This tutorial uses data from Winer’s (1971, 2nd edition) anova book, p. 806. The data are available on my website as ‘Winerp806.xls’.

*** Warning: Use the ‘aov’ methods here for **BALANCED** designs only. See the end of this handout and Handout 610-R12 for methods for designs that are not balanced in the between-participants part of the design.

Contents:

I. Bring in data and arrange

II. Use ‘aov’ for analysis of variance, graph results; includes putting several graphs on one page.

III. Follow-up tests (paired comparisons on cell means, interaction contrasts with pooled or partitioned error) (**Balanced Designs only**)

IV. Use ‘Anova’ in ‘car’ package for same analyses, and also illustrate unbalanced mixed design.

---

**Quick Look Summary of R Code for mixed designs:**

Using ‘aov’ for **BALANCED** designs only:

```r
> model=aov(dv~Betw*Within+Error(S / Within))
```

Using ‘Anova’ in ‘car’ package (unbalanced in between-groups part of the design is ok):

```r
> multmodel=lm(cbind(dv1,dv2,dv3)~G) # G is a grouping factor
> model1=Anova(multmodel, idata=your.factors, idesign=A*B,type="III") # make a dataframe that lays out the order of the factors and use that as ‘idata’, A and B are repeated measures factors
> summary(model1, multivariate=F)
```

---

**I. Bring in data and arrange** for R to do analysis of variance (least squares method).

-- Notice that the data indicate the grouping by numbering the subjects sequentially from 1 to 9, rather than from 1 to 3 within each Group.

-- Notice that each observation is on a separate line.

-- Make sure you make ‘subjects’ into a factor. If you don’t, R will treat it as a numerical variable and you will get nonsense.

```r
> your.data=read.table(pipe("pbpaste"),header=T) # use this to copy from the clipboard into R, OR use the method on the next line (but not both).
```
> your.data = read.table("r1data.txt", header=T)  # first set the correct folder in R under either ‘file’ or ‘Misc’ (see 610—R1), Then name the file you want to read in this statement by replacing “r1data.txt” with your data file name.
> attach(your.data)  # attaching data is convenient but can cause some problems. The alternative is to tell R what variables to use with the ‘file$variable’ method.
> your.data  # we will use ‘temprtr’ as the DV. This example has blood pressure as a covariate in the original analysis in Winer.

<table>
<thead>
<tr>
<th>group</th>
<th>subject</th>
<th>treat</th>
<th>bloodprs</th>
<th>temprtr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>6</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>8</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>

> G=factor(group)  # make factors
> T=factor(treat)
> S=factor(subject)  # make subjects (the variable indicating observation units) into a factor !!!

II. The Anova of the data, make graphs etc.

A. Carry out anova using ‘aov’
> model2=aov(temprtr~G*T+Error(S/T))  # G*T indicates we want the full model with interactions. The question is how to tell R the right error term to use. Because we want SxT as the error for T and GxT, we say that T is nested in S. Right, T is nested in subjects.

> summary(model2)

Error: S  # this is the ‘Between’ part of the design
   Df Sum Sq Mean Sq F value Pr(>F)
G      2 100.00  50.00 1.6949 0.2609
Residuals 6 177.00  29.50

Error: S:T  ## this is the ‘Within’ part of the design
   Df Sum Sq Mean Sq F value Pr(>F)
   T    1 68.056  68.056 31.4103 0.001375 **
   G:T  2 16.444  8.222  3.7949 0.086064 .
Residuals 6 13.000  2.167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
**Check the dfs** to make sure that R has the right error terms for the model you intend to run!! There are lots of ways to run this model incorrectly. Here’s my mental scratchwork on the dfs:

1--Total observations = 18 in this example. Sum of df (without the intercept) should be 17, and it is.

2--Df for error for G (the ‘between’ part of the design) should be (in Keppel’s notation) S/G, or #cells(n-1). We have 3 levels of G (#cells in the between part of the design), and we have 3 people per cell, so df = 6 is correct.

3--For the “Within” part of the design, the error df should be (in Keppel’s notation) T x S/G, or (t –1) (n – 1) (#cells). Because we have two levels of factor T, then this df should also = 6. Dfs are as we want.

