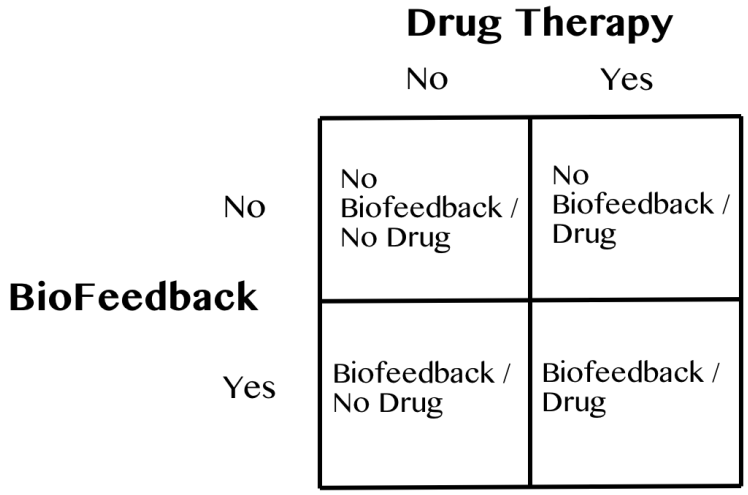
Lab Tutorial 6: Factorial Analysis of Variance (ANOVA): Between-Subjects Designs

Chris M. Fiacconi

To date, we have discussed how to analyze research designs that contain one indpendent variable (IV). When this IV has two levels, a *t*-test is usually appropriate. When three or more levels of an IV are present, ANOVAs are commonly used. In this lab, we will examine how to analyze data from research designs in which two (or more) IVs are present, and in which both variables are manipulated between-subjects. As you'll see, many of the principles from one-way ANOVAs still apply, but are expanded upon in situations where more than one IV is present.

The 2 X 2 Factorial Design

Consider the following experiment: A researcher wants to compare the effectiveness of both Drug Therapy and BioFeedback on systolic blood pressure (SBP). To do so, participants are assigned randomly to four different groups: Biofeedback + Drug (1), BioFeedback only (2), Drug only (3), or no treatment (4; neither BioFeedback nor Drug Therapy). This design can be illustrated in Figure 1 below:



Main Effects

Although you could in principle conduct a one-way omnibus ANOVA on the SBP scores across groups, the researcher instead wants to conduct the following planned linear contrasts:

$$\psi_1 = \frac{1}{2}(\mu_1 + \mu_2) - \frac{1}{2}(\mu_3 + \mu_4)$$

$$\psi_2 = \frac{1}{2}(\mu_1 + \mu_3) - \frac{1}{2}(\mu_2 + \mu_4)$$

Therefore, the weights for the first contrast are (0.5, 0.5, -0.5, -0.5), and the weights for the second contrast are (0.5, -0.5, 0.5, -0.5). **You should note two things here:**

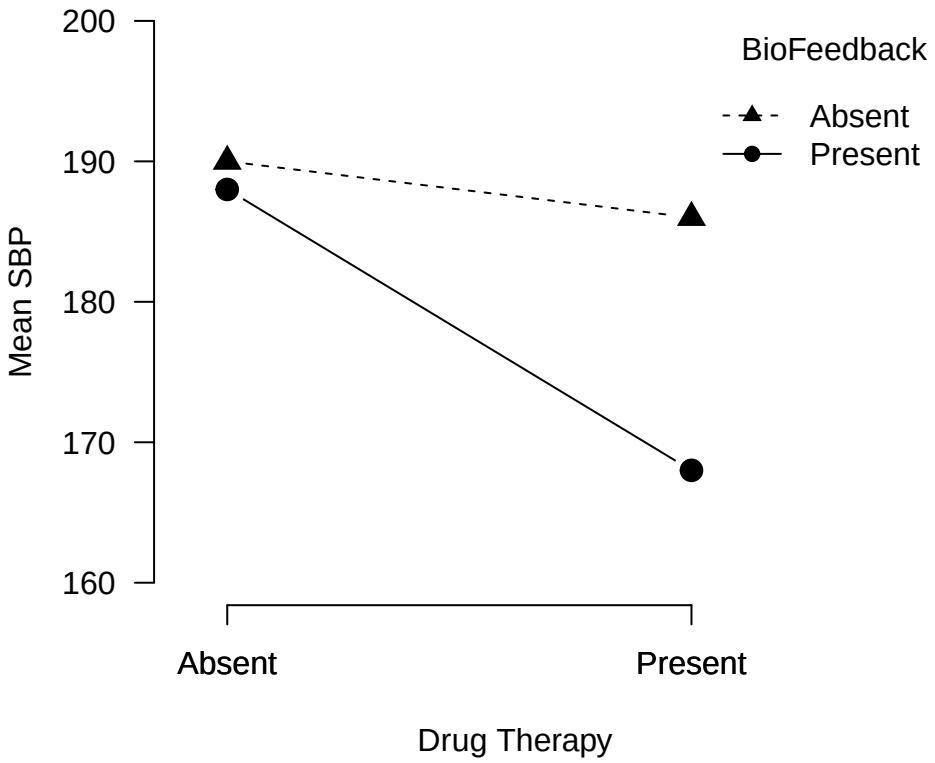


Figure 1: Plot depicting data from MDK Chapter 7 Table 1.

