610 R8 **Prof Colleen F. Moore**

One Way Within-participant or Repeated-measures Analysis of Variance, Balanced Designs

For Psychology 610, University of Wisconsin--Madison.

This tutorial uses data in Table 16.3, Keppel & Wickens, p. 355.

Install the package "car"

Contents of this handout:

I. Basic data setup and analysis, including sphericity and H-F adjusted p-values, means and estimated standard errors

II. Contrasts with partitioned error

III. Line graphs and bar graph with error bars for one-way within design. Note: you need to calculate the estimated standard error from the residual of the anova model.

'Quick Look' Summary of R code for One-way within:

> library(car)

> multmodel=lm(cbind(A1,A2,A3) ~ 1) # make a multivariate linear model with only the intercept as a predictor for your within-participants observations

> Trials=factor(c("A1","A2","A3"), ordered=F) # create a factor for your repeated-measures variable.

> model1=Anova(multmodel,idata=data.frame(Trials),idesign=~Trials,type="III") # use the repeated-measures factor as the 'internal' part of the design using 'Anova' (with a capital A)

I. One-way within-subjects anova -- basic data setup and analysis.

A. Start R and bring in the data.

> library(car) # bring the 'car' package into the environment

> options(contrasts=c("contr.sum","contr.poly")) # set options for contrasts. Not necessary here, but a good practice if you will be analyzing any unbalanced designs

> your.data=read.table(pipe("pbpaste"),header=T) # I copied the data to the clipboard from the excel sheet

> your.data

```
   A1  A2  A3 subj
1 745 764 774   s1
2 777 786 788   s2
3 734 733 763   s3
4 779 801 797   s4
5 756 786 785   s5
6 721 732 740   s6
```

> attach(your.data) # some warn against attaching data.

B. Carry out the analysis using 'Anova' in the 'car' package in two steps.

By using the 'car' package we obtain the sphericity test and p-values adjusted by both the Huynh-Feldt or Greenhouse-Geisser method.

Step 1. Use 'lm' on all the repeated-measures variables together
> multmodel=lm(cbind(A1,A2,A3) ~ 1) # we column bind the 3 response columns together, and use only the intercept as our predictor variable.

Step 2. **Construct the repeated measures variable** and finish the analysis using 'Anova' (capital A) in the 'car' package.
> Trials=factor(c("A1","A2","A3"), ordered=F) # create a factor called "Trials" with labels A1 to A3.
> Trials
```
[1] A1 A2 A3
Levels: A1 A2 A3
```
> model1=Anova(multmodel,idata=data.frame(Trials),idesign=~Trials,type="III") # We name the model we established in Step 1. Then the 'idata' parameter is for the repeated-measures part of the data, the 'idesign' is where you specify the repeated part of the design. This is a one-way within design, so we have only the 'Trials' variable.
> summary(model1,multivariate=F) # omit the 'multivariate=F' part, and you will get the MANOVA results as well.

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

                  SS num Df Error SS den Df        F    Pr(>F)
(Intercept) 10520285      1     8548      5 6153.773 6.378e-09 ***
Trials          1575      2      546     10   14.432  0.001128 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Mauchly Tests for Sphericity

       Test statistic p-value
Trials        0.75802 0.57459


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

        GG eps Pr(>F[GG])
Trials 0.80516   0.002860 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

       HF eps Pr(>F[HF])
Trials 1.1302   0.001128 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Warning message:
In summary.Anova.mlm(model1, multivariate = F) : HF eps > 1 treated as 1
```

*** **Check that the df's are correct!** 'Trials' has 3 levels, and should have 2 df. The error for Trials should be Trials x Subjects interaction, df = (t-1)(n-1) in Keppel's notation. We have 6 people, so df error should be 10.

II. Contrasts with partitioned error.

A. Pairwise test of means, without post-hoc alpha-adjustment
This is easiest to do with the t-test function.

> c1=t.test(A1,A2,alternative="two.sided",mu=0,paired=T) # A1 and A2 are the variable names to test. We say 'paired=True' so that R will do a paired t-test because this is a within-S design.
> c1

```
      Paired t-test

data:  A1 and A2
t = -3.3597, df = 5, p-value = 0.02011
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -26.476815  -3.523185
sample estimates:
mean of the differences
                    -15
```

> c2=t.test(A1,A3,alternative="two.sided",mu=0,paired=T)
> c2

```
      Paired t-test

data:  A1 and A3
t = -7.2181, df = 5, p-value = 0.0007958
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -30.51291 -14.48709
sample estimates:
mean of the differences
                  -22.5
```

> c3=t.test(A2,A3,alternative="two.sided",mu=0,paired=T)
> c3

```
      Paired t-test

data:  A2 and A3
t = -1.5025, df = 5, p-value = 0.1933
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -20.33147   5.33147
sample estimates:
mean of the differences
                   -7.5
```

B. Adjust the p-values using Holm.
> pvec=c(0.02011, 0.0007958, 0.1933) # make a vector of p-values from the pairwise tests
> p.adjust(pvec,method="holm",n=3) # adjust p's by Holm method. Both A1 vs A2 and A1 vs A3 remain significant