**Check that the data are balanced!!**

Copy just the data and factors to a new data frame, then test that data frame for balance (this is because I made new variables for the factors, rather than just turning ‘group’ and ‘treat’ and ‘subject’ into factors themselves).

```R
> same.data=data.frame(G,T,S,temprtr)
> !is.list(replications(temprtr~G*T, data=same.data)) # omit the error part of the model in asking R about balance.
[1] TRUE
```

B. Obtain means, se’s and estimated effects, make some graphs

1. Means, se’s, estimated effects

```R
> model.tables(model2,se=T)
Tables of effects

G
  G
  1  2  3
  1.667 -3.333 1.667

T
  T
  1  2
-1.9444 1.9444

G:T
  T
  G  1  2
  1 -1.2222 1.2222
  2 1.1111 -1.1111
  3 0.1111 -0.1111

Standard errors of effects

G  T  G:T
  2.2174 0.4907 0.8498

replic. 6 9 3

Warning messages:
1: In if (se) if (type == "effects" && is.list(n)) { :
```
the condition has length > 1 and only the first element will be used
2: In if (se) result$se <- se.tables :
the condition has length > 1 and only the first element will be used

R’s estimated se’s are calculated as follows:
-- for G, \( \sqrt{\text{MS between residual} / n} \), n=6, i.e., 6 observations, 2 per individual entered
the G main effect means
-- for T, \( \sqrt{\text{MS within residual} / 9} \)
-- for GxT , \( \sqrt{\text{MS within residual} / 3} \)

\[
\text{model.tables(model2, "means")}
\]
# R won’t give estimated se’s and means both from the same command, so I used the previous statement to obtain the estimated se’s. Here are the means.

**Tables of means**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Graphs**

a) **Main effects and interaction graphs**

\[
\text{par(mfrow=c(2, 2), cex=0.6, mar=c(4, 4, 4, 2), mex=0.8)}
\]
# sets up for 4 graphs on a page. The next 4 statements ask for main effect means, then the GxT interaction.

\[
\text{plot(temprtr~G,main="Winer p 806 -- Mixed Design", xlab="Group", ylab="temperature")}
\]

\[
\text{plot(temprtr~T,main="Winer p 806 -- Mixed Design", xlab="Treatment", ylab="temperature")}
\]

\[
\text{plot(temprtr~S,main="Winer p 806 -- Mixed Design", xlab="Subjects", ylab="temperature")}
\]

\[
\text{interaction.plot(G,T,temprtr,main="Winer p 806", ylab="temperature")}
\]
b) Find predicted values, residuals, and examine assumptions and model fit.

```r
> lmmodel = lm(temprtr~G*T+S)  # fit full model with `lm`. ‘aov’ with a mixed design does not calculate predictions.
> summary(lmmodel) # ignore the sig tests below because they are meaningless.
```

```
Call:
  lm(formula = temprtr ~ G * T + S)

Residuals:
   Min      1Q  Median      3Q     Max
-1.833e+00 -1.667e-01 -4.163e-17  1.667e-01  1.833e+00

Coefficients: (2 not defined because of singularities)
   Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.8333     1.2019   6.518  0.000622 ***
```

---

Call:  
  lm(formula = temprtr ~ G * T + S)

Residuals:  
   Min      1Q  Median      3Q     Max
-1.833e+00 -1.667e-01 -4.163e-17  1.667e-01  1.833e+00

Coefficients: (2 not defined because of singularities)
   Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.8333     1.2019   6.518  0.000622 ***
> anova(lmmodel) # the anova of the ‘lm’ fit uses the INCORRECT error
Analysis of Variance Table, so ignore the p values. Use the ‘aov’ p-values above.
Remember ‘anova’ makes an analysis of variance table for a model, it does NOT carry
out an ‘analysis of variance’ of a set of data from scratch.

