

610 – R6 Three-way Between Groups Designs, Balanced-N

Written by Colleen F. Moore, University of Wisconsin-Madison, Psychology Dept
For Psychology 610

This uses the example on Table 21.1,p. 466 of Keppel & Wickens.

It also uses two packages that you need to install, `sciplot`, and `gplots`. `gplots` requires that you install two or three other packages. All these add-ons are small, so don't worry. See below for instructions on installing these.

Contents of this handout:

Section I: the overall anova and graphs, including how to plot multiple graphs on one page

Section II: simple interactions

Section III: Tukey pair-wise tests for main effect means and cell means

I. Overall anova

I A. Prepare data and bring into R.

```
> your.data=read.table(pipe("pbpaste"),header=T) # paste from the clipboard
```

```
> your.data
```

```
Feedback WordType Grade numrecall
1          1          1     5          7
2          1          1     5          7
3          1          1     5          9
4          1          1     5         10
5          1          1     5          9
6          1          2     5          7
7          1          2     5          8
8          1          2     5          9
9          1          2     5         10
10         1          2     5         10
11         1          3     5          7
.
.
.
83         3          2    12          6
84         3          2    12          9
85         3          2    12          9
86         3          3    12          6
87         3          3    12          5
88         3          3    12          7
89         3          3    12          8
90         3          3    12          9
```

```
> attach(your.data)
```

```
> F=factor(Feedback) # make factors of the numerical codes
```

```
> W=factor(WordType)
```

```
> G=factor(Grade)
```

I B. Do the anova

```
> threeway.aov=aov(numrecall~F*W*G) # specify full model with all main effects and interactions by 'F*W*G'
```

```
> summary(threeway.aov,intercept=T)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
(Intercept)	1	5198.4	5198.4	2752.0941	< 2.2e-16	***
F	2	26.9	13.4	7.1118	0.001519	**
W	2	64.9	32.4	17.1706	7.993e-07	***
G	1	14.4	14.4	7.6235	0.007304	**
F:W	4	14.7	3.7	1.9412	0.112829	
F:G	2	8.6	4.3	2.2765	0.109987	
W:G	2	16.2	8.1	4.2882	0.017397	*
F:W:G	4	10.0	2.5	1.3235	0.269461	
Residuals	72	136.0	1.9			

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have significant main effects of Feedback, Wordtype and Grade, and an interaction of WordType x Grade.

I C. Check that the design is balanced:

```
>newdata=data.frame(numrecall,F,W,G) # I make a new data frame with just the factors and dv because I have those numerical variables in the original data.
```

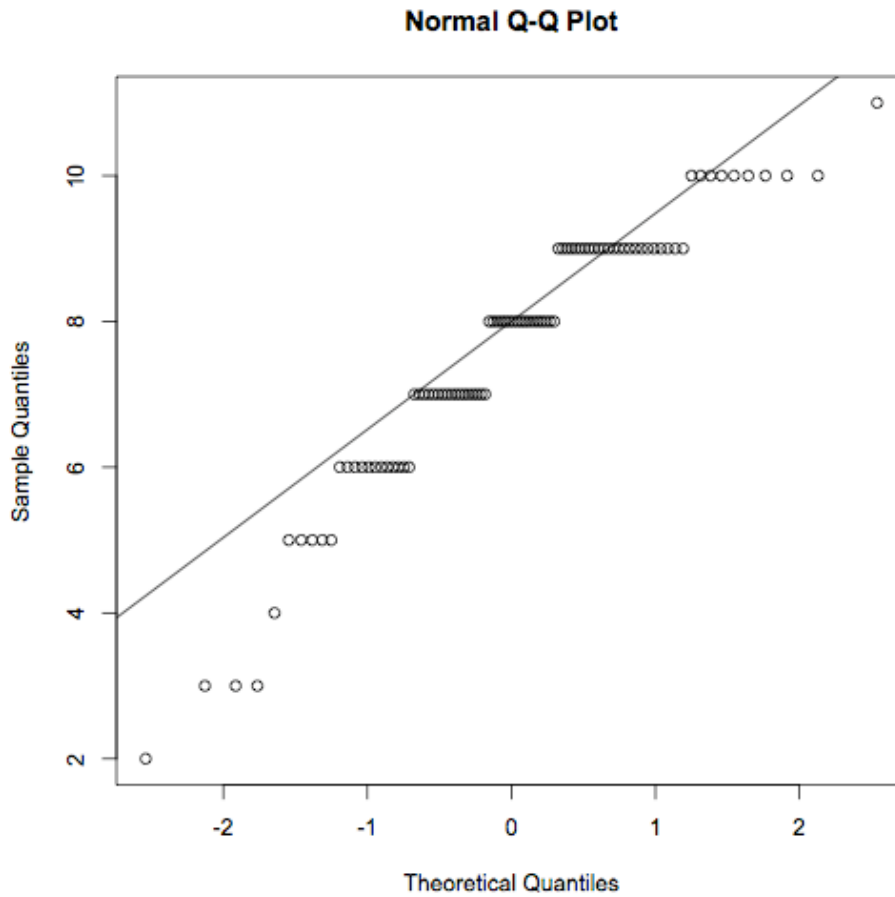
```
> !is.list(replications(numrecall~F*W*G, data=newdata))
```