```
[1] 0.0402200 0.0023874 0.1933000
```

C. Calculate partitioned error contrasts from the data matrix, and then test Ho that psi equals zero with 'aov' or with a t-test. However, we have headings on our data, so we have to bind the data together column-wise ('cbind') and then operate on that.

1. Repeat one of the pairwise contrasts we did above and make sure it matches the t-test results.

> contr1=c(1,-1) # make a vector of contrast values
> contr1
```
[1]  1 -1
```

> xc1=cbind(A1,A2)%*% contr1 # '%*%' is the symbol for matrix multiplication. So this multiplies A1 by 1 and A2 by –1, and puts the result in the vector that I named 'xc1'. 'xc1' is just a table of psi-hats for each individual. Once we have that, we can test it with a t-test or by anova of the grand mean.
> xc1
```
     [,1]
[1,]  -19
[2,]   -9
[3,]    1
[4,]  -22
[5,]  -30
[6,]  -11
```
> aov1=aov(xc1~1) # ask R to do the anova. The '~1' says to use only the intercept

> summary(aov1,intercept=T) # ask for the anova with intercept
```
            Df Sum Sq Mean Sq F value  Pr(>F)
(Intercept)  1 1350.0  1350.0  11.288 0.02011 *
Residuals    5  598.0   119.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Does this agree with the t-test we did above? F = t-squared, or t = sqrt(F). Here's a hand calculation to verify:
> sqrt(11.288)
```
[1] 3.359762
```

2. A more complex contrast.
> contr4=c(.5,.5,-1) # test average of A1 & A2 vs A3. First make contrast vector
> contr4
```
[1]  0.5  0.5 -1.0
```
> xc4 = cbind(A1,A2,A3)%*% contr4 # multiply contrast vector by a matrix made up of A1,A2,A3. Put the results in a vector called 'xc4'.
> xc4 # here is the vector of psi-hats
```
      [,1]
[1,] -19.5
[2,]  -6.5
[3,] -29.5
[4,]  -7.0
[5,] -14.0
[6,] -13.5
```

> summary (aov(xc4 ~ 1), intercept=T) # do the anova & get summary table in one step

```
            Df Sum Sq Mean Sq F value  Pr(>F)
(Intercept)  1   1350    1350  18.243 0.00793 **
Residuals    5    370      74
```

Show the mean of the psi-hats. We put the psi-hats are in the vector called 'xc4'.

> mean(xc4) # 'mean' is a built-in function

```
[1] -15
```

> sd(xc4) # sd is also built-in

```
[1] 8.602325
```

Square this sd and it equals MS-residual in the test of the intercept above.

Can also do a t-test on the vector of psi-hats:

> t.test(xc4)

```
      One Sample t-test

data:  xc4
t = -4.2712, df = 5, p-value = 0.00793
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -24.027587  -5.972413
sample estimates:
mean of x
      -15
```

III Graphs in a repeated measures design.

The functions in 'sciplot' (lineplot.CI, and barplot.CI) requires you to stack your data, and then calculate the standard errors as though it is a between-group design. We can easily build the bar plot by hand, or we can use the 'gplots' package to make a line graph with the standard error bars.

A. Line graph with error bars, using 'gplots' package.

> means=c(mean(A1),mean(A2),mean(A3)) # make a vector of means called 'means'.