Response: temprtr
        Df  Sum Sq Mean Sq F value   Pr(>F)
G          2   100.00  50.000 23.0769 0.001523 **
T          1   68.066  68.066 31.4103 0.001375 **
S          6  177.000  29.500 13.6154 0.002883 **
G:T        2   16.444  8.222  3.7949 0.086064 .
Residuals 6  13.000  2.167

> resid = residuals(lmmodel) # store residuals in ‘resid’

Now graph:
> par(mfrow = c(2, 2), cex=0.6, mar=c(4, 4, 4, 2), mex=0.8) #set up for 4 graphs on a
page
> qqnorm(temprtr,main="QQ Winer p 806"); qqline(temprtr) # make the qq-normal plot
with the qq-normal line to eyeball normal distribution
> plot(fitted(lmmodel),jitter(residuals(lmmodel)),main="Winer p806 mixed");
abline(h=0) # plot fitted vs residuals, jittered so all data points will show
> plot(resid~G,xlab="Group",ylab="standardized residuals",main="Winer p806 mixed")
# plot residuals by factor G
> plot(resid~T,xlab="Treatment",ylab="standardized residuals",main="Winer p806
mixed") # plot residuals by factor T
Notice that the large residuals are all in the Group condition 3. If these were real data, you should examine those residuals and make sure they are not data errors.

III. Followup tests, and planned comparisons.

Remember, you can adjust the p-value of anything by the Holm method, or the Bonferroni method. Holm is slightly more powerful.

A. Tests on the cell means. One way is to use the estimated se for the cell means, do a pairwise test, and adjust p by the Holm method for the number of tests you are doing. This uses a pooled error. Normally we want to partition error.

Example: suppose we want to test 4 pairs of cell means that are ‘between groups’, that is, that use means from different levels of G.
Step 1—First calculate the estimated se of the difference between means.

> sediff = sqrt(2.167*(2/3)); sediff  # we have 3 obs per cell. (1/n1)+(1/n2)=2/3. 2.167 is the MSerror for testing GxT in the original anova.

[1] 1.201943

Step 2—Now use that estimated se of the diff to test any pairwise cell means that differ across Groups. These are UNADJUSTED-p t-tests.

> calct = (18 - 9)/sediff; calct  # 18 and 9 are the group means of interest

[1] 7.487877

> p1 = 2*pt(abs(calct),6,lower.tail=F,log.p=False);p1  # look up the table t, and double it to make it a two-tailed test.

[1] 0.0002931588

> calct2 = (18-11.667)/sediff; calct2

[1] 5.268969

> calct3 = (18-13)/sediff; calct3

[1] 4.159931

> p2 = 2*pt(abs(calct2),6,lower.tail=F,log.p=False);p2

[1] 0.001885082

> p3 = 2*pt(abs(calct3),6,lower.tail=F,log.p=False);p3

[1] 0.005944854

Now adjust the p-values for the # of pairwise tests you are doing all together on your study. Make a vector of the p-values and send it to the function ‘p.adjust’.

> pvec=c(p1,p2,p3)  # put the p values into a vector called ‘pvec’

> pvec

[1] 0.0002931588 0.0018850815 0.0059448542

> p.adjust(pvec,method="holm",n=9)  # assume we are doing 9 tests in all. If you are doing fewer than 9, then use a different value for the ‘n’ parameter in the p.adjust function

[1] 0.002638429 0.015080652 0.041613979  # all 3 tests are sig by Holm adjustment, assuming 9 total tests

B. Interaction contrast. With one-between and one-within factor an interaction contrast will necessarily involve the within factor. We can use either the error term for the ‘Within’ part of the design (‘pooled error’), or we can partition error (recommended by Keppel and other authors to eliminate the sphericity assumption for that particular test).

1) Pooled error method (not recommended). Apply contrast coeff’s to cell means, then use error from original anova.

Steps: a) construct contrast coefficients, b) apply coeff’s to cell means to find psi-hat, c) test MS-psi against error for the interaction from the overall anova.