```
[1] TRUE
```

I D. Make the qq-normal plot and examine:

```
> qqnorm(numrecall)
```

```
> qqline(numrecall)
```



The data deviate from the line quite a bit. Some kind of transformation is probably needed to make the data closer to normal. I am surprised to see this in Keppel's example. But we'll go ahead with the data as they are.

E. Find means and se's

```
> model.tables(threeway.aov,"means",se=T) # tables of means and se's
```

```
Tables of means  
Grand mean
```

```
7.6
```

```
F  
F  
  1    2    3  
8.367 7.300 7.133
```

```
W  
W  
  1    2    3  
8.167 8.233 6.400
```

```
G  
G  
  5  12
```

7.2 8.0

```
F:W
  W
F  1  2  3
1  8.4 8.8 7.9
2  7.9 8.1 5.9
3  8.2 7.8 5.4
```

```
F:G
  G
F  5 12
1  8.400 8.333
2  6.733 7.867
3  6.467 7.800
```

```
W:G
  G
W  5 12
1  8.067 8.267
2  8.133 8.333
3  5.400 7.400
```

```
F:W:G
, , G = 5
```

```
  W
F  1  2  3
1  8.4 8.8 8.0
2  7.8 8.0 4.4
3  8.0 7.6 3.8
```

```
, , G = 12
```

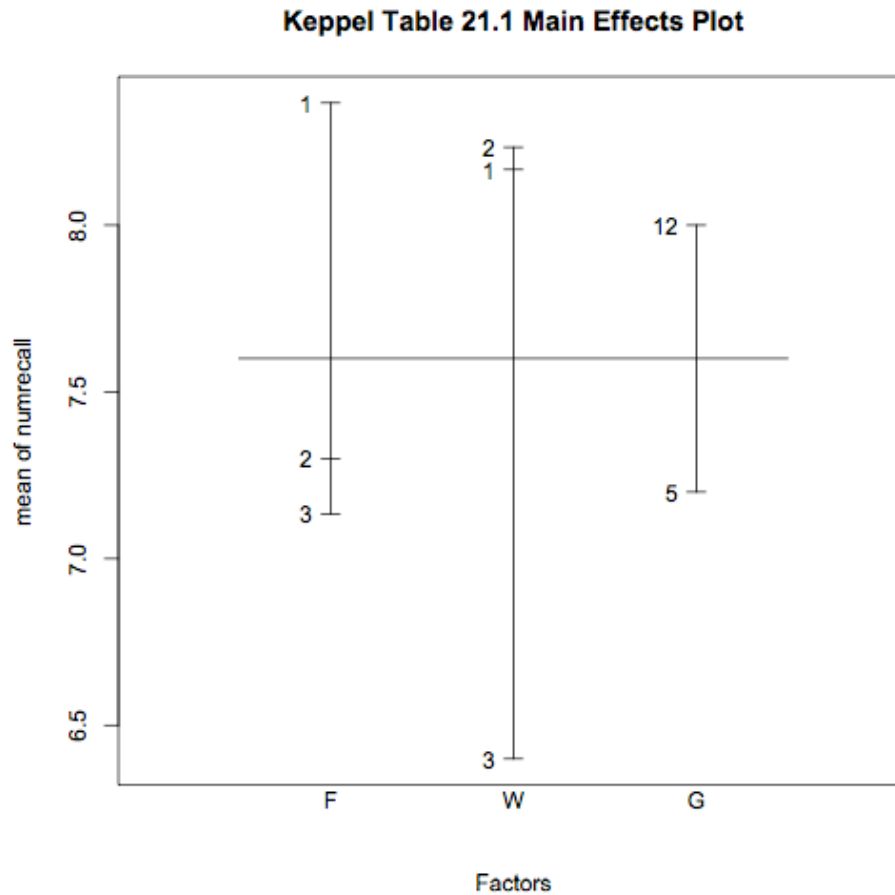
```
  W
F  1  2  3
1  8.4 8.8 7.8
2  8.0 8.2 7.4
3  8.4 8.0 7.0
```

Standard errors for differences of means

	F	W	G	F:W	F:G	W:G	F:W:G
replic.	0.3549	0.3549	0.2897	0.6146	0.5018	0.5018	0.8692
	30	30	45	10	15	15	5

I E. Graph main effects:

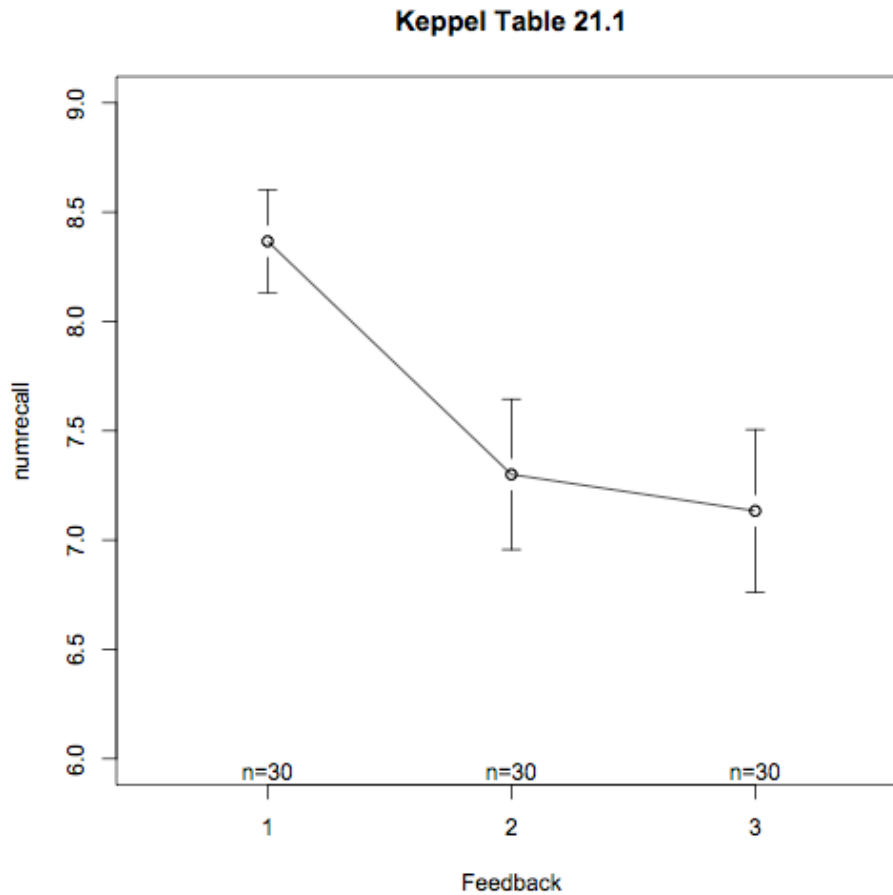
> plot.design(numrecall~F*W*G,data="your.data",main="Keppel Table 21.1 Main Effects Plot") # the numbers on each vertical line indicate the position of the mean for the different levels of each factor.



Another option for graphing main effects:

