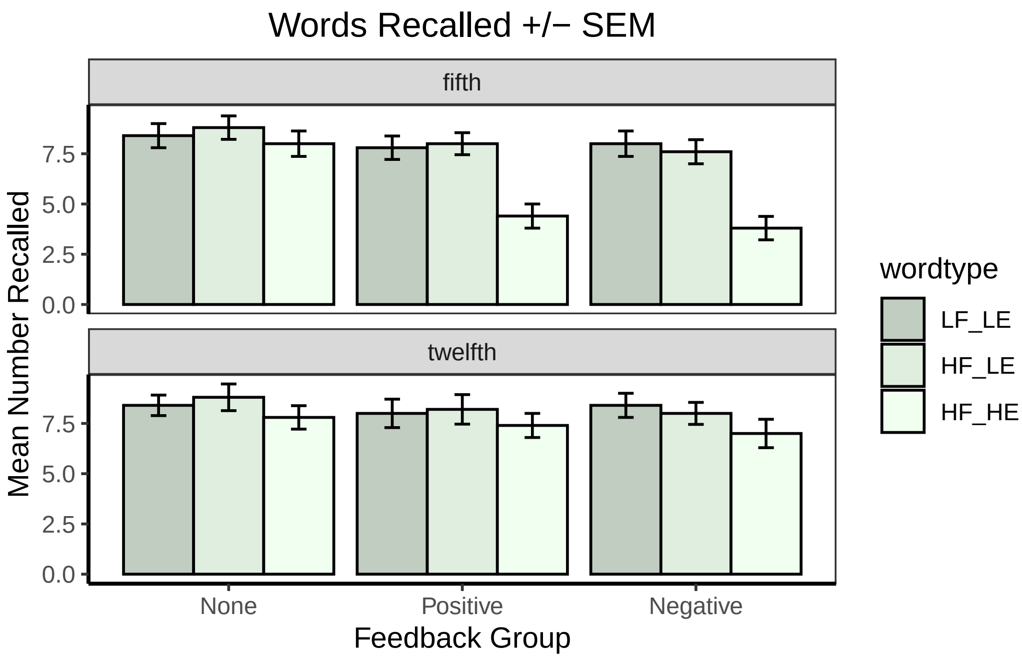
```
F Test:
P-value adjustment method: none
                    feedback1 feedback2    SE1     SE2 Df Sum of Sq      F
  fifth : wordtype1      -3.2      -2.8  2.129   1.229  2   14.0667 3.7235
twelfth : wordtype1      -0.2       0.2  2.129   1.229  2    0.0667 0.0176
Residuals                               72.000 136.000
                     Pr(>F)
  fifth : wordtype1 0.02892 *
twelfth : wordtype1 0.98251
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.4.1.2 Simple and simple-simple main effect contrasts

Contrasts breaking down the simple-simple or simple main effects are also available.

Initially, we can examine simple simple main effect contrasts of feedback at the combined levels of wordtype and grade.

It is helpful to reexamine the plot of the data of all cell means to visualize these simple simple main effects.



As we saw above, care is required in reading the table, which does not label the different contrasts. The first table is for the fbc1 contrast and the second table for the fbc2 contrast, each at the six combined levels of Grade and Wordtype in this first illustration.

Once again, the "feedback1" label in the tables is misleading and not necessary. We have to discern which contrast on feedback is being tested by knowing which code line produced which table.

```
# Now, how about simple SME contrasts for the feedback factor at
# combined levels of grade and wordtype
testInteractions(fit.1aov, fixed=c("wordtype", "grade"), custom=fbc1, adjustment="none")
```

```
F Test:
P-value adjustment method: none
                           Value     SE  Df Sum of Sq       F   Pr(>F)
LF_LE :   fifth : feedback1  -0.1  0.753   1    0.0333  0.0176 0.894689
HF_LE :   fifth : feedback1  -0.8  0.753   1    2.1333  1.1294 0.291452
HF_HE :   fifth : feedback1  -2.4  0.753   1   19.2000 10.1647 0.002119 **
```

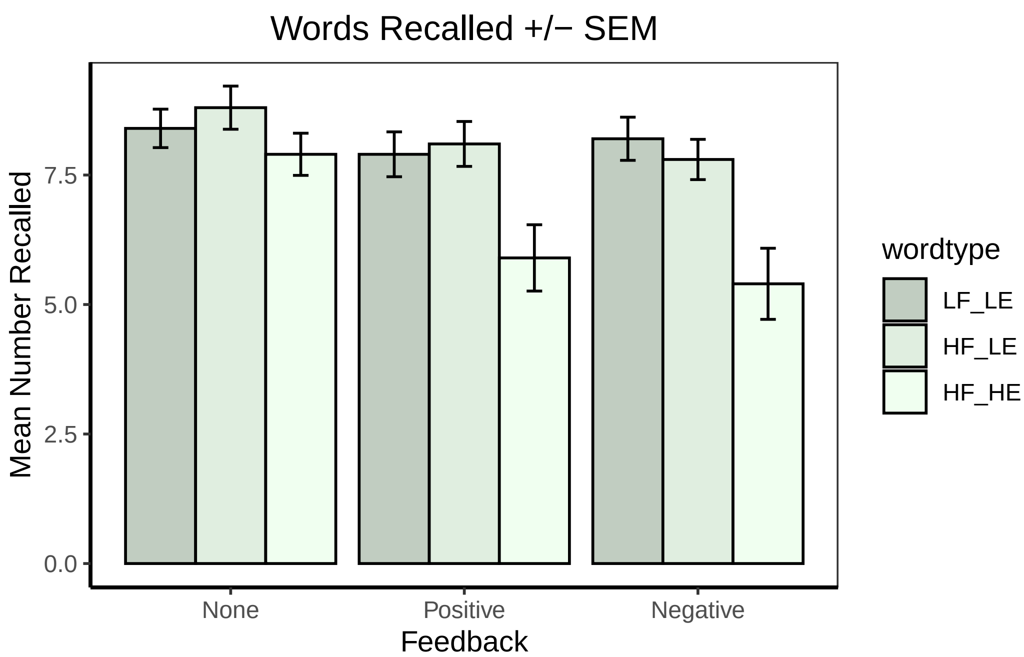
```
LF_LE : twelfth : feedback1   0.2  0.753   1    0.1333  0.0706 0.791242
HF_LE : twelfth : feedback1  -0.5  0.753   1    0.8333  0.4412 0.508677
HF_HE : twelfth : feedback1  -0.6  0.753   1    1.2000  0.6353 0.428041
Residuals                         72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov, fixed=c("wordtype", "grade"), custom=fbc2, adjustment="none")
```

```
F Test:
P-value adjustment method: none
                            Value     SE  Df Sum of Sq       F    Pr(>F)
LF_LE :   fifth : feedback1  -0.6  0.869   1       0.9  0.4765    0.4922
HF_LE :   fifth : feedback1  -0.8  0.869   1       1.6  0.8471    0.3605
HF_HE :   fifth : feedback1  -3.6  0.869   1      32.4 17.1529 9.265e-05 ***
LF_LE : twelfth : feedback1  -0.4  0.869   1       0.4  0.2118    0.6468
HF_LE : twelfth : feedback1  -0.6  0.869   1       0.9  0.4765    0.4922
HF_HE : twelfth : feedback1  -0.4  0.869   1       0.4  0.2118    0.6468
Residuals                         72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we evaluate Simple Main effect contrasts of Feedback at levels of Wordtype, collapsed on Grade.

It would be helpful to repeat the plot of the nine means, collapsed on grade.



These next three code chunks also produce the two contrast tests with and without a "holm" adjustment.

The first code chunk is thus the test of feedback contrast 1 at levels of wordtype, with no p value adjustment. Once again, the "feedback1" label can be ignored. As an interpretation example, consider the third F test with a value of 7.94. This effect is feedback contrast 1 in the HF_HE group (collapsed on grade)

```
# or SME contrasts at levels of wordtype (collapsed on grade)
testInteractions(fit.1aov, fixed=c("wordtype"), custom=fbc1, adjustment="none")
```

```
F Test:
P-value adjustment method: none
                  Value     SE  Df Sum of Sq      F   Pr(>F)
LF_LE : feedback1  0.05  0.532   1    0.0167 0.0088 0.925423
HF_LE : feedback1 -0.65  0.532   1    2.8167 1.4912 0.226018
HF_HE : feedback1 -1.50  0.532   1   15.0000 7.9412 0.006234 **
Residuals               72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The next chunk does the same analysis with a "holm" adjustment.

```r
# note that adjustment for Type I error inflation can be done.  The "adjustment"
# argument uses the methods available in p.adjust
testInteractions(fit.1aov, fixed=c("wordtype"), custom=fbc1, adjustment="holm")
```

```
F Test: 
P-value adjustment method: holm
                  Value     SE  Df Sum of Sq      F Pr(>F)  
LF_LE : feedback1  0.05  0.532   1    0.0167 0.0088 0.9254  
HF_LE : feedback1 -0.65  0.532   1    2.8167 1.4912 0.4520  
HF_HE : feedback1 -1.50  0.532   1   15.0000 7.9412 0.0187 *
Residuals               72.000 136                          
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And the same approach is used for the second feedback contrast.

```r
testInteractions(fit.1aov, fixed=c("wordtype"), custom=fbc2, adjustment="none")
```

```
F Test: 
P-value adjustment method: none
                  Value     SE  Df Sum of Sq       F   Pr(>F)   
LF_LE : feedback1  -0.5  0.615   1      1.25  0.6618 0.418620   
HF_LE : feedback1  -0.7  0.615   1      2.45  1.2971 0.258526   
HF_HE : feedback1  -2.0  0.615   1     20.00 10.5882 0.001735 **
Residuals               72.000 136                              
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov, fixed=c("wordtype"), custom=fbc2, adjustment="holm")
```

```
F Test: 
P-value adjustment method: holm
                  Value     SE  Df Sum of Sq       F   Pr(>F)   
LF_LE : feedback1  -0.5  0.615   1      1.25  0.6618 0.517053   
HF_LE : feedback1  -0.7  0.615   1      2.45  1.2971 0.517053   
HF_HE : feedback1  -2.0  0.615   1     20.00 10.5882 0.005204 **
Residuals               72.000 136                              
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are many other simple main effect contrasts that could be exmained (following up on the six sets of possible simple main effects described above). The important to examine in the context of this data set would be contrasts on wordtype at levels of grade, since the omnibus wordtype by grade interaction was significant. In order to shorten this doc, only one of those SME contrasts is shown here, wordtype contrast2 @ fifth grade.