1) First, note that these two contrasts are orthogonal:

$$\sum_{j=1}^{a} c_{1j}c_{2j} = 0$$

2) The first contrast compares the *average* of the groups where BioFeedback is present vs. the *average* of the groups where BioFeedback is absent, whereas the second contrast compares the *average* of the groups where Drug Therapy is present vs. the *average* of the groups where Drug Therapy is absent. In other words, the first contrast examines the effect BioFeedback *irrespective* of whether Drug Therapy is present or absent, and the second contrast examines the effect of Drug Therapy *irrespective* of whether BioFeedback is present or absent. Therefore, each contrast tests the *main effect* of BioFeedback and Drug Therapy, respectively. Let's take a look at how these main effects look visually in Figure 2:

```
# Load in data
bp.data<-read.table(file="Ch7T1.txt",header=T)
```

```
# Make figure
with(bp.data,interaction.plot(x.factor=Drug,
                              trace.factor=BioFeedback,response=Score,las=1,
                              ylab="Mean SBP",xlab="Drug Therapy",ylim=c(160,200),
                              bty="n",type="b",cex=1.5,pch=c(17,19))) # make plot, remove outer box

axis(side=1,at=c(1,2),labels=c("Absent","Present")) # add in x-axis with ticks and labels
```

How many participants are in each condition?

```r
# Load tidyverse
library(tidyverse)

bp.n<-bp.data %>% group_by(Drug,BioFeedback) %>% summarize(n=length(Score)) # Get n per group
print(bp.n)
```

```
## # A tibble: 4 x 3
## # Groups:   Drug [?]
##   Drug    BioFeedback     n
##   <fct>   <fct>       <int>
## 1 Absent  Absent          5
## 2 Absent  Present         5
## 3 Present Absent          5
## 4 Present Present         5
```

Let's get the means and SDs for each condition.

```r
# Get mean and standard deviation
bp.descriptives<-bp.data %>% group_by(Drug,BioFeedback) %>%
  summarize(Mean=mean(Score),S.Dev=sd(Score))

print(bp.descriptives)
```

```
## # A tibble: 4 x 4
## # Groups:   Drug [?]
##   Drug    BioFeedback  Mean S.Dev
##   <fct>   <fct>       <dbl> <dbl>
## 1 Absent  Absent        190  7.91
## 2 Absent  Present       188  7.91
## 3 Present Absent        186  7.91
## 4 Present Present       168  7.91
```

From Figure 2 we can see that the circle points (BioFeedback present) are generally higher than the triangle points (BioFeedback absent), suggesting that overall, BioFeedback lowers SBP. Likewise, the points on the right of the graph are generally lower than the points on the left of the graph, suggesting that overall, Drug Therapy also lowers SBP.

Testing Main Effects: Model Comparison

Although it is not incorrect to think of this design as involving four separate conditions, it is customary to treat this design as resulting from the *crossing* of two different IVs (BioFeedback & Drug Therapy). In this example, each IV has two levels, with each level being paired with each level of the other IV to form the four different conditions/groups.

As was the case when we considered one-way designs with a single IV, we can also apply the model comparison perspective to understand how a factorial design can be accomodated within the framework of the General Linear Model. In the general case of two indpendent variables with Factor A having a levels, and Factor B having b levels, the full model can be written as:

$$Y_{ijk} = \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

where μ again represents the grand mean, α_j represents the effect of being in the jth level of Factor A, β_k represents the effect of being in the kth level of Factor B, $(\alpha\beta)_{jk}$ represents the *interaction* effect of being

in level j of Factor A and level k of Factor B in combination with one another. We'll discuss the interaction term in greater detail later on.

To test for the main effect of Factor A, we would need to compare the relative fit of the full model with a restricted model that is identical to the full model except for the absence of the α_j parameters:

$$Y_{ijk} = \mu + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

The null hypothesis that is being tested in this comparison is that all μ_j's are equal, or equivalently, that all $\alpha_j = 0$.

Similarly, to test for the main effect of Factor B, we would need to compare the relative fit of the full model with a restricted model that is identical to the full model except for the absence of the β_k parameters:

$$Y_{ijk} = \mu + \alpha_j + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

The null hypothesis that is being tested in this comparison is that all μ_k's are equal, or equivalently, that all $\beta_k = 0$.

The relative fit of the full and restricted models is again computed by examining the difference in the residuals between the models, and can be expressed as an F-statistic:

$$F = \frac{(E_R - E_F)/(df_R - df_F)}{E_F/df_F}$$

You can calculate the *df* for the numerator and the denominator of the F-statistic when testing the main effect of Factor A as follows:

$$df_R - df_F = a - 1$$
$$df_F = ab(n-1)$$

You can calculate the *df* for the numerator and the denominator of the F-statistic when testing the main effect of Factor B as follows:

$$df_R - df_F = b - 1$$
$$df_F = ab(n-1)$$

Once again, all α_j and β_k parameters are constrained to sum to zero:

$$\sum_{j=1}^{a} \alpha_j = 0$$

$$\sum_{k=1}^{b} \beta_k = 0$$

Each parameter is defined relative to the grand mean as before:

$$\alpha_j = \bar{Y}_{j.} - \mu$$
$$\beta_k = \bar{Y}_{.k} - \mu$$

where $\bar{Y}_{j.}$ is the *marginal mean* of the jth level of Factor A calculated as:

$$\bar{Y}_{j.} = \sum_{k=1}^{b} \bar{Y}_{jk}/b$$

and $\bar{Y}_{.k}$ is the *marginal mean* of the kth level of Factor B, calculated as:

$$\bar{Y}_{.k} = \sum_{j=1}^{a} \bar{Y}_{jk}/a$$

The α_j and β_k parameters can also be used to calculate the sums-of-squares associated with each main effect. For an equal-n design:

$$SS_A = nb\sum_{j=1}^{a}\alpha_j^2$$

$$SS_B = na\sum_{k=1}^{b}\beta_k^2$$

Alternatively, both SS_A and SS_B can be calculated from their respective marginal means along with the grand mean:

$$SS_A = nb\sum_{j=1}^{a}(\bar{Y}_{j\cdot} - \mu)^2$$

$$SS_B = na\sum_{k=1}^{b}(\bar{Y}_{\cdot k} - \mu)^2$$

where n is the number of observations within each cell.

The null hypothesis for Factor A is that all $\alpha_j = 0$, and the null hypothesis for Factor B is that all $\beta_k = 0$.

Finally, the sum-of-squares associated with the residuals, or SS_{Within}, is calculated as follows:

$$SS_{Within} = E_F = \sum_{j=1}^{a}\sum_{k=1}^{b}\sum_{i=1}^{n}(Y_{ijk} - \bar{Y}_{jk})^2$$

Essentially, SS_{Within} represents the sum across individuals and groups of how far an individual's score deviates from their corresponding cell mean, $\bar{Y}_{jk}$, and can be thought of as a measure of prediction error.