```
> library(gplots) # first, install gplots and gtools and gdata, then you only  
need to say 'library' for gplots itself. I kept adding packages until 'gplots'  
came into the library without a message about another package.
```

```
> plotmeans(numrecall~F,xlab="Feedback",ylab="numrecall", p=.68,  
ylim=c(6.0,9.0),main="Keppel Table 21.1",barcol="black") # this plots a line  
graph with error bars. After seeing the main effect plot above, I set 'ylim' to  
values that I thought would work for all 3 main effect graphs.
```

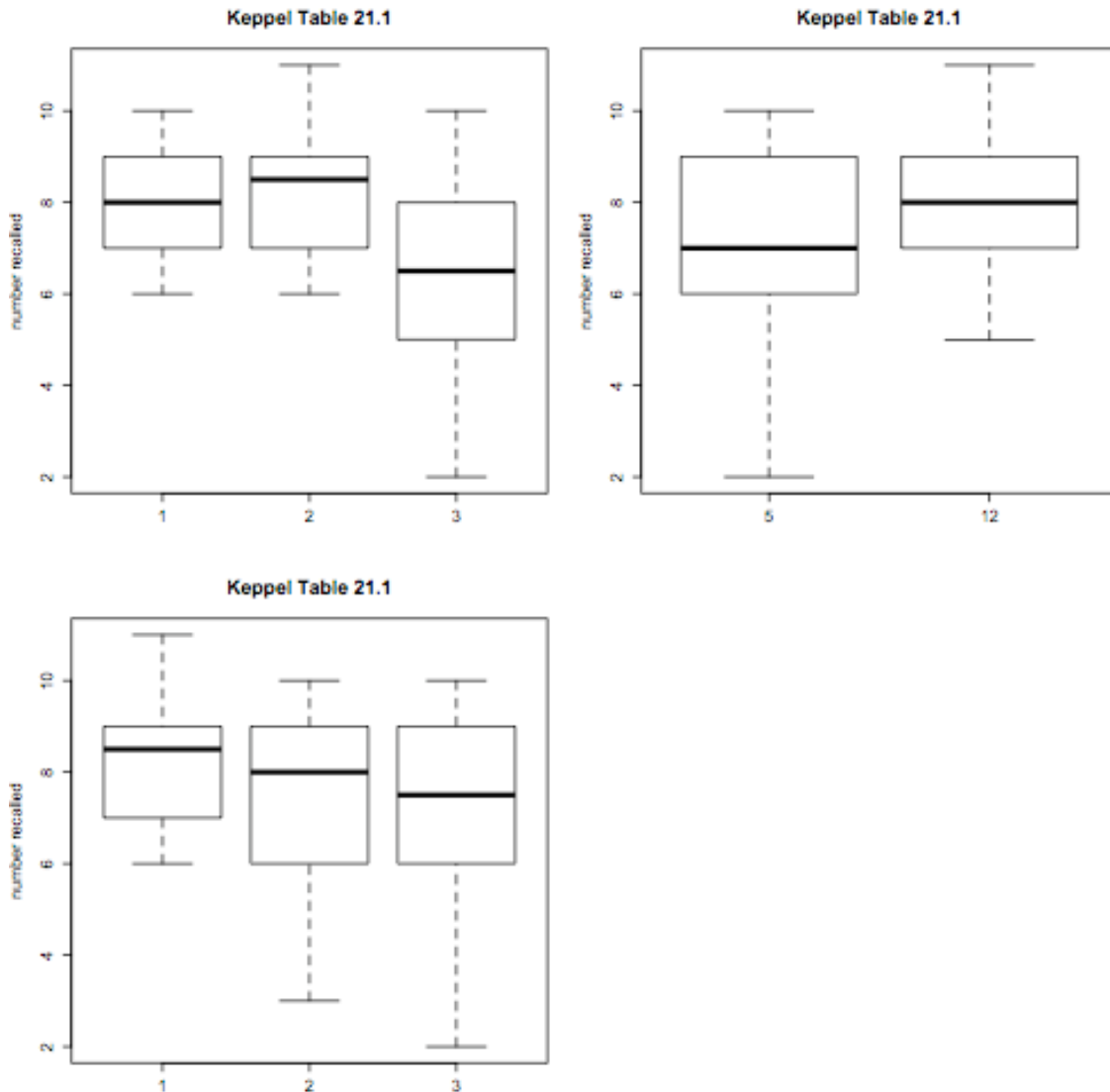


The error bars are done separately for each level of Feedback so these are likely based on the overall sd's rather than using MSerror. I prefer the model-based se's given by the 'model.tables' statement above.

Or plot **box-n-whiskers** for all 3 main effects on one page:

```
> par(mfrow=c(2, 2), cex=0.6, mar=c(4, 4, 4, 2), mex=0.8) # 'mfrow' partitions the plot window into 2 rows and 2 columns. 'mar' sets the margins.
```