```
testInteractions(fit.1aov, fixed="grade", custom=fbc2, adjustment="none")
```

```
F Test:
P-value adjustment method: none
                        Value     SE  Df Sum of Sq       F   Pr(>F)
  fifth : feedback1 -1.66667  0.502   1   20.8333 11.0294 0.001411 **
twelfth : feedback1 -0.46667  0.502   1    1.6333  0.8647 0.355532
Residuals                    72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.4.2 3way interaction contrasts with phia

Next, we will examine 3-way interaction contrasts. This shows that **phia** can recreate what we did above with the summary/split approach and with the `summary.lm` function. Those two match because of the equal sample sizes in this example data set. If the data set were unbalanced, the tests here, from `testInteractions` would match the squared t-values from tests obtained in the `summary.lm` output.

Here, two of the three factors have contrasts available and those specifications are put in the "custom" argument. The third factor is only a 2-level factor (already a contrasts), and can just be specified in the "across" argument.

The first of these is thus fbc1 x wtypec1 x grade, etc….

```
testInteractions(fit.1aov, custom=c(fbc1,wtypec1), adjustment="none",across="grade")
```

```
F Test:
P-value adjustment method: none
                       Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1  0.75  1.304   1     0.625 0.3309 0.5669
Residuals                    72.000 136
```

```
testInteractions(fit.1aov, custom=c(fbc1,wtypec2), adjustment="none",across="grade")
```

```
F Test:
P-value adjustment method: none
                      Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1  -1.5  1.506   1     1.875 0.9926 0.3224
Residuals                   72.000 136
```

```
testInteractions(fit.1aov, custom=c(fbc2,wtypec1), adjustment="none",across="grade")
```

```
F Test:
P-value adjustment method: none
                      Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1   1.5  1.506   1     1.875 0.9926 0.3224
Residuals                   72.000 136
```

```
testInteractions(fit.1aov, custom=c(fbc2,wtypec2), adjustment="none",across="grade")
```

```
F Test:
P-value adjustment method: none
                      Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1    -3  1.738   1     5.625 2.9779 0.0887 .
Residuals                   72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.4.3 Two-way interaction contrasts

By leaving the "across=grade" argument out of the above code chunk, we can obtain the two-way interaction contrasts of Feedback an Wordtype (collapsed on Grade).

```
testInteractions(fit.1aov, custom=c(fbc1,wtypec1), adjustment="none")
```

```
F Test:
P-value adjustment method: none
                      Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1 1.125  0.652   1     5.625 2.9779 0.0887 .
Residuals                   72.000 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
testInteractions(fit.1aov, custom=c(fbc1,wtypec2), adjustment="none")
```

```
F Test: 
P-value adjustment method: none
                      Value     SE  Df Sum of Sq     F Pr(>F)
feedback1 : wordtype1 -0.85  0.753   1    2.4083 1.275 0.2626
Residuals                   72.000 136                       
```

```r
testInteractions(fit.1aov, custom=c(fbc2,wtypec1), adjustment="none")
```

```
F Test: 
P-value adjustment method: none
                      Value     SE  Df Sum of Sq     F Pr(>F)
feedback1 : wordtype1  0.85  0.753   1    2.4083 1.275 0.2626
Residuals                   72.000 136                       
```

```r
testInteractions(fit.1aov, custom=c(fbc2,wtypec2), adjustment="none")
```

```
F Test: 
P-value adjustment method: none
                      Value     SE  Df Sum of Sq      F Pr(>F)
feedback1 : wordtype1  -1.3  0.869   1     4.225 2.2368 0.1391
Residuals                   72.000 136                        
```

Finally, in this data set, the most interesting interaction effect to follow up would have been the Wordtype By Grade interaction which was the only significant one. Simple main effects of Wordtype at Levels of Grade (and the contrasts of those SME) were obtained above. Here, the interaction contrasts were obtained to decompose this two way interaction.

Notice that the sum of the SS for the two effects equals the SS for the Wordtype by Grade interaction seen in the initial ANOVA summary table, as it should for an orthogonal partitioning.

In this pair of analyses, only the second of the two contrasts is significant and this should not be surprising. Most of what was varying in the wordtype by grade array of means was the position of the HF_HE group. So, it makes sense that the second contrast which compares HF_LE to HF_HE interacts with grade since the difference is larger in the 12th graders.

```r
testInteractions(fit.1aov, custom=wtypec1, across="grade", adjustment="none")
```

```
F Test:
P-value adjustment method: none
          Value     SE  Df Sum of Sq      F Pr(>F)
wordtype1   0.9  0.615   1      4.05 2.1441 0.1475
Residuals       72.000 136
```

```r
testInteractions(fit.1aov, custom=wtypec2, across="grade", adjustment="none")
```

```
F Test:
P-value adjustment method: none
          Value    SE  Df Sum of Sq      F  Pr(>F)
wordtype1  -1.8  0.71   1     12.15 6.4324 0.01338 *
Residuals       72.00 136
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.4.4 A note on orthogonal contrasts

The reader should understand that use of the `testInteractions` function does not require that contrasts submitted to it be orthogonal. Each contrast was submitted one at a time and not used in calculating any other full model. The orthogonality approach taken here was simply recognition that there are some advantages of thinking about contrast sets as orthogonal - from the *a prior* point of view.

10.5 Conclusion about use of phia and `testInteractions`

The `testInteractions` function is very powerful, permitting all simple and contrast effects to be specified (we did not examine main effect contrasts here, but they can be done as was outlined in the 2 factor ANOVA tutorial document). I am very positive about the relative ease of use of the arguments in `testInteractions`. However, the layout and labeling in the tables produced by the function are difficult. It was only with careful comparison to other known analyses that I was able to be certain what the `testInteraction` function was testing, at times. With more regular use, the style of the table layout becomes easier to understand, but for the novice user the labeling (e.g., :feedback1") can be misleading. If it were not for this output formatting difficulty, I would be strongly recommending use of the **phia** functions for this type of analysis. But **emmeans** may produce output that is more easily interpreted for most users.

Finally, I am unaware of a direct wa of producing effect sizes on effects or contrasts produced by `testInteractions`. However, since `testInteractions` does provide SS for effects, manual computation of eta or partial eta squareds can begin with those SS values.

11 Use of emmeans for contrasts and simple effects

The tools of the **emmeans** package can produce all of the same analyses as those just accomplished above with **phia**. Some types of effects are slightly easier to obtain from **emmeans**, but others are more difficult (e.g., interaction contrasts). Either package provides strong tools for these follow up analyses.

This **emmeans** section does not perform the detailed examination of nearly all possible contrasts as was done in the **phia** section. Instead a few examples are used in order to provide templates and the choice of which effects to examine is partly driven by patterns in the data set. Once again, the goal here is the provision of templates for use in other analyses, not the "best" possible evaluation of this textbook data set.

11.1 Simple two-way interactions.

The logic and introductory wording here matches that in the **phia** section above:

Simple two way interactions are available (three sets). We don't expect to be interested in any of these since the three way interaction was not significant, but we might have had an a-priori hypothesis about one or more of them, so illustration is included for completeness.

- wordtype by grade @ levels of feedback
- wordtype by feedback @ levels of grade
- feedback by grade @ levels of wordtype

Even if there were a significant 3=way interaction, we would probably only have been interested in one of these three sets of simple two-ways. It would probably have been the second one since wordtype and feedback are the two primary manipulated IVs.

Either the `aov` or `anova_car` fit objects will work with **emmeans**. We will use the `anova_car` object here. First, extract the grid of means and other descriptive info, for all of the cells.

```
wgf.emm <- emmeans(fit_base.afex,~wordtype:grade:feedback)
wgf.emm
```

```
 wordtype grade   feedback emmean    SE df lower.CL upper.CL
 LF_LE    fifth   None        8.4 0.615 72     7.17     9.63
 HF_LE    fifth   None        8.8 0.615 72     7.57    10.03
 HF_HE    fifth   None        8.0 0.615 72     6.77     9.23
 LF_LE    twelfth None        8.4 0.615 72     7.17     9.63
 HF_LE    twelfth None        8.8 0.615 72     7.57    10.03
 HF_HE    twelfth None        7.8 0.615 72     6.57     9.03
 LF_LE    fifth   Positive    7.8 0.615 72     6.57     9.03
```

```
 HF_LE    fifth   Positive    8.0 0.615 72     6.77     9.23
 HF_HE    fifth   Positive    4.4 0.615 72     3.17     5.63
 LF_LE    twelfth Positive    8.0 0.615 72     6.77     9.23
 HF_LE    twelfth Positive    8.2 0.615 72     6.97     9.43
 HF_HE    twelfth Positive    7.4 0.615 72     6.17     8.63
 LF_LE    fifth   Negative    8.0 0.615 72     6.77     9.23
 HF_LE    fifth   Negative    7.6 0.615 72     6.37     8.83
 HF_HE    fifth   Negative    3.8 0.615 72     2.57     5.03
 LF_LE    twelfth Negative    8.4 0.615 72     7.17     9.63
 HF_LE    twelfth Negative    8.0 0.615 72     6.77     9.23
 HF_HE    twelfth Negative    7.0 0.615 72     5.77     8.23

Confidence level used: 0.95
```