Test G-linear x T. Here are the contrast coeff’s.

\[
\begin{array}{ccc}
1 & 0 & -1 \\
-1 & 0 & 1
\end{array}
\]
```r
> meanvec=c(11.667,9,13,18,10.667,16.667)  # make a vector of cell means
> coeffs=c(1,0,-1,-1,0,1)               # put the coeffs into a vector in same arrangement as cell means
> psihat=sum(meanvec*coeffs)         # multiply means by coefficients to calculate psi-hat
> psihat
[1] -2.666
> sspsi=3*(psihat^2)/4; sspsi       # calc SS-psi. n=3 for each cell mean. Sum of coeff’s squared = 4.
[1] 5.330667
> Fpsi=sspsi/2.167; Fpsi            # denominator of F is MS error from the ‘within’ part of the original anova. Remember this is pooled error, and normally we want to partition error. Because there are only 2 levels of the within factor, we actually don’t need to partition in this case.
[1] 2.459929
> pf(Fpsi, 1, 6, lower.tail = False, log.p = FALSE) # look up the prob of calc F.
[1] 0.1678321 # result is nonsig.
```

2) **Partitioned error method.**

*Steps:* a) convert data to wide format (unless you have the data arranged that way already, b) apply coeffs to individual data, find a psi-hat for each individual, c) analyze the psi-hats by a one-way between groups anova. The test of the interaction contrast is the test of the grand mean, or intercept. The tough part here is rearranging the data.

```r
> short.data=reshape(your.data,direction="wide",v.names = c("bloodprs","temprtr"), idvar="subject",timevar="treat",ids="subject" ) # this converts the data to the wide format. You can also use the data in this format to do a simple effect tests
> short.data
        group subject   bloodprs.1   temprtr.1   bloodprs.2   temprtr.2
1         1      1          3          8          4          14
3         1      2          5          11         9          18
5         1      3          11         16         14          22
7         2      4          2          6          1          8
9         2      5          8          12         9          14
11        2      6          10         9          9          10
13        3      7          7          10         4          10
15        3      8          8          14         10          18
17        3      9          9          15         12          22
```

Then use the method for calculating individual psi-hats that is shown in the One-way and Two-way within handouts. Because we have just two levels of the repeated-measures variable here, we don’t have to worry about sphericity anyway, so we don’t have to worry about partitioning error.

**IV. Use ‘Anova’ in ‘car’ package.**

The advantage of using the ‘car’ package is that we can also analyze unbalanced designs, if the lack of balance is in the between-groups part of the design. If your design is unbalanced, make sure you set the ‘options’ statement below. See section C for an unbalanced example.
A. Bring the data into R in the more natural short format. I have it in the spreadsheet both ways.

```r
> your.data=read.table(pipe("pbpaste"),header=T)
> your.data
    group partic treat temp1 temp2
1      1      1      1     8    14
2      1      2      1    11    18
3      1      3      1   16    22
4      2      4      1     6     8
5      2      5      1    12    14
6      2      6      1    10    10
7      3      7      1    10    10
8      3      8      1   14    18
9      3      9      1   15    22
```

```r
> attach(your.data) # warning: attaching data can create some problems
> G=factor(group) # make factor for grouping variable
> options(contrasts=c("contr.sum","contr.poly")) # set options for contrasts
> library(car) # activate the package for this session
> Trials=factor(c("A1","A2"), ordered=F) # make the repeated measures factor
```

B. Carry out the analysis.

First, construct a multivariate model using ‘lm’.

Second, use ‘Anova’ (capital A) to construct the analysis of the whole model.

```r
> multmod=lm(cbind(temp1,temp2)~G) # here we name ‘G’ the grouping factor as the only predictor

> model2 = Anova(multmod,idata=data.frame(Trials),idesign=~Trials,type="III")
> summary(model2,multivariate=F)
```

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

                            SS  num   Df Error SS  den Df     F      Pr(>F)
(Intercept)          3120.50     1  177.00     6 105.7797  4.934e-05 ***
G                   100.00      2  177.00     6   1.6949  0.260904
Trials              68.06      1   13.00     6  31.4103  0.001375 **
G:Trials           16.44      2   13.00     6   3.7949  0.086064 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