Interactions

Recall that in a study with $a = 4$ groups, there are $a - 1$ possible orthogonal contrasts. So far, we have discussed only two of the three. The third orthogonal contrast is of the form:

$$\psi_3 = \frac{1}{2}(\mu_1 - \mu_3) - \frac{1}{2}(\mu_2 - \mu_4)$$

with the weights for this contrast being (0.5, -0.5, -0.5, 0.5). We could also use the weights (-0.5, 0.5, 0.5, -0.5), but this is essentially the same contrast, just flipped.

Importantly, this contrast captures the *interaction* between Drug and BioFeedback, allowing the effect of one IV to depend on the level of the other IV. In particular, the above contrast tests whether the difference between the levels of the biofeedback variable when drug therapy is present is different than the difference between the levels of the biofeedback variable when drug therapy is absent. In other words, it tests whether the effect of biofeedback depends on whether drug therapy is present or absent. Such dependency of the effect of one IV on the level of the other is the hallmark of interaction effects, and the ability to test for an interaction is a critical advantage of factorial designs.

Referring back to Figure 2, it can be seen that indeed it appears as though the effect of biofeedback depends heavily on the presence of drug therapy. The effect of biofeedback is much greater when drug therapy is also present than when it is absent.

Testing Interactions: Model Comparisons

In terms of model comparisons, testing for the presence of an interaction is tantamount to comparing the full model

$$Y_{ijk} = \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

with the following restricted model which omits only the $(\alpha\beta)_{jk}$ interaction term:

$$Y_{ijk} = \mu + \alpha_j + \beta_k + \epsilon_{ijk}$$

This comparison again involves comparing the relative fit of the two models using the same F-statistic:

$$F = \frac{(E_R - E_F)/(df_R - df_F)}{E_F/df_F}$$

where the numerator df is equal to:

$$df_R - df_F = (a-1)(b-1)$$

and the denominator df is calculated the same way as before:

$$df_F = ab(n-1)$$

The $(\alpha\beta)_{jk}$ parameters represent the degree to which the cell mean $\bar{Y}_{jk}$ deviates from the sum of the corresponding α_j and β_k effects and are calculated as follows:

$$(\alpha\beta)_{jk} = \bar{Y}_{jk} - (\mu + \alpha_j + \beta_k)$$

Therefore, the null hypothesis tested in this comparison is that all $(\alpha\beta)_{jk} = 0$.

Similar to SS_A and SS_B, SS_{AB} can be calculated as follows:

$$SS_{AB} = n\sum_{j=1}^{a}\sum_{k=1}^{b}(\alpha\beta)_{jk}^2$$

SS_{AB} is also the difference in the residuals between the restricted and full models $(E_R - E_F)$ that capture the interaction effect shown above.

The null hypothesis for the interaction between Factor A and Factor B is that all $(\alpha\beta)_{jk} = 0$.

Note that a factorial ANOVA (with equal n groups) is really a collection of three orthogonal contrasts that evaluate each main effect and the interaction between IVs. You might wonder whether we need to introduce a correction to ensure that $\alpha_{FW} = .05$. However, such a correction is not typically done as each contrast and any subsequent comparisons are considered to be a different family of tests.

Simple Main Effects

A significant interaction usually supercedes any significant main effects because it tells us that the effect of one IV depends on the level of the other IV. However, a researcher will often want to probe further to test whether the effect of one IV is significant *within* each level of the other. Such tests are known as *simple main effects* because they examine the effect of an IV within each level of the other IV. I will show you how to conduct such tests in the following example.

Example #1 - 2 X 2 Design

Let's use the data from Figure 1 to illustrate how to analyze a 2 X 2 design using *R*. First, there are many different ways to conduct a factorial ANOVA in *R*, but we'll start by showing how each main effect and the interaction can be computed using the **lm** followed by the **anova** function:

```
options(contrasts=c("contr.sum","contr.poly"))

model.full<-lm(Score~1+Drug+BioFeedback+Drug:BioFeedback,data=bp.data)
# Equivalent to lm(Score~Drug*BioFeedback,data=bp.data)

anova(model.full) # Compare appropriate nested models for each main effect and interaction
```

```
## Analysis of Variance Table
##
## Response: Score
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## Drug              1    720   720.0   11.52 0.003706 **
## BioFeedback       1    500   500.0    8.00 0.012109 *
## Drug:BioFeedback  1    320   320.0    5.12 0.037917 *
## Residuals        16   1000    62.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA indicates that there is a significant main effect of both Drug, $F(1, 16) = 11.52, p = .004$, and BioFeedback, $F(1, 16) = 8.00, p = .012$, as well as a significant Drug:BioFeedback interaction, $F(1, 16) = 5.12, p = .038$.

You can also use the **aov** and **summary** commands as before:

```
sbp.aov<-aov(Score~Drug*BioFeedback,data=bp.data) # Do ANOVA
print(summary(sbp.aov)) # print summary of results
```

```
##                  Df Sum Sq Mean Sq F value  Pr(>F)
## Drug              1    720   720.0   11.52 0.00371 **
## BioFeedback       1    500   500.0    8.00 0.01211 *
## Drug:BioFeedback  1    320   320.0    5.12 0.03792 *
## Residuals        16   1000    62.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The *SS* for each effect in the model (2 main effects + interaction) are calculated by comparing the difference in the residuals between the full and restricted models for each effect as described above. Let's illustrate this by comparing the difference in the sum of the squared residuals from the following nested models to obtain SS_{Drug}.

```
# Examine main effect of Drug using intercept-only restricted model
drug.restricted<-lm(Score~1,data=bp.data) # Define intercept-only model
drug.full<-lm(Score~1+Drug,data=bp.data) # Define intercept + Drug model

SS.drug<-sum((residuals(drug.restricted)^2))-sum((residuals(drug.full)^2))
print(SS.drug)
```

```
## [1] 720
```

```
anova(drug.restricted,drug.full)
```

```
## Analysis of Variance Table
##
## Model 1: Score ~ 1
## Model 2: Score ~ 1 + Drug
##   Res.Df  RSS Df Sum of Sq      F  Pr(>F)
## 1     19 2540
## 2     18 1820  1       720 7.1209 0.01566 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that SS_{Drug} and the main effect of Drug can also be evaluated by comparing two nested models that control for the influence of BioFeedback. Because the procedure for evaluating each main effect essentially involves two orthogonal contrasts, SS_{Drug} will be identical in both cases. The following code illustrates this point by comparing two nested models in which BioFeedback is already entered into each model:

```
# Examine main effect of Drug after controlling for BioFeedback
drug2.restricted<-lm(Score~1+BioFeedback,data=bp.data)
drug2.full<-lm(Score~1+BioFeedback+Drug,data=bp.data)

SS.drug2<-sum((residuals(drug2.restricted)^2))-sum((residuals(drug2.full)^2))
print(SS.drug2)
```

```
## [1] 720
```

```
anova(drug2.restricted,drug2.full)
```

```
## Analysis of Variance Table
##
## Model 1: Score ~ 1 + BioFeedback
## Model 2: Score ~ 1 + BioFeedback + Drug
##   Res.Df  RSS Df Sum of Sq      F   Pr(>F)
## 1     18 2040
## 2     17 1320  1       720 9.2727 0.007315 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that $SS_{Drug} = E_R - E_F = 720$ in both comparisons, which is the same as in the output from the ANOVA. However, the F and p-values associated with the main effect of Drug differ in each comparison. Why might this be? It's because when we did the factorial ANOVA (see *sbp.aov*), the main effect of Drug was evaluated *after* accounting for the main effect of Biofeedback *and* the Drug:BioFeedback interaction. Therefore, E_F in the denominator of the F-statistic would be smaller in this case, consequently yielding a larger F-value and a smaller p-value.

Further Evaluation Of A Significant Interaction

As alluded to in the previous section, the presence of a significant interaction usually (but not always!) renders the main effects moot because it signifies that the effect of an IV is dependent upon the level of the

other IV, and therefore it does not make sense to interpret the effect of this IV while ignoring the other IV. If a significant interaction is present, a researcher will usually want to conduct simple main effect tests that compare the levels of one IV within each level of the other IV.

In our example, we could choose to evaluate the simple main effect of drug therapy both when biofeedback is present as well as when biofeedback is absent. Alternatively, we could also choose to evaluate the simple main effect of biofeedback both when drug therapy is present and when drug therapy is absent. Typically only one set of simple main effects would be tested. The simple main effects that you choose to conduct should be informed by the primary goal of the research question. If we were primarily interested in the role of drug therapy in SBP reduction, it might be most informative to examine the effectiveness of drug therapy with and without concurrent biofeedback. These tests can be carried out using the **lm** and **testInteractions** commands found in the *phia* package.

```
library(phia) # Make sure phia package is installed - install.packages("phia")

sbp.lm<-lm(Score~Drug*BioFeedback,data=bp.data) # Create lm object

testInteractions(model=sbp.lm,fixed="BioFeedback",pairwise="Drug",
                 adjustment="none") # test simple main effect of Drug Therapy
```

```
## F Test:
## P-value adjustment method: none
##                            Value Df Sum of Sq     F   Pr(>F)
## Absent-Present :  Absent     4  1        40  0.64 0.435428
## Absent-Present : Present    20  1      1000 16.00 0.001032 **
## Residuals                      16      1000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see from the output that drug therapy significantly lowers SBP when it is paired with biofeedback, but that it does not have any effect on SBP if biofeedback absent.

Which Error Term To Use for Simple Main Effect Analyses?

Why not simply do two *t*-tests - each comparing the effect of Drug within each level of BioFeedback? The answer has to do with which error terms you want to use in your analyses. If we were to conduct *t*-tests, our estimate of the within-group error variance would be based solely on the two groups being compared in the *t*-test. Most of the time, it is better to use the estimate of error variance based on all experimental groups, MS_W, as we will get a more precise estimate of the true population-level within-group variance. The **testInteractions** function does this for us, using MS_W from the factorial ANOVA as the error term for the simple main effects tests.

However, if the homogeneity of variance assumption is violated, it may instead be preferable to use an error term that is based only on the groups being compared. In this case, *t*-tests would be more appropriate. Let's evaluate the homogeneity of variance for the *bp.data* scores. First, let's create a boxplot so we can examine how the scores across all 4 groups are distributed:

```
boxplot(Score~Drug+BioFeedback,data=bp.data,las=1,xaxt="n",ylab="SBP")
axis(side=1,at=c(1,2,3,4),labels=c("DAb-BAb","DPr-BAb","DAb-BPr","DPr-BPr"),las=1)
```

Even though it appears each of our groups has equal variance, let's formally evaluate this assertion with the **bartlett.test** command:

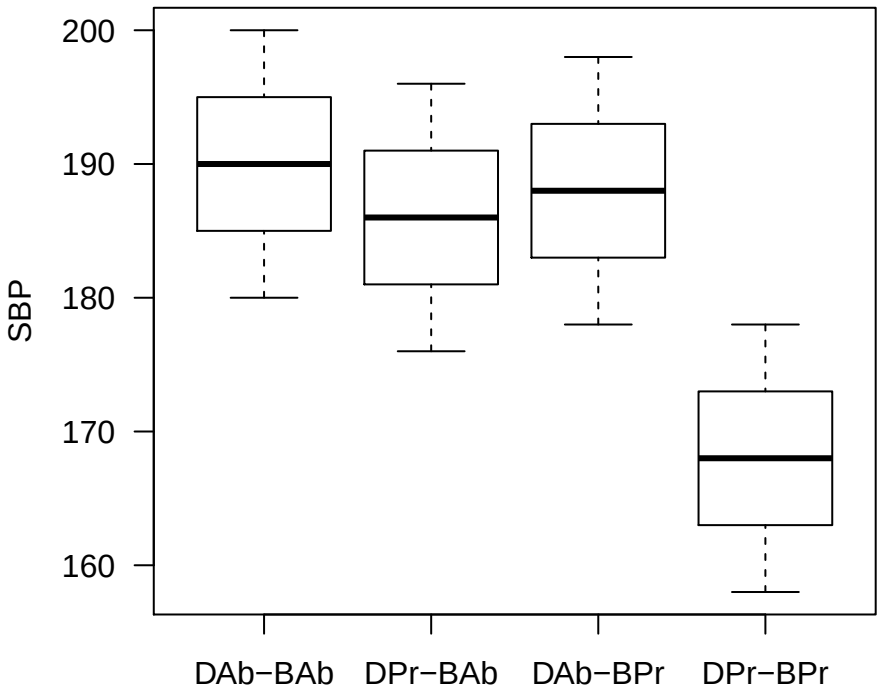


Figure 2: Boxplot of of data from MDK Chapter 7 Table 1