In order to test the simple two way interactions, a direct and isolated approach is not possible. Instead, we request what amounts to examination of three kinds of effects at the levels of feedback. This code provides the simple main effects of both wordtype and grade as well as their interaction (and here the interaction term uses the colon symbol), all AT levels of feedback, using the omnibus error term. Thus the simple two-way interactions that we set out to obtain are the third effect tested in each of these three sets of tables/analyses. Note that the second and third of them have identical F and p values - this is correct output, but suggests that the data set may have been fabricated for use in the Keppel textbook.

```
# first, examine wordtype*grade at levels of feedback
joint_tests(wgf.emm, by="feedback")
```

```
feedback = None:
 model term     df1 df2 F.ratio p.value
 wordtype         2  72   1.076  0.3462
 grade            1  72   0.018  0.8947
 wordtype:grade   2  72   0.018  0.9825

feedback = Positive:
 model term     df1 df2 F.ratio p.value
 wordtype         2  72   7.835  0.0008
 grade            1  72   5.100  0.0270
 wordtype:grade   2  72   3.459  0.0368

feedback = Negative:
 model term     df1 df2 F.ratio p.value
 wordtype         2  72  12.141  <.0001
 grade            1  72   7.059  0.0097
```

```
 wordtype:grade   2  72   3.459  0.0368
```

Now the other two sets of simple two-ways are obtained.

```
# second wordtype*feedback at levels of grade
(joint_tests(wgf.emm, by="grade"))
```

```
# second grade*feedback at levels of wordtype
joint_tests(wgf.emm, by="wordtype")
```

One thing that I do not like about using **emmeans** functions for these types of followup analyses is that the tabled results do not provide SS and MS for the effects. It is useful to doublecheck software for its accuracy by verifying that things work out as they should. For example the SS for the set of grade*feedback simple two ways at levels of wordtype should sum to the pooling of the omnibus two way of grade by feedback and the three way of grade by feedback by wordtype. We could work backwards from the F values (using the MSerror from above) to find these values, but that added work is not completed here. Initial code to show how to manually find a SS from the F value is here:

```
# e.g. for grade by feedback at HF_HE
# the F value was 4.818
# df for this simple two way is 2
# MSerror of 1.889 from the omnibus analyses above
df_effect <- 2
MSeffect <- 1.889*4.818
SSeffect <- MSeffect*df_effect
SSeffect
```

```
[1] 18.2024
```

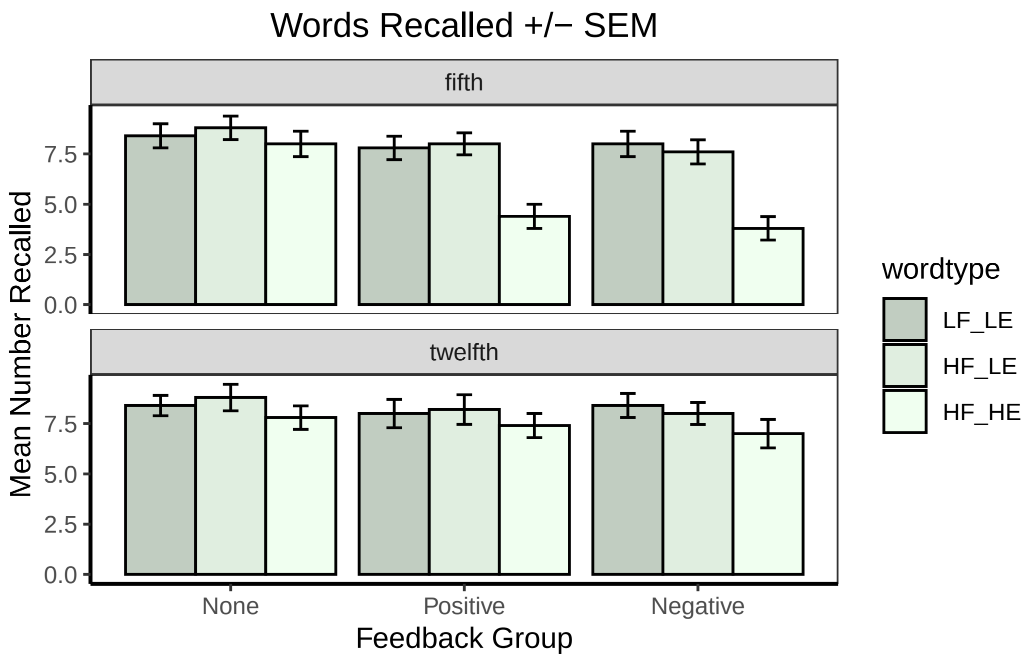
Doing this with the other two would provide the value that when summed should equal the pooled SS described above.

11.2 Simple Simple main effects

Since the 3way interaction was not significant, the analyst would probably not go on to examine simple simple main effects. Nonetheless one set is outlined here for the sake of demonstrating the **emmeans** capability.

Working from the full grid of means from all cells (the wgf.emm grid), we can specify the oneway type of question that simple simple main effects ask. The variable to be examined is wordtype - at the different levels of feedback and grade combined.

A repetition of the cell means graph here can help visualization of where the question is asked. With this plot it is reinforced that the 2df effect of wordtype can be examined in six places (the combinations of feedback and grade)



The briefest way to request the simple simple main effects is to use the `joint_test` function on the 3way grid of means. Note that the focal variable is not directly named in an argument. Rather, it is implied because the wgf.emm object contains means from combinations of all three IVs. When the feedback and grade variables are set as the variables AT which the questions are to be asked (by "by"). that leaves only wordtype as the IV of the effect.

Not surprisingly, the only places that wordtype is a significant effect is in the negative and positive fifth grade conditions.

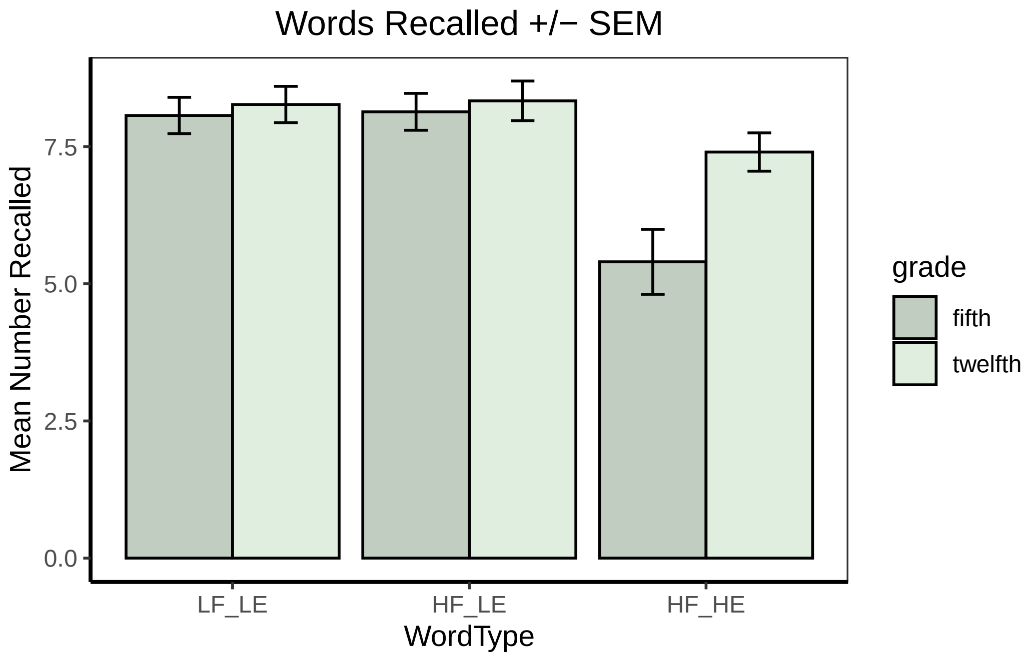
```
joint_tests(wgf.emm, by=c("feedback","grade"))
```

```
feedback = None, grade = fifth:
 model term df1 df2 F.ratio p.value
 wordtype     2  72   0.424  0.6564

feedback = Positive, grade = fifth:
```

```
 model term df1 df2 F.ratio p.value
 wordtype     2  72  10.835  0.0001

feedback = Negative, grade = fifth:
 model term df1 df2 F.ratio p.value
 wordtype     2  72  14.224  <.0001

feedback = None, grade = twelfth:
 model term df1 df2 F.ratio p.value
 wordtype     2  72   0.671  0.5146

feedback = Positive, grade = twelfth:
 model term df1 df2 F.ratio p.value
 wordtype     2  72   0.459  0.6339

feedback = Negative, grade = twelfth:
 model term df1 df2 F.ratio p.value
 wordtype     2  72   1.376  0.2590
```

11.3 Simple Main Effects.

In this particular data set the 3way interaction was not significant and the only two way interaction that was significant was the wordtype by grade effect. The graph of these collapsed/marginal means is redrawn here to provide a reference point.