C. Use ‘Anova’ for an unbalanced design. I delete one person’s data to create an unbalanced design, and reanalyze to illustrate the use of ‘Anova’ for unbalanced designs.

```r
> new.data=your.data[1:8,1:5] # I create a copy of the original data, but without the last line
> new.data
    group partic treat temp1 temp2
1      1      1      1     8    14
2      1      2      1    11    18
3      1      3      1   16    22
4      2      4      1     6     8
5      2      5      1   12    14
```
> detach(your.data)  # detach the original data
> rm(your.data)      # remove original data from R’s environment
> rm(G)              # remove the original grouping factor, which has one too many values for the truncated data set.
> G=factor(new.data$group)  # re-make the grouping factor from the new data
> G
[1] 1 1 1 2 2 2 3 3
Levels: 1 2 3

> multmod2=lm(cbind(new.data$temp1,new.data$temp2)~G)  # create the multivariate model with the grouping factor as the predictor
> model3=Anova(multmod2,idata=data.frame(Trials),idesign=~Trials,type="III")
> summary(model3,multivariate=F)  # here are the Type III SS below.

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

<table>
<thead>
<tr>
<th></th>
<th>SS</th>
<th>num Df</th>
<th>Error SS</th>
<th>den Df</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>2432.19</td>
<td>1</td>
<td>136.67</td>
<td>5</td>
<td>88.9826</td>
<td>0.000226 ***</td>
</tr>
<tr>
<td>G</td>
<td>76.33</td>
<td>2</td>
<td>136.67</td>
<td>5</td>
<td>1.3963</td>
<td>0.329767</td>
</tr>
<tr>
<td>Trials</td>
<td>42.86</td>
<td>1</td>
<td>4.67</td>
<td>5</td>
<td>45.9184</td>
<td>0.001064 **</td>
</tr>
<tr>
<td>G:Trials</td>
<td>19.33</td>
<td>2</td>
<td>4.67</td>
<td>5</td>
<td>10.3571</td>
<td>0.016672 *</td>
</tr>
</tbody>
</table>

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

Let’s compare Type II and Type III SS solutions for these data:
> model4=Anova(multmod2,idata=data.frame(Trials),idesign=~Trials,type="II")
> summary(model4,multivariate=F)  # here are the Type II SS

Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

<table>
<thead>
<tr>
<th></th>
<th>SS</th>
<th>num Df</th>
<th>Error SS</th>
<th>den Df</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>76.33</td>
<td>2</td>
<td>136.67</td>
<td>5</td>
<td>1.3963</td>
<td>0.329767</td>
</tr>
<tr>
<td>Trials</td>
<td>49.000</td>
<td>1</td>
<td>4.667</td>
<td>5</td>
<td>52.5000</td>
<td>0.0007818 ***</td>
</tr>
<tr>
<td>G:Trials</td>
<td>19.33</td>
<td>2</td>
<td>4.667</td>
<td>5</td>
<td>10.3571</td>
<td>0.0166720 *</td>
</tr>
</tbody>
</table>

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

Note that Type III and Type II SS do not match for the Trials variable in this example.

D. Compare ‘aov’ to ‘Anova’ Type III SS for unbalanced design

To rerun the unbalanced version of this example in ‘aov’ we enter it in the ‘long’ format with just one observation per line. I did this by re-pasting the long version from the excel sheet into R, omitting the last two lines in order to omit one participant.
> rm(G)  # remove the grouping factor because it is set up for the ’short’ version of the data
> omitone.data=read.table(pipe("pbpaste"),header=T)
> omitone.data
   group partic treat bloodprs temptrt
1  1   1   1   3   8
2  1   1   2   4  14
3  1   2   1   5  11
> G=factor(omitone.data$group)  # re-create the factors
> T=factor(omitone.data$treat)
> P=factor(omitone.data$partic)
> model5=aov(omitone.data$temprtr~G*T+Error(P/T),data=omitone.data)
> summary(model5)  # 'aov' gives Type I SS

Error: P
   Df Sum Sq Mean Sq F value Pr(>F)
G   2 76.333  38.167  1.3963 0.3298
Residuals 5 136.667  27.333

Error: P:T
   Df Sum Sq Mean Sq F value    Pr(>F)
T   1 49.000  49.000 52.5000 0.0007818 ***
G:T 2 19.333   9.667 10.3570 0.0166720 *
Residuals 5  4.667   0.933
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1