```
bartlett.test(Score~Group,data=bp.data) # Note that the Group variable ignores Drug/BioFeedback
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Score by Group
## Bartlett's K-squared = 0, df = 3, p-value = 1
```

We can see from this analysis that there is no evidence to suggest that the homogeneity of variance assumption is violated, so we're better off using MS_W as the error term in our simple main effect analyses.

Example #2 - Extension To IVs With More Than 2 Levels: 2 X 3 Designs

So far we have been dealing with cases where each IV has only two levels. Often, however, IVs have three or more levels. Statistical analysis of such designs are very similar to cases in which each IV has only two levels, with only a slight change in the way in which main effects and simple main effects are examined.

In the next example, researchers were interested in testing the effectivness of two different therapeutic techniques on hypertension (high blood pressure) - Drug therapy and BioFeedback. Three different drugs were compared either in combination with Biofeedback, or alone. The dependent variable measured in this study was systolic blood pressure (SBP). Therefore, we have two IVs (Drug, BioFeedback), with Drug having three levels (drugX, drugY, drugZ) and BioFeedback having two levels (Absent, Present). Let's load in the data file "Ch7T5.txt".

```r
bp.data2<-read.table(file="Ch7T5.txt",header=T) # load data from MDK Chapter 7 Table 5
str(bp.data2) # Look at the structure of our data file
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ Score      : int  170 175 165 180 160 186 194 201 215 219 ...
##  $ BioFeedback: Factor w/ 2 levels "Absent","Present": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Drug       : Factor w/ 3 levels "drugX","drugY",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Subj       : int  1 2 3 4 5 6 7 8 9 10 ...
```

How many participants are in each condition?

```r
# Get n per group
bp2.n<-bp.data2 %>% group_by(Drug,BioFeedback) %>% summarize(n=length(Score))
print(bp2.n)
```

```
## # A tibble: 6 x 3
## # Groups:   Drug [?]
##   Drug  BioFeedback     n
##   <fct> <fct>       <int>
## 1 drugX Absent          5
## 2 drugX Present         5
## 3 drugY Absent          5
## 4 drugY Present         5
## 5 drugZ Absent          5
## 6 drugZ Present         5
```

Let's get the means and SDs for each condition:

```r
bp2.descriptives<-bp.data2 %>% group_by(Drug,BioFeedback) %>%
  summarize(Mean=mean(Score),S.Dev=sd(Score))

print(bp2.descriptives)
```

```
## # A tibble: 6 x 4
## # Groups:   Drug [?]
##   Drug  BioFeedback  Mean S.Dev
##   <fct> <fct>       <dbl> <dbl>
## 1 drugX Absent        186 10.8
## 2 drugX Present       170  7.91
## 3 drugY Absent        201 10.9
## 4 drugY Present       203 13.9
## 5 drugZ Absent        210 15.8
## 6 drugZ Present       188 13.8
```

Now, let's plot the means for each condition:

```r
with(bp.data2,interaction.plot(x.factor=Drug,
                               trace.factor=BioFeedback,response=Score,las=1,
                               ylab="Mean SBP",xlab="Drug",ylim=c(160,220),
                               bty="n",type="b",cex=1.5,pch=c(17,19))) # make plot, remove outer box

axis(side=1,at=c(1,2,3),labels=c("drugX","drugY","drugZ")) # add in x-axis with ticks and labels
```

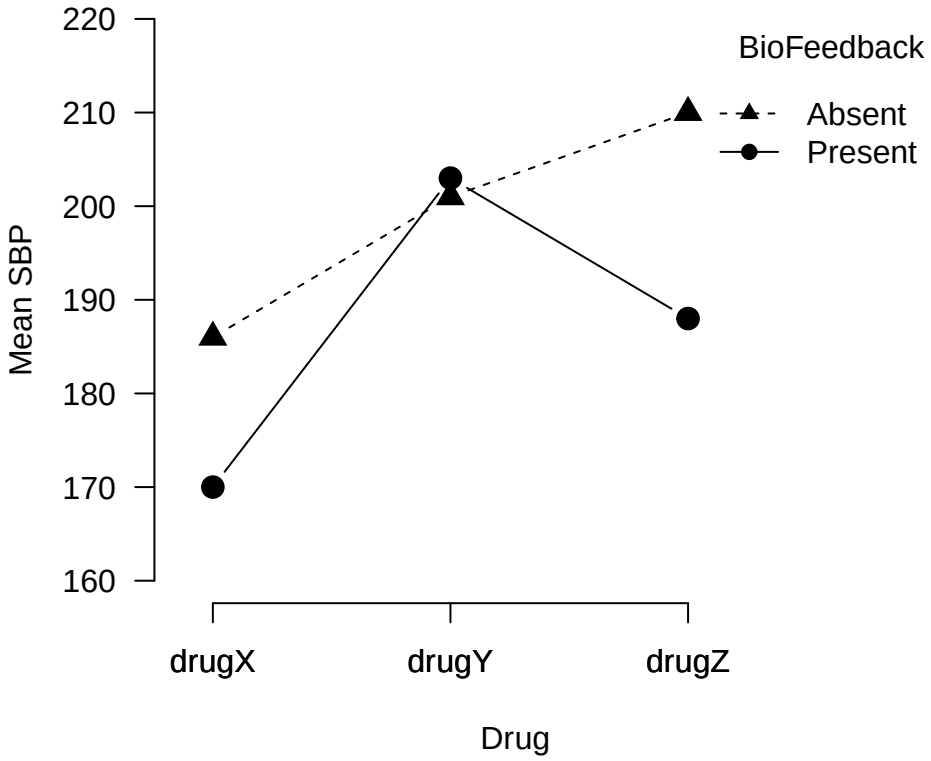


Figure 3: Plot of data from MDK Chapter 7 Table 5

From looking at the plot in Figure 3, it seems like there might be a main effect of Drug, and possibly a main effect of BioFeedback. Let's take a look at the marginal means for each IV:

```
drug.marg<-bp.data2 %>% group_by(Drug) %>% summarize(Mean=mean(Score))
print(drug.marg)
```

```
## # A tibble: 3 x 2
##   Drug   Mean
##   <fct> <dbl>
## 1 drugX   178
## 2 drugY   202
## 3 drugZ   199
```

```
fdbk.marg<-bp.data2 %>% group_by(BioFeedback) %>% summarize(Mean=mean(Score))
print(fdbk.marg)
```

```
## # A tibble: 2 x 2
##   BioFeedback  Mean
##   <fct>       <dbl>
## 1 Absent        199
## 2 Present       187
```

From these marginal means, we can see that if there is a main effect of Drug, it's likely driven by drugX which seems to produce the lowest SBP reading. Moreover, it seems as though the presence of Feedback may also help reduce SBP. Let's formally evaluate these main effects and interaction using a 2 X 3 Factorial ANOVA:

```r
bp2.aov<-aov(Score~Drug*BioFeedback,data=bp.data2) # perform 2 X 3 ANOVA
summary(bp2.aov)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## Drug               2   3420  1710.0  10.979 0.000411 ***
## BioFeedback        1   1080  1080.0   6.934 0.014563 *
## Drug:BioFeedback   2    780   390.0   2.504 0.102877
## Residuals         24   3738   155.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output from the ANOVA yields a main effect of Drug, $F(2, 24) = 10.98, p < .001$, and Feedback, $F(1, 24) = 6.93, p = .01$, but no significant interaction, $F(2, 24) = 2.50, p = .10$. Because there are only two levels of Feedback, and the main effect was significant, we can conclude that the presence of Feedback ($M = 187$) reduced SBP relative to when Feedback was absent ($M = 199$). But what about Drug? Because there are three levels, the main effect only tells us that the mean SBP readings are not equal for each drug, but doesn't specify which specific drugs have an effect.

Examining Main Effects With More Than Two Levels

As in the case of a one-way between-subjects ANOVA, we now need to specify which drugs have an effect by conducting more targeted comparisons. If only a few pairwise comparisons are of interest and they were planned before collecting the data, then Tukey's HSD procedure would not be appropriate as it is too conservative in this case. Instead, you should conduct pairwise linear contrasts using Bonferroni's adjustment (see Lab Tutorial 5) to maintain $\alpha_{FW} = .05$. Alternatively, some would argue that if you are doing a small number of such planned contrasts, you could consider proceeding without correcting for Type I Error inflation because you would only do such tests when the main effect is significant in the omnibus test.

If none of the pairwise comparisons among levels of the Drug variable were planned, and all such comparisons are of interest, we could use the **TukeyHSD** function to perform post-hoc pairwise comparisons between the marginal means for all three drugs while maintaining $\alpha_{FW} = .05$. As before, we'll provide the *bp2.aov* model object as an argument to the **TukeyHSD** function, but we'll also need to specify which IV we need to examine pairwise contrasts for with the *which* argument.

All Pairwise Comparisons (For Post-Hoc Analyses)

```r
drug.phoc<-TukeyHSD(bp2.aov,which="Drug") # Use Tukey's HSD to compare marginal means for Drug
print(drug.phoc)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Score ~ Drug * BioFeedback, data = bp.data2)
##
## $Drug
##              diff        lwr      upr     p adj
## drugY-drugX    24  10.062095 37.93791 0.0006958
## drugZ-drugX    21   7.062095 34.93791 0.0026568
## drugZ-drugY    -3 -16.937905 10.93791 0.8537283
```

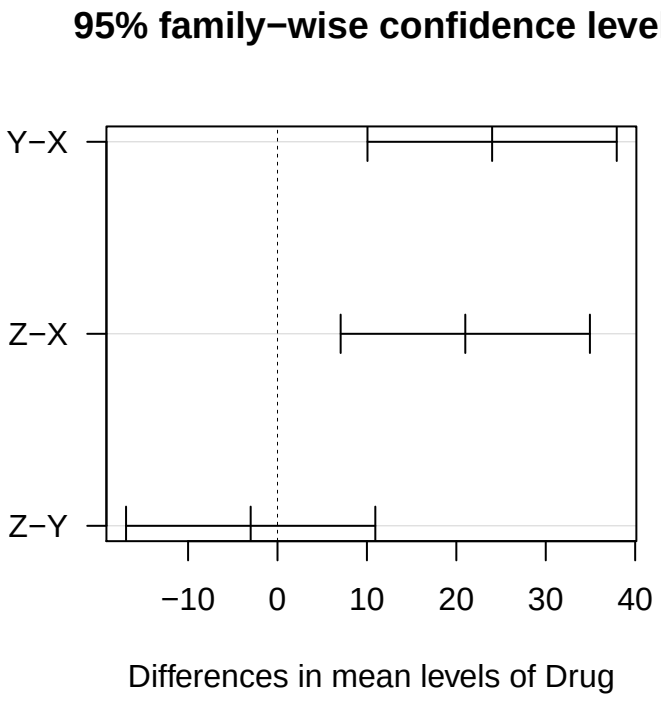


Figure 4: Simultaneous 95% CIs for pairwise comparisons of Drug marginal means

```
# Plot simultaneous 95% CIs
plot(drug.phoc,las=1,yaxt="n")
axis(side=2,at=c(3,2,1),labels=c("Y-X","Z-X","Z-Y"),las=1)
```

The post-hoc tests reveal that the difference between drugX and drugY is significant, and that the difference between drugX and drugZ is significant. There is no reliable difference between drugY and drugZ. These tests line up well with our visual inspection of the means (see plot above).