Now extract descriptive stats into an `emmeans` grid.

```
w.g.emm <- emmeans(fit_base.afex, "wordtype", by="grade")
```

```
NOTE: Results may be misleading due to involvement in interactions
```

```
w.g.emm
```

```
grade = fifth:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE      8.07 0.355 72     7.36     8.77
 HF_LE      8.13 0.355 72     7.43     8.84
 HF_HE      5.40 0.355 72     4.69     6.11

grade = twelfth:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE      8.27 0.355 72     7.56     8.97
 HF_LE      8.33 0.355 72     7.63     9.04
 HF_HE      7.40 0.355 72     6.69     8.11

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

Now perform the tests of wordtype at levels of grade:

```
test(contrast(w.g.emm),joint=TRUE, adjust=none)
```

```
 grade   df1 df2 F.ratio p.value note
 fifth     2  72  19.306  <.0001  d
 twelfth   2  72   2.153  0.1236  d

d: df1 reduced due to linear dependence
```

It is not clear what the warning about df and linear dependence is referring to here. Each of these simple main effects has the 2 numerator df expected for a comparison involving three means. I suspect that I may have used the "joint" argument in a way that is unexpected by the `contrast` function, even though it has produced correct results.

11.4 Contrasts with emmeans

There are five places where contrasts will be demonstrated here. These are on the 3-way and one 2-way interaction, on the set of simple simple main effects examined above, the simple main effects, and the main effects. We will not do all possible ones but be guided (partly) by the outcome of the omnibus tests.

It is easier to begin with the lower order effects.

11.4.1 Main Effect contrasts and pairwise main effect comparisons

In this data set, feedback did not interact with other factors, so we could examine contrasts on that main effect.

```
f.emm <- emmeans(fit_base.afex, "feedback")
```

```
NOTE: Results may be misleading due to involvement in interactions
```

```
f.emm
```

```
 feedback emmean    SE df lower.CL upper.CL
 None       8.37 0.251 72     7.87     8.87
 Positive   7.30 0.251 72     6.80     7.80
 Negative   7.13 0.251 72     6.63     7.63
```

```
Results are averaged over the levels of: wordtype, grade
Confidence level used: 0.95
```

The orthogonal contrast set requested here for this factor is different than the one employed earlier in this document. Here, the Helmert set is specified since the levels are a control (first level) and two treatment conditions. The change was also made just to remind the user that *a priori* choices should drive contrast choice. In examining the means from the grid just produced, the expectation would be that the first of these two contrasts would account for the bulk of the feedback main effect variation, and it does. This change is also why the tests of these two contrasts do not match what was produced with the `summary.lm` analysis on the omnibus model above.

```
lincombs_f <- contrast(f.emm,
                       list(ac1=c(2, -1, -1),
                            ac2=c(0, 1,-1)))
test(lincombs_f, adjust="none")
```

```
 contrast estimate    SE df t.ratio p.value
 ac1         2.300 0.615 72   3.742  0.0004
 ac2         0.167 0.355 72   0.470  0.6400

Results are averaged over the levels of: wordtype, grade
```

It is worth noting at this point that there is not a direct way of obtaining effect sizes with these types of `emmeans` analyses. In addition, since the output simply does a t-test, it is not possible to use a presented table of SS to calculate eta or partial eta squareds manually. This is a downside of using **emmeans** (or **phia**). I have seen ways of obtaining effect sizes for pairwise comparisons using `emmeans`, but not for contrasts - still searching. The `testInteractions` function from **phia** does provide SS for contrast effects as seen above, so manual computation of effect sizes can be accomplished with those quantities.

It is also worth remembering that the **emmeans** approach can also produce adjusted p values. The main effect contrast test is repeated here with a request for bonferroni-sidak correction of the p values (others are possible.

```
test(lincombs_f, adjust="sidak")
```

```
 contrast estimate    SE df t.ratio p.value
 ac1         2.300 0.615 72   3.742  0.0007
 ac2         0.167 0.355 72   0.470  0.8704

Results are averaged over the levels of: wordtype, grade
P value adjustment: sidak method for 2 tests
```

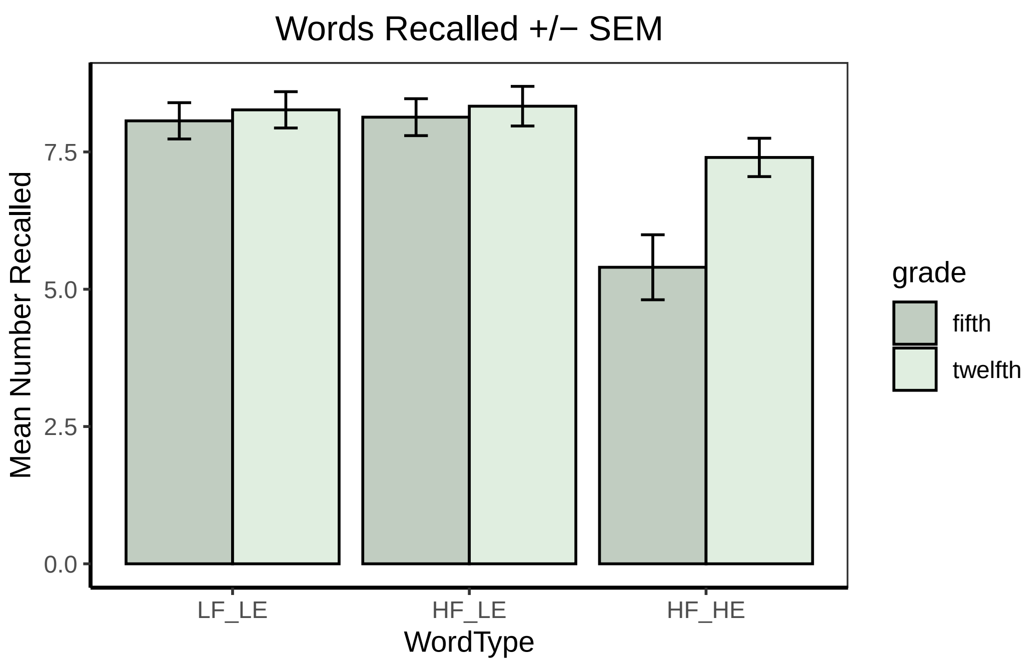
Pairwise comparisons can also be performed, with Tukey-adjusted p values. Note that since the error df is specified as 72, the pooled within-cell error term is used, appropriately if there is homogeneity of variance.

```
pairs(f.emm, adjust="tukey")
```

```
 contrast            estimate    SE df t.ratio p.value
 None - Positive        1.067 0.355 72   3.006  0.0101
 None - Negative        1.233 0.355 72   3.476  0.0025
 Positive - Negative    0.167 0.355 72   0.470  0.8857

Results are averaged over the levels of: wordtype, grade
P value adjustment: tukey method for comparing a family of 3 estimates
```

11.4.2 Simple Main Effect Contrasts

This section focuses on simple main effect following up on the wordtype by grade interaction since that was the only two-way interaction that was significant. It can be visualized again with this graph.



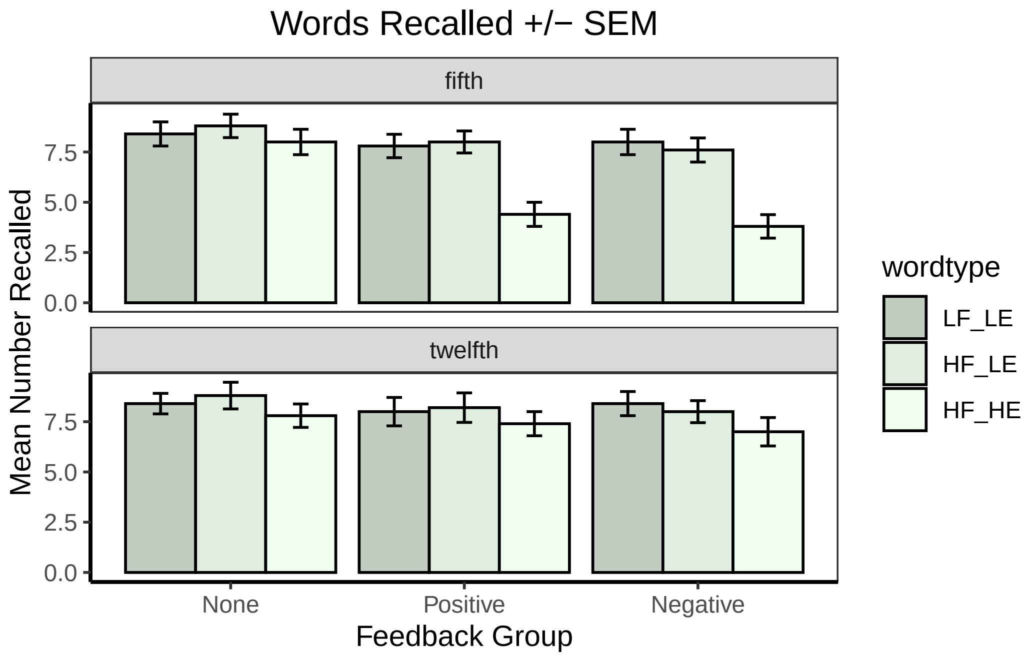
The contrast set on the wordtype variable is also the helmert set. Based on the visual patterns of the means we might expect that both contrasts on wordtype would be significant in fifth graders and neither in 12th graders. This is precisely what the outcome is.

```r
lincombs_w <- contrast(w.g.emm,
                       list(ac1=c(2, -1, -1),
                            ac2=c(0, 1,-1)))
test(lincombs_w, adjust="none")
```

```
grade = fifth:
 contrast estimate    SE df t.ratio p.value
 ac1         2.600 0.869 72   2.991  0.0038
 ac2         2.733 0.502 72   5.447  <.0001

grade = twelfth:
 contrast estimate    SE df t.ratio p.value
 ac1         0.800 0.869 72   0.920  0.3605
 ac2         0.933 0.502 72   1.860  0.0670

Results are averaged over the levels of: feedback
```

11.4.3 Simple Simple Main effect contrasts

Since we examined the Simple Simple main effect of wordtype at combined levels of feedback and grade above, we will now decompose those effects into their simple simple main effect contrasts. It is worth seeing the graph of the cell means again in order to visualize where those simple simple main effects and their contrasts originate.



The contrast set will match what was used for wordtype earlier in the document, the Helmert set.