```
> plot(numrecall~W*G*F,main="Keppel Table 21.1", xlab=NULL, ylab="number recalled") # box'n'whiskers is default for 'plot' when factors are specified.
```



Plot all three two-way interactions on one page. You need to repeat the ‘par’ statement after you close a graph.

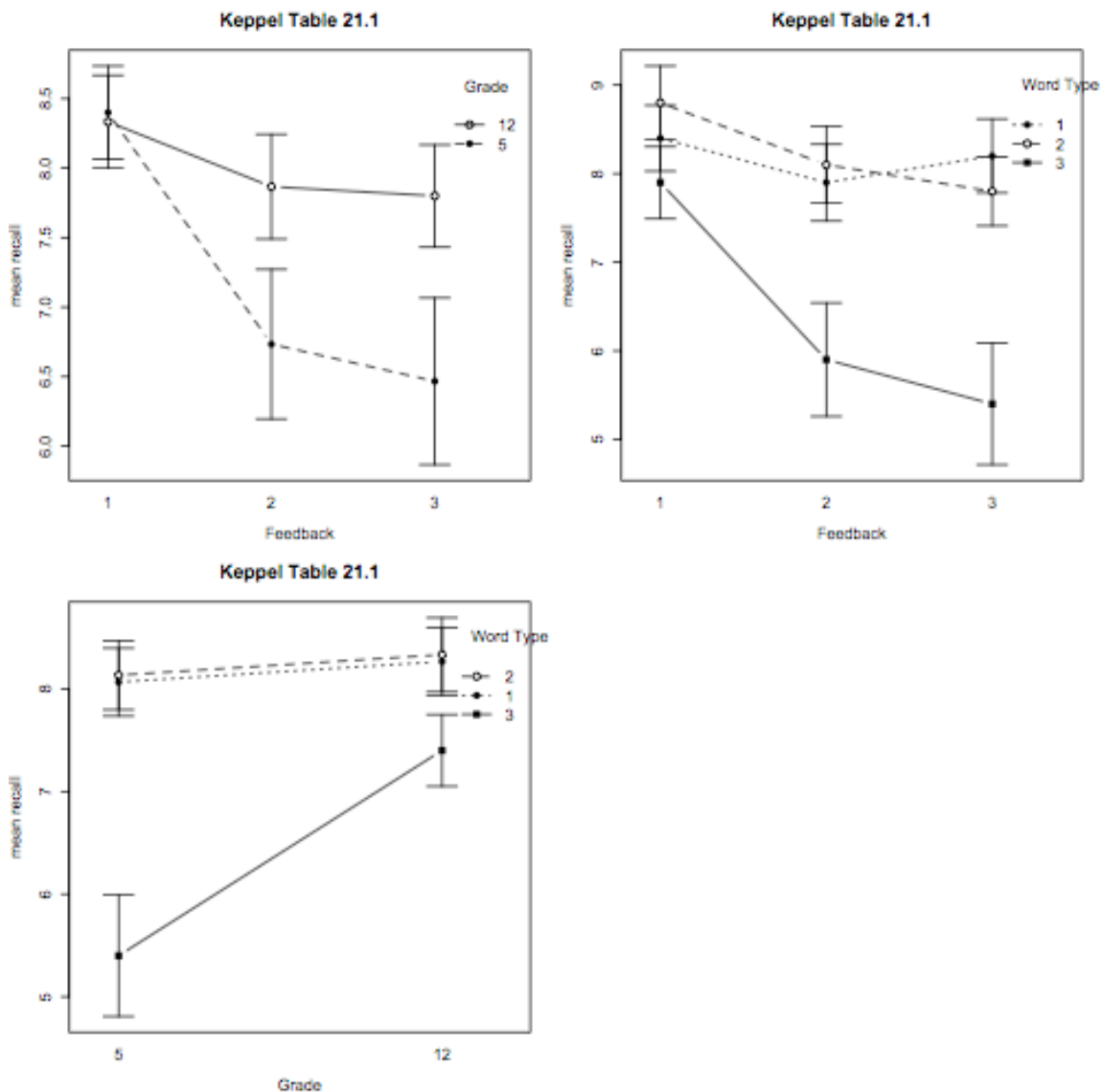
```
> library(sciplot) # bring sciplot package into environment
> par(mfrow=c(2, 2), cex=0.6, mar=c(4, 4, 4, 2), mex=0.8) # set up to make up to 4
graphs on one page, in a 2x2 arrangement
> lineplot.CI(F, numrecall, group=G , type="b", legend=TRUE, trace.label="Grade",
leg.lab=NULL, fixed=FALSE, x.legend=NULL, y.legend=NULL, cex.legend=1, ncol=1,
xlab="Feedback",ylab="mean recall", pch=c(16, 21, 15, 22, 17, 24, c(3:14)),
xlim=NULL, ylim=NULL, cex=NULL, lwd=NULL, col="black", cex.axis=1, xaxt="s",
data=NULL, subset=NULL, main="Keppel Table 21.1") # Plot the F x G interaction
```

```
> lineplot.CI(F, numrecall, group=W , type="b", legend=TRUE, trace.label="Word
Type", leg.lab=NULL, fixed=FALSE, x.legend=NULL, y.legend=NULL, cex.legend=1, ncol=1,
xlab="Feedback",ylab="mean recall", pch=c(16, 21, 15, 22, 17, 24, c(3:14)),
```

`xlim=NULL, ylim=NULL, cex=NULL, lwd=NULL, col="black", cex.axis=1, xaxt="s", data=NULL, subset=NULL, main="Keppel Table 21.1") #plot the F x W interaction.`
 Change the 'group' and the 'trace.label' parameters

`> lineplot.CI(G, numrecall, group=W, type="b", legend=TRUE, trace.label="Word Type", leg.lab=NULL, fixed=FALSE, x.leg=NULL, y.leg=NULL, cex.leg=1, ncol=1, xlab="Grade", ylab="mean recall", pch=c(16, 21, 15, 22, 17, 24, c(3:14)), xlim=NULL, ylim=NULL, cex=NULL, lwd=NULL, col="black", cex.axis=1, xaxt="s", data=NULL, subset=NULL, main="Keppel Table 21.1") #plot the G x W interaction.` Change the first factor named, and the 'xlab' parameters

I left 'ylim' as 'NULL', but it would be helpful to set ylim to the same value for these 3 plots to facilitate comparison.



Section II, Simple interaction test

A simple interaction test of FxW at each grade level separately.

We don't have a significant 3-way interaction in these data. But if we did, we would partition the data by grade and do anovas on each grade. By partitioning the data and doing the anovas, these tests use partitioned error. We then require homogeneity of variance over only the grade level being considered. If you want these tests to use the pooled error from the overall anova, just do a touch of hand calculations using the original error term. You can use R to look up the exact p-value of your calculated tests. Notice that in the partitioned tests below, the df for the partitioned error is smaller than the df for error from the overall anova.

--First, make a data frame with only the 5th graders, and another data frame with only the 12th graders:

```
> Fifthdata=subset(your.data,Grade==5) # double == sign is logical. I used the original
numerical variable called 'Grade'.
> Twelfthdata=subset(your.data,Grade==12)
> detach(your.data) # detaches original data.
> attach(Fifthdata) # attach the 5th grade data
> rm(F) # remove the factor for feedback because it goes with the full data set
> rm(W) # remove factor W for the full data also. It has too many rows
> F=factor(Feedback) # the new factor F will have the correct number of rows, same as
the 5th grade sample
> W=factor(WordType)
```

Do Anova for 5th graders only:

```
> Grade5.aov=aov(numrecall~F*W, data=Fifthdata) # do two-way anova on just the 5th
graders
```

```
> summary(Grade5.aov,intercept=T)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
(Intercept)	1	2332.80	2332.80	1312.2000	< 2.2e-16	***
F	2	32.93	16.47	9.2625	0.0005685	***
W	2	72.93	36.47	20.5125	1.133e-06	***
F:W	4	23.33	5.83	3.2812	0.0215830	*
Residuals	36	64.00	1.78			

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5th graders show sig effects of Feedback, Word Type, and an FxW interaction.

Graph the FxW interaction for the 5th graders.

Bar graph with error bars, built from scratch.

Step 1: Make matrix of 5th grade means, and label the rows.

```
> fifthmeans=tapply(numrecall, IND=list(F,W),mean)
> rownames(fifthmeans)=c("Word type 1","Word type 2","Word type 3")
> fifthmeans
```

	1	2	3
Word type 1	8.4	8.8	8.0
Word Type 2	7.8	8.0	4.4

Word Type 3 8.0 7.6 3.8

Step 2: Find the estimated se's. Can use model-based or base on the cell sd.

2a) Model-based. Use MSresidual from either 5th grade or overall anova.

```
> estse5th=sqrt(1.78 / 5) # est se is sqrt(MSres divided by n)
```

```
> estse5th
```

```
[1] 0.5966574
```

```
> se5th=matrix(c(estse5th),3,3) # this puts the same value in the whole 3x3 matrix
```

```
> se5th
```

```
      [,1]      [,2]      [,3]
[1,] 0.5966574 0.5966574 0.5966574
[2,] 0.5966574 0.5966574 0.5966574
[3,] 0.5966574 0.5966574 0.5966574
```

2b) Or, use cell sd's as basis for se's.

```
> se5th=tapply(numrecall, IND=list(F,W),sd)
```

```
> se5th=se5th/sqrt(5) # cell n is 5
```

```
> se5th
```

```
      1      2      3
1 0.6000000 0.5830952 0.6324555
2 0.5830952 0.5477226 0.6000000
3 0.6324555 0.6000000 0.5830952
```

Now set up to put two graphs on a page, because we'll add the 12th graders shortly:

```
> par(mfrow=c(1, 2), cex=0.6, mar=c(28, 4, 4, 2), mex=0.8)
```

change the first margin parameter 'mar' to shrink the graphs vertically so that you like the proportions. With only one row of figures they will be tall and skinny unless you put something large there.

```
> fifthplot=barplot(fifthmeans,beside=T,ylim=c(0,12),main="Fifth
```

```
Graders",xlab="Feedback", ylab="mean recall", axes=T, legend.text=T, offset=0)
```

```
> superpose.eb(fifthplot,fifthmeans,se5th,col="black",lwd=1) # make sure you have entered this function prior to trying to use it.
```

```
> box() # put a box around the plot.
```