Linear Contrasts

Instead of performing all pairwise comparisons among the marginal means of Drug, we could also perform a post-hoc linear contrast in which we test whether the marginal mean for drugX is less than the average of the marginal means for drugs Y and Z. Recall the general formula for the F-statistic for a linear contrast:

$$F = \frac{\psi^2 / \sum_{j=1}^{a}(c_j^2/n_j)}{MS_W}$$

Because this is contrast is post-hoc, we'll adjust for Type I Error inflation using Scheffe's Method. If this was a planned contrast, you would not need to adjust your p-value according to the procedures introduced in Lab Tutorial 5.

```
MS.w<-155.8 # Taken from ANOVA table above
n.pergroup<-10 # Note that n = 10 here because we're testing the marginal means!!
c.weights<-c(-2,1,1) # Define contrast weights
psi<-sum(c.weights*drug.marg$Mean) # Calculate psi
F<-((psi^2)/(sum((c.weights^2)/n.pergroup)))/MS.w # Calculate observed F-statistic

F.crit<-qf(p=.95,df1=1,df2=30-3) # Get critical value of F
```

```r
F.scheffe<-(3-1)*F.crit # Multiply F.crit by (a - 1) to get F Scheffe

F > F.scheffe # Compare observed F to F.scheffe
```

```
## [1] TRUE
```

```r
p.value<-1-pf(F,df1=1,df2=30-3)
print(p.value)
```

```
## [1] 7.712441e-05
```

Note that our observed F > F Scheffe, so we can conclude that our contrast was indeed significant. You could report this finding as, $F(1, 27) = 21.66, p < .001$, where Scheffe's Method was used to correct for inflation of Type I Error and maintain $\alpha_{FW} = .05$.

Example #3 - Extension To IVs With More Than 2 Levels: 3 X 3 Designs

Statistical analysis of designs where each IV has 3 levels are similar to the cases we have discussed so far, but subsequent analysis of a significant interaction becomes a bit more complex.

In the next example, researchers were interested in comparing the performance of 3 different patient groups (Amnesic, Huntingtons, and Controls) on 3 different tasks (Recognition, Classification, and Artificial Grammar). Both Classification and Artificial Grammar are implicit memory tasks whereas Recognition is an explicit memory task. Since Amnesic patients are known to have a deficit in explicit memory and Huntingtons patients are thought to have a deficit in implicit memory, the researchers wanted to directly test this idea by examining how both patient groups (and healthy controls) performed across these three tasks. Let's load in the data file "Ch7T11.txt".

```r
patient.data<-read.table(file="Ch7T11.txt",header=T)
str(patient.data)
```

```
## 'data.frame':    45 obs. of  3 variables:
##  $ Diagnosis: Factor w/ 3 levels "Amnesic","Control",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Task     : Factor w/ 3 levels "Classification",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Y        : int  44 63 76 72 45 72 66 55 82 75 ...
```

```r
# Re-order Diagnosis conditions
patient.data$Diagnosis<-factor(patient.data$Diagnosis, levels=c("Amnesic","Huntingtons","Control"))
```

Let's plot the means of each of the 9 conditions as we've done before:

```r
with(patient.data,interaction.plot(x.factor=Diagnosis,
                                   trace.factor=Task,response=Y,las=1,
                                   ylab="Mean Score",xlab="Patient Group",
                                   bty="n",type="b",cex=1.5,pch=c(17,18,19))) # make plot, remove outer bo

axis(side=1,at=c(1,2,3),labels=c("Amnesic","Huntingtons","Control")) # add in x-axis with ticks and lab
```

How many participants are in each condition?