```
w.gf.emm <- emmeans(fit_base.afex, "wordtype", by=c("grade","feedback"))
w.gf.emm
```

```
grade = fifth, feedback = None:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       8.4 0.615 72     7.17     9.63
 HF_LE       8.8 0.615 72     7.57    10.03
 HF_HE       8.0 0.615 72     6.77     9.23

grade = twelfth, feedback = None:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       8.4 0.615 72     7.17     9.63
 HF_LE       8.8 0.615 72     7.57    10.03
 HF_HE       7.8 0.615 72     6.57     9.03

grade = fifth, feedback = Positive:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       7.8 0.615 72     6.57     9.03
 HF_LE       8.0 0.615 72     6.77     9.23
 HF_HE       4.4 0.615 72     3.17     5.63
```

```
grade = twelfth, feedback = Positive:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       8.0 0.615 72     6.77     9.23
 HF_LE       8.2 0.615 72     6.97     9.43
 HF_HE       7.4 0.615 72     6.17     8.63

grade = fifth, feedback = Negative:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       8.0 0.615 72     6.77     9.23
 HF_LE       7.6 0.615 72     6.37     8.83
 HF_HE       3.8 0.615 72     2.57     5.03

grade = twelfth, feedback = Negative:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE       8.4 0.615 72     7.17     9.63
 HF_LE       8.0 0.615 72     6.77     9.23
 HF_HE       7.0 0.615 72     5.77     8.23

Confidence level used: 0.95
```

```
lincombs_w.gf <- contrast(w.gf.emm,
                         list(ac1=c(2, -1, -1),
                              ac2=c(0, 1,-1)))
test(lincombs_w.gf, adjust="none")
```

```
grade = fifth, feedback = None:
 contrast estimate    SE df t.ratio p.value
 ac1           0.0 1.510 72   0.000  1.0000
 ac2           0.8 0.869 72   0.920  0.3605

grade = twelfth, feedback = None:
 contrast estimate    SE df t.ratio p.value
 ac1           0.2 1.510 72   0.133  0.8947
 ac2           1.0 0.869 72   1.150  0.2538

grade = fifth, feedback = Positive:
 contrast estimate    SE df t.ratio p.value
 ac1           3.2 1.510 72   2.125  0.0370
 ac2           3.6 0.869 72   4.142  0.0001

grade = twelfth, feedback = Positive:
```

```
 contrast estimate    SE df t.ratio p.value
 ac1           0.4 1.510 72   0.266  0.7912
 ac2           0.8 0.869 72   0.920  0.3605

grade = fifth, feedback = Negative:
 contrast estimate    SE df t.ratio p.value
 ac1           4.6 1.510 72   3.055  0.0032
 ac2           3.8 0.869 72   4.372  <.0001

grade = twelfth, feedback = Negative:
 contrast estimate    SE df t.ratio p.value
 ac1           1.8 1.510 72   1.196  0.2358
 ac2           1.0 0.869 72   1.150  0.2538
```

11.4.4 Two way interaction contrasts

Once again the emphasis is on the wordtype by grade interaction since that was the only omnibus 2way interaction that was significant.

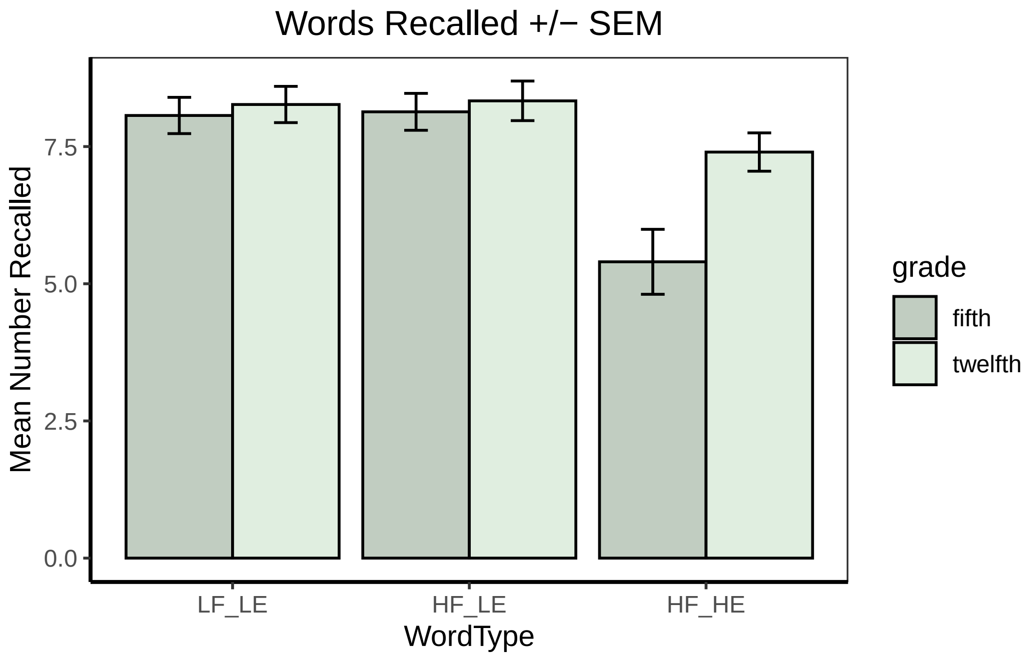
First we set up the 3x2 grid that dimensions wordtype and grade, collapsed on feedback.

```r
wg.emm <- emmeans(fit_base.afex, c("wordtype","grade"))
```

```
NOTE: Results may be misleading due to involvement in interactions
```

```r
wg.emm
```

```
 wordtype grade   emmean    SE df lower.CL upper.CL
 LF_LE    fifth     8.07 0.355 72     7.36     8.77
 HF_LE    fifth     8.13 0.355 72     7.43     8.84
 HF_HE    fifth     5.40 0.355 72     4.69     6.11
 LF_LE    twelfth   8.27 0.355 72     7.56     8.97
 HF_LE    twelfth   8.33 0.355 72     7.63     9.04
 HF_HE    twelfth   7.40 0.355 72     6.69     8.11

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



Using **emmeans** for interaction contrasts is a bit more complex. It requires passing a whole contrast vector rather than just the main effect contrasts - in other words, **emmeans** cannot combine single IV contrasts to obtain the coefficients for an interaction contrast. We have to do it first. And, the order has to match the order of the cells in the grid.

One approach is to take the contrast of interest for wordtype (this initial example is with the first of the two contrasts of interest in the Helmert set) and using matrix operations multiply it by the single contrast for grade. This produces a list that has to be "unlisted" to create a string of six coefficients in order for the six cells of the 3x2.

These operations produce ic1 and ic2 which are the strings of coefficients for the two interaction contrasts for the wordtype by grade 2way interaction.

```
# first, make sure that the wordtype coefficients are helmert
# follows from changing to that specification in a section above
contrasts(bg.3way$wordtype)
```

```
      [,1] [,2]
LF_LE    2    0
HF_LE   -1   -1
HF_HE   -1    1
```

```
wc <- contrasts(bg.3way$wordtype)
wc
```

```
      [,1] [,2]
LF_LE    2    0
HF_LE   -1   -1
HF_HE   -1    1
```

```
# now make sure the contrast for grade is 1, -1
contrasts(bg.3way$grade) <- contr.sum
gc <- contrasts(bg.3way$grade)
gc
```

```
        [,1]
fifth      1
twelfth   -1
```

```
# now create the 6 coefficient interaction contrast vectors
# (two of them)
ic1 <- wc[,1]%*%t(gc)
ic1 <- matrix(unlist(ic1), nrow=1)
ic2 <- wc[,2]%*%t(gc)
ic2 <- matrix(unlist(ic2), nrow=1)
ic1
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2   -1   -1   -2    1    1
```

```
ic2
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0   -1    1    0    1   -1
```

Now that the contrast vectors are known, they can be passed to the `test` function.

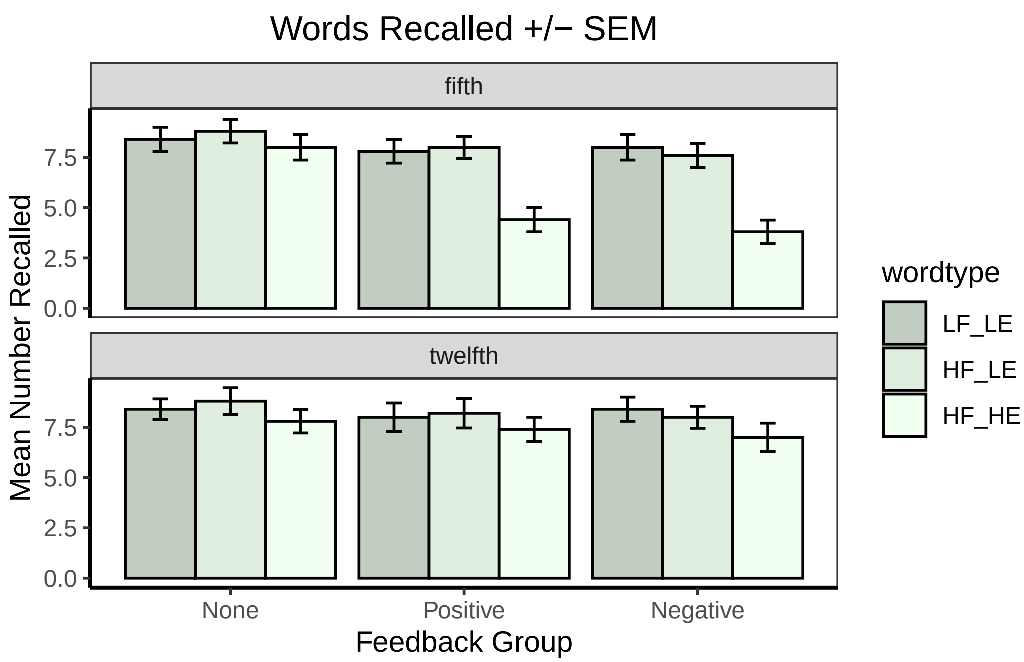
If we look at the graph, we can see that the HF_LE group differs from HF_HE in fifth graders more than it does in 12th graders. This is exactly what the second interaction contrast tests and it is significant by traditional NHST criteria.

```
lincombs.intwg <- contrast(wg.emm,
                        list(intc1=c(2,-1,-1,-2,1,1),
                             intc2=c(0,1,-1,0,-1,1))
                        )
test(lincombs.intwg, adjust="none")
```

```
 contrast estimate   SE df t.ratio p.value
 intc1         1.8 1.23 72   1.464  0.1475
 intc2         1.8 0.71 72   2.536  0.0134

Results are averaged over the levels of: feedback
```