> means

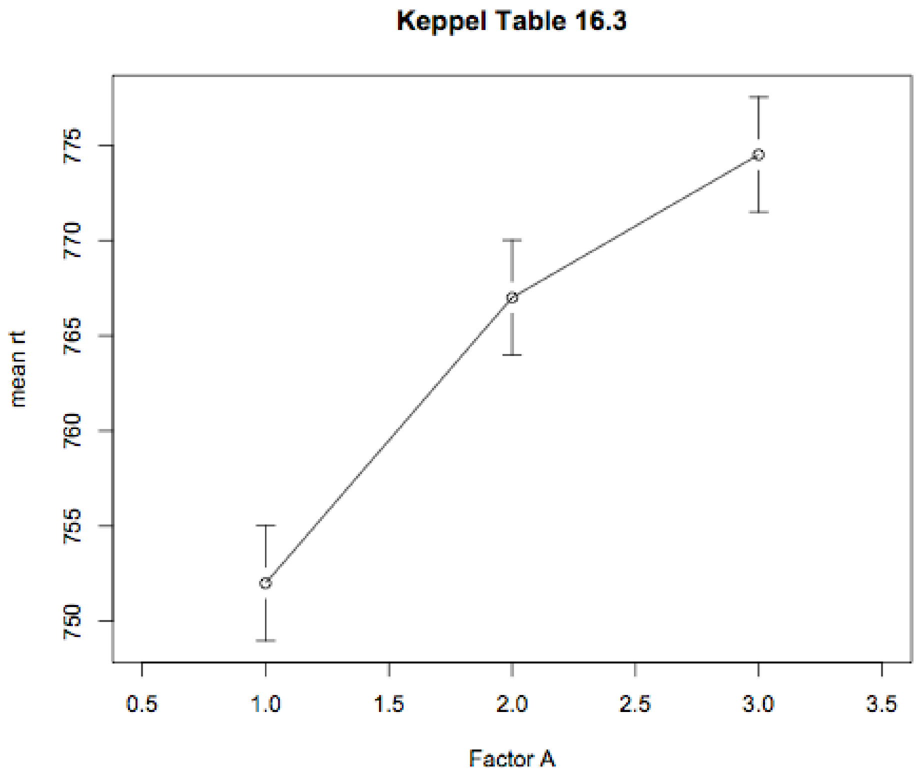
```
[1] 752.0 767.0 774.5
```

> se=sqrt(54.6 / 6); se # make a variable called 'se', and use it to calculate estimated standard error. I used the MS error from the overall anova.

```
[1] 3.016621
```

> library (gplots)

> plotCI(x = means, uiw = se, ylab="mean rt", type="l", xlab="Factor A", main="Keppel Table 16.3",xlim=c(.5,3.5)) # After looking at the first graph, I changed 'xlim' to move the bars in off the edges of the graph. I am not sure how to get "plotCI" to treat the x-axis as categorical



B. Bar plot with error bars.

We calculated the means and standard errors above. Set up the 'superpose' function (from the website of Raoul Grasman: http://users.fmg.uva.nl/rgrasman/rpages/2005/09/error-bars-in-plots.html

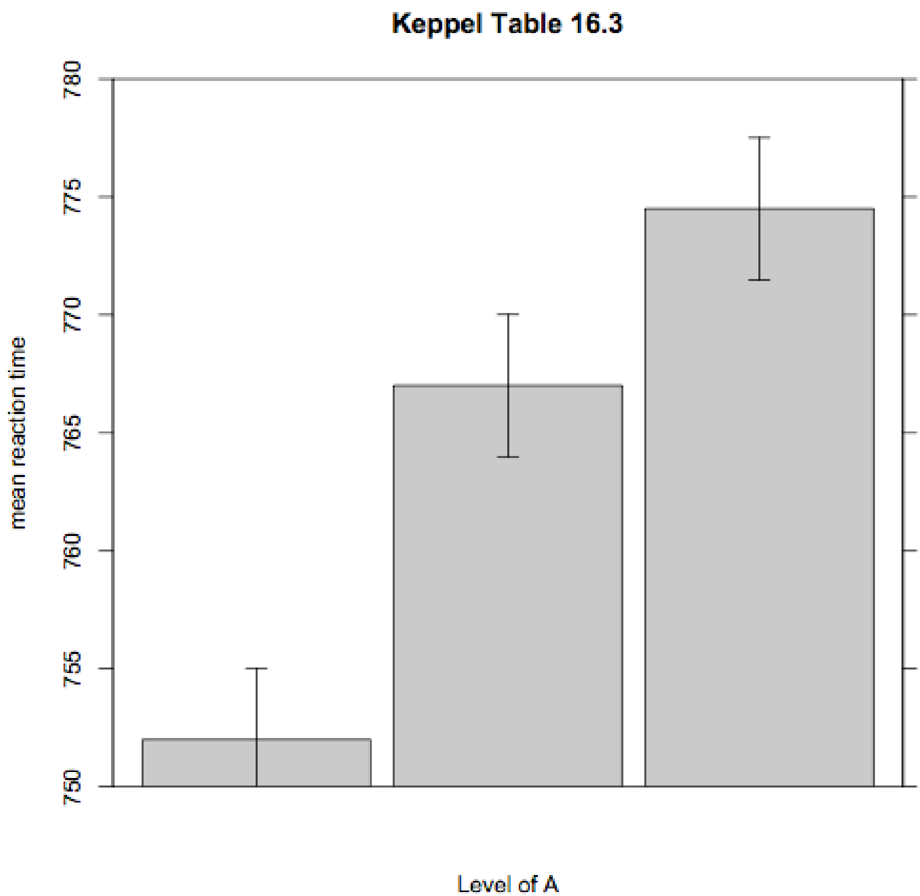
> superpose.eb = function (x, y, ebl, ebu = ebl, length = 0.08, ...) arrows(x, y + ebu, x, y - ebl, angle = 90, code = 3, length = length, ...)

> Keppelbars = barplot(means, beside=T, ylim=c(750,780), space=c(.1,.8), main="Keppel Table 16.3", xlab="Level of A", ylab="mean reaction time", legend =T, axis.lty=1, xpd=F) # 'xpd = F' tells R not to plot outside of the x and y limits you set. 'means' is our vector of data.

> superpose.eb(x=Keppelbars, y=means, ebl=c(se,se,se), col="black", lwd=1) # superpose wants a value for each error bar. For this design our estimated standard errors are model-based, so they are identical across the 3 levels of factor A. 'se' contains our estimated standard error, so we make a vector that repeats it 3 times.

> box() # add a box around the graph

> axis(4,labels=F) # add tick marks on the right-hand side


Keppel Table 16.3
mean reaction time
750
755
760
765
770
775
780
Level of A