Repeat everything for the 12th graders:

```
> detach(Fifthdata)
```

```
> attach(Twelfthdata)
```

```
> Twelve.aov=aov(numrecall~F*W)
```

```
> summary(Twelve.aov,intercept=T)
```

```
      Df Sum Sq Mean Sq  F value Pr(>F)
(Intercept) 1 2880.00 2880.00 1440.0000 <2e-16 ***
F            2    2.53    1.27    0.6333 0.5366
W            2    8.13    4.07    2.0333 0.1457
F:W          4    1.33    0.33    0.1667 0.9539
Residuals   36   72.00    2.00
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The 12th graders have no significant effects. Next, find means and graph.

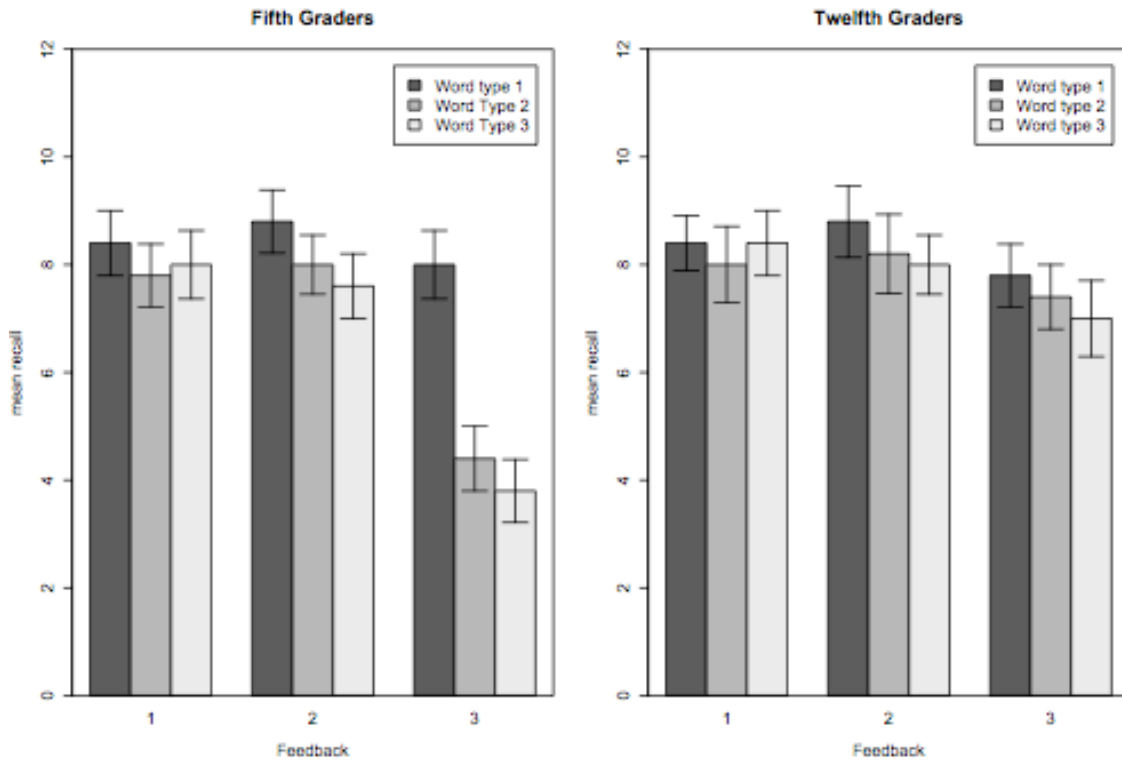
```
> twelvemeans=tapply(numrecall, IND=list(F,W),mean)
> rownames(twelvemeans)=c("Word type 1","Word type 2","Word type 3")
> twelvemeans
      1  2  3
Word type 1 8.4 8.8 7.8
Word type 2 8.0 8.2 7.4
Word type 3 8.4 8.0 7.0
```

```
> twelverse=tapply(numrecall, IND=list(F,W),sd) # find cell sd's
> twelverse=twelverse/sqrt(5) # convert sd to se
> twelverse
      1  2  3
1 0.5099020 0.6633250 0.5830952
2 0.7071068 0.7348469 0.6000000
3 0.6000000 0.5477226 0.7071068
```

Now graph the 12th graders:

```
> twelveplot=barplot(twelvemeans,beside=T,ylim=c(0,12),main="Twelfth
Graders",xlab="Feedback", ylab="mean recall", axes=T, legend.text=T,
offset=0)
> superpose.eb(twelveplot,twelvemeans,twelverse,col="black",lwd=1)
> box()
```

Here's the final graph with both grades on one page:



Section III. Post-hoc tests and contrasts

III A. TukeyHSD tests of pairwise differences.

First I asked for the pairwise tests of just the Feedback factor levels. This does the pairwise tests on the main effect means.

`> TukeyHSD(all.aov,"F")` # list the factors in quotes. The default is all possible pairwise tests. 'all.aov' is the place where I stored my overall anova.

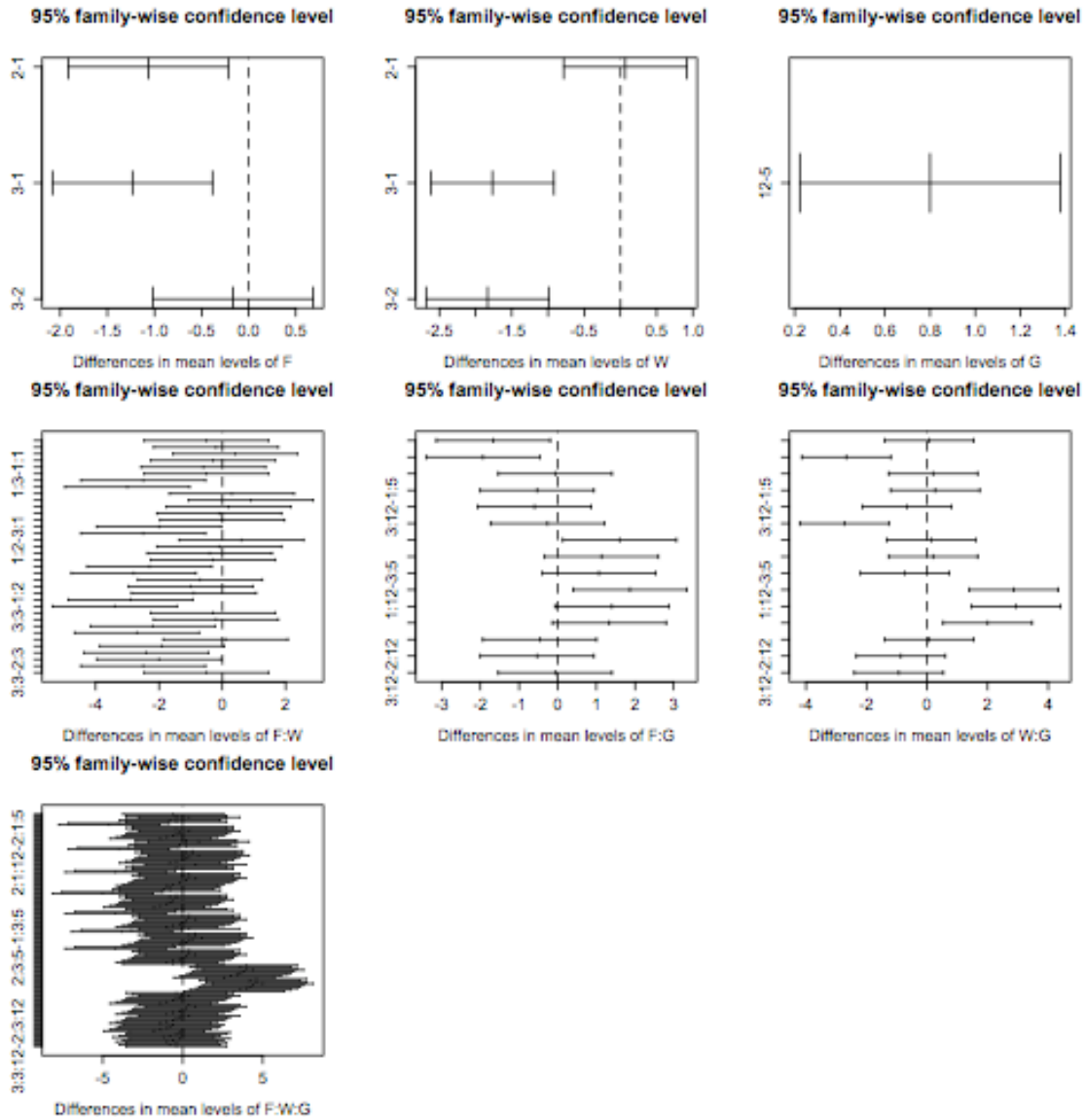
```
Tukey multiple comparisons of means  
95% family-wise confidence level
```

```
Fit: aov(formula = numrecall ~ F * W * G)
```

```
$F
      diff      lwr      upr      p adj
2-1 -1.0666667 -1.915892 -0.2174417 0.0100711
3-1 -1.2333333 -2.082558 -0.3841084 0.0024719
3-2 -0.1666667 -1.015892  0.6825583 0.8857204
```

Also can graph the Tukey confidence intervals:

```
> par(mfrow=c(3, 3), cex=0.6, mar=c(4, 4, 4, 2), mex=0.8) ## set up to make one page
of graphs arranged in a 3x3 grid
> plot(TukeyHSD(all.aov,"F")); abline(v=0, lty=2) ## graph the CI's for the main effect
of F. Add a vertical line at zero so you can judge significance by eye. 'lty=2' tells R to
make a dashed line.
> plot(TukeyHSD(all.aov,"W")); abline(v=0, lty=2)
> plot(TukeyHSD(all.aov,"G")); abline(v=0, lty=2)
> plot(TukeyHSD(all.aov,"F:W")); abline(v=0, lty=2) ## test pairwise diffs for F x W
means
> plot(TukeyHSD(all.aov,"F:G")); abline(v=0, lty=2)
> plot(TukeyHSD(all.aov,"W:G")); abline(v=0, lty=2)
> plot(TukeyHSD(all.aov,"F:W:G")); abline(v=0, lty=2) ## with 18 cells in the design,
there are so many pairwise differences that you can't really read the graph, especially
when it is small. Could do this one on a separate page by itself.
```



III B. Contrasts. (to be added later).