11.4.5 Simple two way interaction contrasts

The wordtype by feedback interaction at levels of grade is a good illustration for simple two-way interaction contrasts because of the way that the base graph was set up.




```
wf.g.emm <- emmeans(fit_base.afex,~wordtype:feedback, by="grade")
wf.g.emm
```

```
grade = fifth:
```

```
 wordtype feedback emmean    SE df lower.CL upper.CL
 LF_LE    None        8.4 0.615 72     7.17     9.63
 HF_LE    None        8.8 0.615 72     7.57    10.03
 HF_HE    None        8.0 0.615 72     6.77     9.23
 LF_LE    Positive    7.8 0.615 72     6.57     9.03
 HF_LE    Positive    8.0 0.615 72     6.77     9.23
 HF_HE    Positive    4.4 0.615 72     3.17     5.63
 LF_LE    Negative    8.0 0.615 72     6.77     9.23
 HF_LE    Negative    7.6 0.615 72     6.37     8.83
 HF_HE    Negative    3.8 0.615 72     2.57     5.03

grade = twelfth:
 wordtype feedback emmean    SE df lower.CL upper.CL
 LF_LE    None        8.4 0.615 72     7.17     9.63
 HF_LE    None        8.8 0.615 72     7.57    10.03
 HF_HE    None        7.8 0.615 72     6.57     9.03
 LF_LE    Positive    8.0 0.615 72     6.77     9.23
 HF_LE    Positive    8.2 0.615 72     6.97     9.43
 HF_HE    Positive    7.4 0.615 72     6.17     8.63
 LF_LE    Negative    8.4 0.615 72     7.17     9.63
 HF_LE    Negative    8.0 0.615 72     6.77     9.23
 HF_HE    Negative    7.0 0.615 72     5.77     8.23

Confidence level used: 0.95
```

For this analysis we change the contrasts to better fit the pattern seen in the graph - this is blatantly post hoc, but we can "see" the possibility of a 2way interaction in grade 5 but not grade 12 data. And most of that is the way that the third wordtype level (HF_HE) differs from the other two and how that, in turn, depends on whether one is examining feedback condition 1 (none) vs. 2 and 3 (positive and negative). This sets up one primary interaction contrast that will be pursued as a template for how to do any/all simple interaction contrasts.

A guiding principle in construction of these contrasts is that the interaction contrast vector has to match the pattern of the order of the 9 cells in the grid of means shown just above.

```r
# change to reverse helmert
reverse <- matrix(c(-1,-1,2,1,-1,0),ncol=2)
reverse
```

```
     [,1] [,2]
[1,]   -1    1
[2,]   -1   -1
[3,]    2    0
```

```
contrasts(bg.3way$wordtype) <- reverse
wc <- contrasts(bg.3way$wordtype)
wc
```

```
      [,1] [,2]
LF_LE   -1    1
HF_LE   -1   -1
HF_HE    2    0
```

```
# set feedback contrasts to helmert
helmert <- matrix(c(2,-1,-1,0,1,-1),ncol=2)
contrasts(bg.3way$feedback) <- helmert
fb <- contrasts(bg.3way$feedback)
fb
```

```
         [,1] [,2]
None        2    0
Positive   -1    1
Negative   -1   -1
```

```
# now create the set of 9-coefficient # interaction contrast vectors
# (four of them)
wc1byfb1 <- wc[,1]%*%t(fb[,1])
wc1byfb1 <- matrix(unlist(wc1byfb1), nrow=1)
wc1byfb2 <- wc[,1]%*%t(fb[,2])
wc1byfb2 <- matrix(unlist(wc1byfb2), nrow=1)
wc2byfb1 <- wc[,2]%*%t(fb[,2])
wc2byfb1 <- matrix(unlist(wc2byfb1), nrow=1)
wc2byfb2 <- wc[,2]%*%t(fb[,2])
wc2byfb2 <- matrix(unlist(wc2byfb2), nrow=1)
```

For demonstration, we will focus only on the first of the four interaction contrasts - since that is the post hoc impression referred to above.

```
wc1byfb1
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]   -2   -2    4    1    1   -2    1    1   -2
```

Wordtype_c1 by Feedback_c1 Coefficients
wordtype_columns

feedback_rows	LF_LE	HF_LE	HF_HE
none	-2	1	1
positive	-2	1	1
negative	4	-2	-2

Again, note that the order of the coefficients is important - it matches the cell order from the grid of means shown above. A block diagram/table can help here.

In R, there should be a way to pass those nine coefficients in the wc1byfb1 vector to the `emmeans::contrast` function but there is a gap in my R understanding so I've just typed them in manually.

The analysis shows what was expected. The simple interaction of the first contrast of wordtype and the first contrast of feedback produced a significant outcome in fifth graders but not in 12th grades, as the graph implies, visually. It is worth recalling here that even though the pattern appears to differ in 5th and 12th graders, the three way interaction was not significant. We can't simple use the presence and absence of significance in simple effects to imply the presence of a higher order effect - it needs to be tested. On the other hand, if these simple interaction contrasts were *a priori* hypotheses, then there is a rationale for examining them without the presence of an omnibus 3way interaction.

```
lincombs.intwf.g <- contrast(wf.g.emm,
                         list(intcontr1=
                         c(-2,-2,4,1,1,-2,1,1,-2)
                         ))
test(lincombs.intwf.g, adjust="none")
```

```
grade = fifth:
 contrast  estimate   SE df t.ratio p.value
 intcontr1     12.6 3.69 72   3.417  0.0010

grade = twelfth:
 contrast  estimate   SE df t.ratio p.value
 intcontr1      0.6 3.69 72   0.163  0.8712
```

11.4.6 Three way interaction contrasts

The simplest way of obtaining the 3-way interaction contrasts is to return to the use of `summary.lm` on the basic `aov` fit object as outlined in section 5.1.1 above. **emmeans** can also produce the three-way interaction contrasts, but with much labor. (see later section)

For the data set under analysis here, the 3-way interaction was NS, so we would probably not undertake to evaluate these interaction contrasts unless there were *a priori* reasons to do so. However, they are shown here anyway (next section) in order to provide a template for situations where their evaluation would be important.

11.4.6.1 Using `summary.lm`

Here is the repetition of how it was done in the sections on 2-way interaction contrasts and simple 2-way interaction contrasts above (with `emmeans`). First, change the contrast set to the desired ones. The sets for wordtype and feedback are the same ones used in the previous `emmans` sections where the two way interaction contrasts and the simple two way interaction of wordtype and feedback were partitioned into contrasts. Note that these are different contrast sets than those used in the earlier section so the results will not match. I changed them just to provide another and different opportunity to match what the analysis says to what the eye sees on the plot.

```
contrasts.wordtype <- matrix(c(-1,-1,2,1,-1,0),ncol=2)
contrasts(bg.3way$wordtype) <- contrasts.wordtype
contrasts(bg.3way$wordtype)
```

```
      [,1] [,2]
LF_LE   -1    1
HF_LE   -1   -1
HF_HE    2    0
```

```
contrasts.feedback <- matrix(c(2,-1,-1,0,1,-1),ncol=2)
contrasts(bg.3way$feedback) <- contrasts.feedback
contrasts(bg.3way$feedback)
```

```
         [,1] [,2]
None        2    0
Positive   -1    1
Negative   -1   -1
```

The grade factor only has two levels so in a sense, it is already a contrast, and recall that we changed it earlier to a 1, -1 vector.

Here, we refit the basic `aov` model, using the factors with these redefined orthogonal contrast sets just to refresh memory and have it on hand in this section. (Same results as appeared earlier when the `aov` fit was produced "fit.1aov")

```r
fit.2aov <- aov(numrecall~wordtype*feedback*grade,data=bg.3way)
car::Anova(fit.2aov, type=3)
```

```
Anova Table (Type III tests)

Response: numrecall
                        Sum Sq Df   F value    Pr(>F)    
(Intercept)             5198.4  1 2752.0941 < 2.2e-16 ***
wordtype                  64.9  2   17.1706 7.993e-07 ***
feedback                  26.9  2    7.1118  0.001519 ** 
grade                     14.4  1    7.6235  0.007304 ** 
wordtype:feedback         14.7  4    1.9412  0.112829    
wordtype:grade            16.2  2    4.2882  0.017397 *  
feedback:grade             8.6  2    2.2765  0.109987    
wordtype:feedback:grade   10.0  4    1.3235  0.269461    
Residuals                136.0 72                        
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Recall that if we use `summary.lm` tests of the regression coefficients are provided and these are the main effect, two way interaction and 3 way interaction contrast vectors - accomplished with t-tests. They are type 3 SS calculations, so these interaction contrasts suit our purposes.

There are four interaction contrasts. We might label them:

1. Wc1 by FBc1 by grade
2. Wc2 by FBc1 by grade
3. Wc1 by FBc2 by grade
4. Wc2 by FBc2 by grade

And they are in that order in the output.

Notice that only the first is significant. In fact the t values for the 2nd, 3rd, and fourth are all zero. This means that all of the 3way interaction is associated with that first contrast. This never happens, so we can assume that the data set is a fabricated textbook data set.

Nonetheless, this provides tests of the 3 way interaction contrasts. Unfortunately, it does not provide SS, only test statistics.

```r
summary.lm(fit.2aov)
```

```
Call:
aov(formula = numrecall ~ wordtype * feedback * grade, data = bg.3way)

Residuals:
   Min     1Q Median     3Q    Max 
  -2.2   -1.0    0.1    1.0    2.2 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 7.600e+00  1.449e-01  52.460  < 2e-16 ***
wordtype1                  -6.000e-01  1.024e-01  -5.857  1.3e-07 ***
wordtype2                  -3.333e-02  1.774e-01  -0.188 0.851509    
feedback1                   3.833e-01  1.024e-01   3.742 0.000364 ***
feedback2                   8.333e-02  1.774e-01   0.470 0.640011    
grade1                     -4.000e-01  1.449e-01  -2.761 0.007304 ** 
wordtype1:feedback1         1.833e-01  7.244e-02   2.531 0.013560 *  
wordtype2:feedback1        -8.333e-02  1.255e-01  -0.664 0.508677    
wordtype1:feedback2         8.333e-02  1.255e-01   0.664 0.508677    
wordtype2:feedback2        -1.500e-01  2.173e-01  -0.690 0.492245    
wordtype1:grade1           -3.000e-01  1.024e-01  -2.929 0.004557 ** 
wordtype2:grade1           -6.975e-16  1.774e-01   0.000 1.000000    
feedback1:grade1            2.167e-01  1.024e-01   2.115 0.037885 *  
feedback2:grade1            5.000e-02  1.774e-01   0.282 0.778905    
wordtype1:feedback1:grade1  1.667e-01  7.244e-02   2.301 0.024296 *  
wordtype2:feedback1:grade1 -8.311e-16  1.255e-01   0.000 1.000000    
wordtype1:feedback2:grade1 -2.027e-17  1.255e-01   0.000 1.000000    
wordtype2:feedback2:grade1  9.435e-17  2.173e-01   0.000 1.000000    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 1.374 on 72 degrees of freedom
Multiple R-squared:  0.5336,    Adjusted R-squared:  0.4235 
F-statistic: 4.846 on 17 and 72 DF,  p-value: 9.884e-07
```

By using `summary.lm`, R has combined the main effect coding vectors in the appropriate ways to create the full rank X matrix that contains all 2 and 3 way interaction contrasts as well as main effect contrasts. This just comes out at part of the omnibus analysis with the `split` argument or here with `summary.lm`.

11.4.7 Using `emmeans` for 3 way interaction contrasts.

`emmeans` is a very inefficient way of obtaining higher order interaction contrasts. We cannot simply provide coefficient sets for each IV separately and have `emmeans` combine them. We have to create the vectors "manually", perhaps using matrix algebra tools. The logic is that `emmeans` can only evaluate a contrast if the coefficients are provided. (my approach was to create the "lincombs" objects). In other words, `emmeans` will not do the combinatorial work to produce interaction from knowledge of main effect vectors. We have to "manually" create those vectors.

We have established the contrasts for each IV above (the "main effect" coding vectors). They are.....

first, for wordtype:

```
wc <- contrasts(bg.3way$wordtype)
wc
```

```
      [,1] [,2]
LF_LE   -1    1
HF_LE   -1   -1
HF_HE    2    0
```

then for feedback:

```
fb <- contrasts(bg.3way$feedback)
fb
```

```
         [,1] [,2]
None        2    0
Positive   -1    1
Negative   -1   -1
```

and then for grade:

```
gc <- contrasts(bg.3way$grade)
gc
```

```
        [,1]
fifth      1
twelfth   -1
```

There are four interaction contrasts formed with the 3way product of the three sets of component vectors. We can create them using the same matrix logic that we used above.

This is tricky because we have to make certain that the order of the coefficients matches what will be in the grid of the `emmeans` output.

First I show the vector for the 2way combination of Wc1 and fb1, then the three way with grade.

```
wc1byfb1 <- wc[,1]%*%t(fb[,1])
wc1byfb1 <- matrix(unlist(wc1byfb1), nrow=1)
wc1byfb1
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]   -2   -2    4    1    1   -2    1    1   -2
```

```
wc1byfb1bygc <- t(wc1byfb1)%*%t(gc)
wc1byfb1bygc
```

```
      fifth twelfth
 [1,]    -2       2
 [2,]    -2       2
 [3,]     4      -4
 [4,]     1      -1
 [5,]     1      -1
 [6,]    -2       2
 [7,]     1      -1
 [8,]     1      -1
 [9,]    -2       2
```

The `emmeans` grid is produced here in a way that keeps the order of cells parallel to the order that the coefficient set describes.

```
wfg.emm <- emmeans(fit_base.afex,~wordtype:feedback:grade)
wfg.emm
```

```
 wordtype feedback grade   emmean    SE df lower.CL upper.CL
 LF_LE    None     fifth      8.4 0.615 72     7.17     9.63
 HF_LE    None     fifth      8.8 0.615 72     7.57    10.03
 HF_HE    None     fifth      8.0 0.615 72     6.77     9.23
 LF_LE    Positive fifth      7.8 0.615 72     6.57     9.03
```

```
 HF_LE    Positive fifth      8.0 0.615 72     6.77     9.23
 HF_HE    Positive fifth      4.4 0.615 72     3.17     5.63
 LF_LE    Negative fifth      8.0 0.615 72     6.77     9.23
 HF_LE    Negative fifth      7.6 0.615 72     6.37     8.83
 HF_HE    Negative fifth      3.8 0.615 72     2.57     5.03
 LF_LE    None     twelfth    8.4 0.615 72     7.17     9.63
 HF_LE    None     twelfth    8.8 0.615 72     7.57    10.03
 HF_HE    None     twelfth    7.8 0.615 72     6.57     9.03
 LF_LE    Positive twelfth    8.0 0.615 72     6.77     9.23
 HF_LE    Positive twelfth    8.2 0.615 72     6.97     9.43
 HF_HE    Positive twelfth    7.4 0.615 72     6.17     8.63
 LF_LE    Negative twelfth    8.4 0.615 72     7.17     9.63
 HF_LE    Negative twelfth    8.0 0.615 72     6.77     9.23
 HF_HE    Negative twelfth    7.0 0.615 72     5.77     8.23

Confidence level used: 0.95
```

Now we can pass those 18 coefficients to the contrast function, in the proper order and use `test`, as before, to evaluate that three way interaction contrast. Notice that the t value for this first of four 3way interaction contrasts matches what was produced by `summary.lm`

```
lincombs.3wayint <- contrast(wfg.emm,
                         list(intcontr1.3way=
                         c(-2,-2,4,1,1,-2,1,1,-2,2,2,-4,-1,-1,2,-1,-1,2)
                         ))
test(lincombs.3wayint, adjust="none")
```

```
 contrast       estimate   SE df t.ratio p.value
 intcontr1.3way       12 5.22 72   2.301  0.0243
```

The other three 3way interaction contrasts could be tested in the same manner. That is not done here since the 3way was not significant and the purpose of demonstration was accomplished with the first of the four.

11.4.7.1 Conclusions on 3 way interaction contrasts.

Obtaining the test of the 3way interaction contrasts with `emmeans` is considerable work. This illustration only outlined the way to obtain one of the four possible ones. This inefficiency is not desirable. It is simpler to use `summary.lm`. Neither approach produces SS and F tests or effect sizes. I will continue to prefer SPSS MANOVA.

11.5 Alpha rate adjustments wtih contrasts.

Recall that each use of the `emmeans::test` function had the capability of p value adjustments using the bonferroni/sidak/holm/fdr family of adjustments. Any of the sets of contrasts tested above might have used that argument, especially if the orthogonal sets were not `a priori`. The exception would be in the final contrast, the simple 2way interaction contrast where only one contrast was tested - thus nothing to be adjusted. However, if the full orthogonal set of four interaction contrasts were employed, then the adjustment might have made sense.

11.6 Pairwise follow up comparisons with `emmeans`

If a preference for follow up analyses chooses pairwise comparisons rather than contrasts, then `emmeans` permits that approach.

Consider the set of simple main effects of wordtype at levels of grade that was examined above. The grid of means is reproduced here as a reminder.

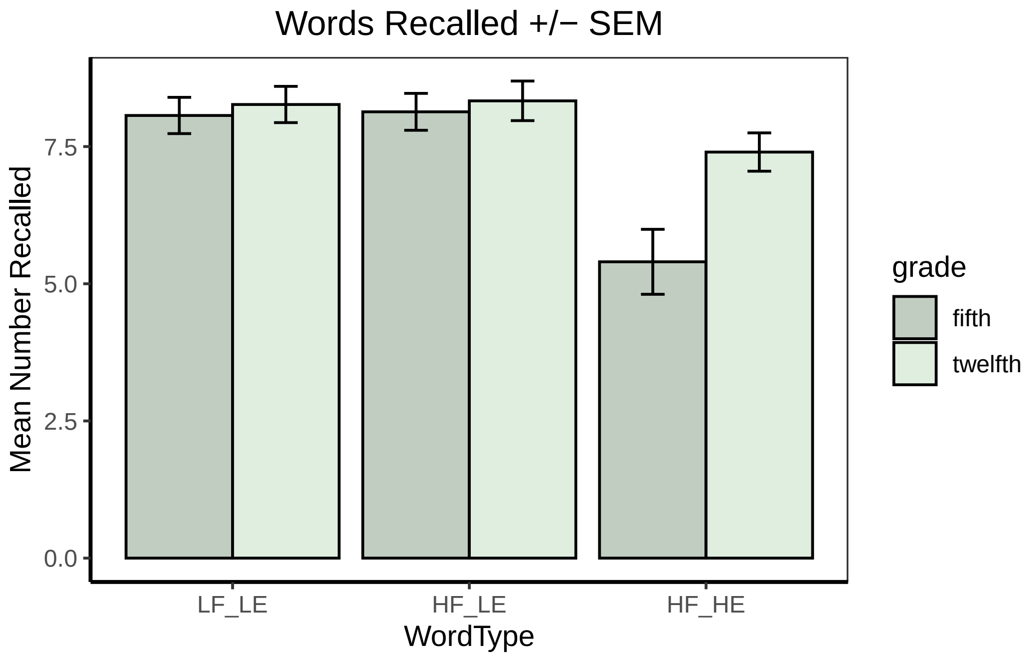
```
w.g.emm <- emmeans(fit_base.afex, "wordtype", by="grade")
```

```
NOTE: Results may be misleading due to involvement in interactions
```

```
w.g.emm
```

```
grade = fifth:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE      8.07 0.355 72     7.36     8.77
 HF_LE      8.13 0.355 72     7.43     8.84
 HF_HE      5.40 0.355 72     4.69     6.11

grade = twelfth:
 wordtype emmean    SE df lower.CL upper.CL
 LF_LE      8.27 0.355 72     7.56     8.97
 HF_LE      8.33 0.355 72     7.63     9.04
 HF_HE      7.40 0.355 72     6.69     8.11

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

The plot of those marginal means in the wordtype by grade layout is also reproduced here for visual reference.



If we wanted to examine pairwise difference among the levels of wordtype, separately at levels of grade we could use the `pairs` function. In its default format, this produces all three pairwise t-tests separately at each level of grade (six tests in all) where all use the pooled within cell error term and its 72 df rather than the error specific to the two cells involved.

Unsurprisingly the HF_HE group differs from the other two levels of wordtype in fifth graders but no differences were found in 12th graders.

```
pairs(w.g.emm, adjust="none")
```

```
grade = fifth:
 contrast      estimate    SE df t.ratio p.value
 LF_LE - HF_LE  -0.0667 0.502 72  -0.133  0.8947
 LF_LE - HF_HE   2.6667 0.502 72   5.314  <.0001
 HF_LE - HF_HE   2.7333 0.502 72   5.447  <.0001

grade = twelfth:
 contrast      estimate    SE df t.ratio p.value
 LF_LE - HF_LE  -0.0667 0.502 72  -0.133  0.8947
 LF_LE - HF_HE   0.8667 0.502 72   1.727  0.0885
 HF_LE - HF_HE   0.9333 0.502 72   1.860  0.0670

Results are averaged over the levels of: feedback
```

If these were explicitly *post hoc* pairwise comparisons, then it would be appropriate to do p value adjustments for error rate inflation. One nice thing about the pairs.emmGrid function is its capability for a tukey adjustment method in addition to the bonferroni/sidak/fdr family. Note that the adjustment is done within families of pairwise comparisons - here that means adjusting for three tests at each level of grade separately, not for a total of six tests.

```
pairs(w.g.emm, adjust="tukey")
```

```
grade = fifth:
 contrast      estimate    SE df t.ratio p.value
 LF_LE - HF_LE  -0.0667 0.502 72  -0.133  0.9903
 LF_LE - HF_HE   2.6667 0.502 72   5.314  <.0001
 HF_LE - HF_HE   2.7333 0.502 72   5.447  <.0001

grade = twelfth:
 contrast      estimate    SE df t.ratio p.value
 LF_LE - HF_LE  -0.0667 0.502 72  -0.133  0.9903
 LF_LE - HF_HE   0.8667 0.502 72   1.727  0.2022
 HF_LE - HF_HE   0.9333 0.502 72   1.860  0.1579

Results are averaged over the levels of: feedback
P value adjustment: tukey method for comparing a family of 3 estimates
```

There is a way to produce effect size statistics (Cohen's d) for these pairwise comparisons. See the help documentation for the `eff_size` function in **emmeans**.

12 Additional Post Hoc Tests

Tukey's HSD is easily accomplished on an AOV fit via a function in the base stats package that is loaded upon startup note that this compares all pairs of cell means. It would severely over correct for alpha inflation since not all of the pairs of cell means are interesting comparisons (comparing applies and oranges in many instances). I cannot recommend using this method for a factorial.

The only satisfactory solutions that I've seen for application of *Post Hoc* tests variously to marginal tables of means collapsed from the full factorial table or for arrays of means in simple effect configurations are those provided in the `emmeans` suite of tools. We saw the availability of the `adjust` argument in both the contrasts approach and the pairwise comparison approach. Analogous methods may be found in the **phia** approach.

13 Reproducibility

Ver 1.3 March 31, 2025

- Converted document to Quarto
- Cleaned up some language and added needed exposition
- Corrected a few typos

Ver 1.2 April 17, 2023

- major revision of much language and sequencing of approach

- added major sections on follow up analyses with the emmeans and phia packages

Ver 1.1 March 31, 2021

- edited graph styles

Ver 1.0 Sep 9, 2020

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 22631)

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

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base
```

```
other attached packages:
 [1] gridExtra_2.3       qqplotr_0.0.6       e1071_1.7-16
 [4] tibble_3.2.1        sjstats_0.19.0      sciplot_1.2-0
 [7] Rmisc_1.5.1         lattice_0.22-6      psych_2.4.12
[10] plyr_1.8.9          phia_0.3-1          nortest_1.0-4
[13] lmtest_0.9-40       zoo_1.8-12          knitr_1.49
[16] gt_0.11.1           ggrain_0.0.4        ggthemes_5.1.0
[19] ggplot2_3.5.1       emmeans_1.10.6      effectsize_1.0.0
[22] car_3.1-3           carData_3.0-5       bcdstats_0.0.0.9009
[25] afex_1.4-1          lme4_1.1-36         Matrix_1.7-1

loaded via a namespace (and not attached):
  [1] RColorBrewer_1.1-3      vcd_1.4-13              rstudioapi_0.17.1
  [4] jsonlite_1.8.9          datawizard_1.0.0        magrittr_2.0.3
  [7] TH.data_1.1-2           estimability_1.5.1      farver_2.1.2
 [10] pwr_1.3-0               nloptr_2.1.1            rmarkdown_2.29
 [13] vctrs_0.6.5             Cairo_1.6-2             minqa_1.2.8
 [16] base64enc_0.1-3         htmltools_0.5.8.1       polynom_1.4-1
 [19] plotrix_3.8-4           Formula_1.2-5           pracma_2.4.4
 [22] HH_3.1-52               htmlwidgets_1.6.4       sandwich_3.1-1
 [25] iterators_1.0.14        mime_0.12               lifecycle_1.0.4
 [28] pkgconfig_2.0.3         R6_2.5.1                fastmap_1.2.0
 [31] rbibutils_2.3           shiny_1.10.0            digest_0.6.37
 [34] numDeriv_2016.8-1.1     colorspace_2.1-1        miscTools_0.6-28
 [37] Hmisc_5.2-2             labeling_0.4.3          abind_1.4-8
 [40] compiler_4.4.2          proxy_0.4-27            doParallel_1.0.17
 [43] withr_3.0.2             htmlTable_2.4.3         backports_1.5.0
 [46] performance_0.13.0      MASS_7.3-64             quantreg_5.99.1
 [49] ggpp_0.5.8-1            caTools_1.18.3          tools_4.4.2
 [52] foreign_0.8-88          httpuv_1.6.15           qqconf_1.3.2
 [55] nnet_7.3-20             glue_1.8.0              dabestr_2023.9.12
 [58] nlme_3.1-166            promises_1.3.2          checkmate_2.3.2
 [61] cluster_2.1.8           reshape2_1.4.4          generics_0.1.3
 [64] gtable_0.3.6            shinyBS_0.61.1          class_7.3-23
 [67] data.table_1.16.4       xml2_1.3.6              foreach_1.5.2
 [70] pillar_1.10.1           stringr_1.5.1           later_1.4.1
 [73] robustbase_0.99-4-1     splines_4.4.2           dplyr_1.1.4
 [76] gghalves_0.1.4          survival_3.8-3          gmp_0.7-5
 [79] deldir_2.0-4            SparseM_1.84-2          tidyselect_1.2.1
 [82] reformulas_0.4.0        xfun_0.50               yacca_1.4-2
 [85] DEoptimR_1.1-3-1        stringi_1.8.4           yhat_2.0-4
 [88] yaml_2.3.10             boot_1.3-31             evaluate_1.0.3
 [91] codetools_0.2-20        interp_1.1-6            twosamples_2.0.1
```

```
 [94] cli_3.6.3              rpart_4.1.24           pbmcapply_1.5.1
 [97] xtable_1.8-4           parameters_0.24.1      Rdpack_2.6.2
[100] munsell_0.5.1          Rcpp_1.0.14            coda_0.19-4.1
[103] png_0.1-8              parallel_4.4.2         leaps_3.2
[106] MatrixModels_0.5-3     bayestestR_0.15.0      latticeExtra_0.6-30
[109] jpeg_0.1-10            bitops_1.0-9           opdisDownsampling_1.0.1
[112] Rmpfr_1.0-0            mvtnorm_1.3-3          lmerTest_3.1-3
[115] scales_1.3.0           insight_1.0.1          purrr_1.0.2
[118] rlang_1.1.4            mnormt_2.1.1           multcomp_1.4-26
```