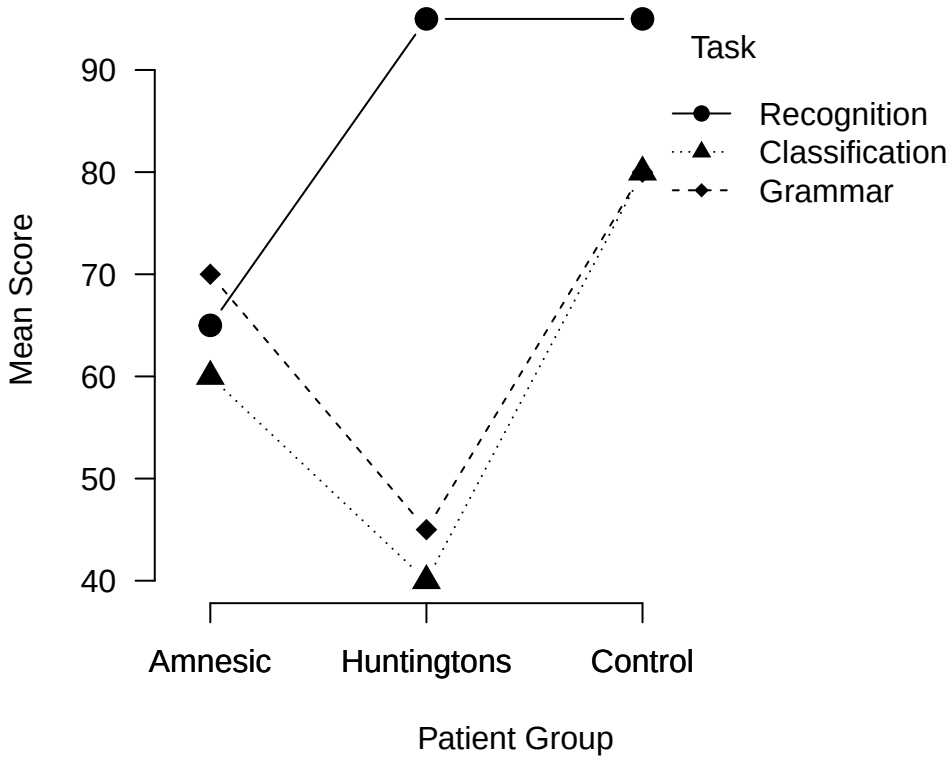


Figure 5: Plot of means of data taken from MDK Chapter 7 Table 11

```
# Get n per group
patient.n<-patient.data %>% group_by(Diagnosis,Task) %>% summarize(n=length(Y))
print(patient.n)
```

```
## # A tibble: 9 x 3
## # Groups:   Diagnosis [?]
##   Diagnosis   Task              n
##   <fct>       <fct>         <int>
## 1 Amnesic     Classification     5
## 2 Amnesic     Grammar            5
## 3 Amnesic     Recognition        5
## 4 Huntingtons Classification     5
## 5 Huntingtons Grammar            5
## 6 Huntingtons Recognition        5
## 7 Control     Classification     5
## 8 Control     Grammar            5
## 9 Control     Recognition        5
```

Let's get the means and SDs for each condition:

```
patient.descriptives<-patient.data %>% group_by(Diagnosis,Task) %>%
  summarize (Mean=mean(Y),S.Dev=sd(Y))

print(patient.descriptives)
```

```
## # A tibble: 9 x 4
```

```
## # Groups:   Diagnosis [?]
##   Diagnosis   Task            Mean S.Dev
##   <fct>       <fct>          <dbl> <dbl>
## 1 Amnesic     Classification    60  14.9
## 2 Amnesic     Grammar           70  10.2
## 3 Amnesic     Recognition       65  12.2
## 4 Huntingtons Classification    40  13.2
## 5 Huntingtons Grammar           45  10.9
## 6 Huntingtons Recognition       95  13.4
## 7 Control     Classification    80  11.7
## 8 Control     Grammar           80  13.0
## 9 Control     Recognition       95  13.0
```

Let's test for the presence of each main effect and interaction using a 3 X 3 factorial ANOVA:

```
patient.aov<-aov(Y~Diagnosis*Task,data=patient.data)
print(summary(patient.aov))
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## Diagnosis       2   5250  2625.0  16.637 7.64e-06 ***
## Task            2   5250  2625.0  16.637 7.64e-06 ***
## Diagnosis:Task  4   5000  1250.0   7.923 0.000109 ***
## Residuals      36   5680   157.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MS.w<-157.8 # Store MS Within for later
```

Although we have a significant main effect of Diagnosis, $F(2, 36) = 16.37, p < .001$, and Task, $F(2, 36) = 16.37, p < .001$, the significant interaction, $F(4, 36) = 7.92, p < .001$ indicates that the effect of Diagnosis on performance (Y) depends on the Task being performed.

At this point, we need to decide whether we want to examine the simple main effect of Diagnosis for each Task, or examine the simple main effect of Task for each Diagnosis. Given that researchers seemed most interested in comparing the performance of the three patient groups across Task, we might decide to examine the effect of Diagnosis within each level of Task.

Examining The Interaction - Two Approaches

Approach #1 - Multiple One-Way ANOVAs

For the first approach, we need to separate the data into subgroups based on the levels of the Task variable. Let's use the **filter** command found in the *tidyverse* package:

```
task.recog<-filter(patient.data,Task=="Recognition")
task.class<-filter(patient.data,Task=="Classification")
task.grammar<-filter(patient.data,Task=="Grammar")
```

Now we need to compare the performance of each patient group within each of the three Task levels by conducting 3 one-way ANOVAs using MS_W from the omnibus ANOVA as the error term. We could then follow each significant one-way ANOVA with a series of pairwise tests:

```r
# Do one-way ANOVA for Recognition Task
recog.aov<-aov(Y~Diagnosis,data=task.recog)
print(summary(recog.aov))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Diagnosis    2   3000  1500.0   9.082 0.00396 **
## Residuals   12   1982   165.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
MS.diagnosis<-1500
F.recog<-MS.diagnosis/MS.w # Re-calculate F using MS.w
p.value.recog<-1-pf(F.recog,df1=3-1,df2=36)
print(p.value.recog)
```

```
## [1] 0.0004845019
```

```r
TukeyHSD(recog.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Y ~ Diagnosis, data = task.recog)
##
## $Diagnosis
##                      diff       lwr      upr     p adj
## Huntingtons-Amnesic   30   8.31523 51.68477 0.0080262
## Control-Amnesic       30   8.31523 51.68477 0.0080262
## Control-Huntingtons    0 -21.68477 21.68477 1.0000000
```

```r
# Do one-way ANOVA for Classification Task
classification.aov<-aov(Y~Diagnosis,data=task.class)
print(summary(classification.aov))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Diagnosis    2   4000  2000.0   11.22 0.00179 **
## Residuals   12   2138   178.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
MS.diagnosis<-2000
F.classification<-MS.diagnosis/MS.w # Re-calculate F using MS.w
p.value.classification<-1-pf(F.classification,df1=3-1,df2=36)
print(p.value.classification)
```

```
## [1] 6.8072e-05
```

```r
TukeyHSD(classification.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Y ~ Diagnosis, data = task.class)
##
## $Diagnosis
##                       diff        lwr       upr     p adj
## Huntingtons-Amnesic  -20 -42.521995  2.521995 0.0839217
## Control-Amnesic       20  -2.521995 42.521995 0.0839217
## Control-Huntingtons   40  17.478005 62.521995 0.0012925
```

```
# Do one-way ANOVA for Grammar Task
grammar.aov<-aov(Y~Diagnosis,data=task.grammar)
print(summary(grammar.aov))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Diagnosis    2   3250    1625    12.5 0.00116 **
## Residuals   12   1560     130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
MS.diagnosis<-1625
F.grammar<-MS.diagnosis/MS.w # Re-calculate F using MS.w
p.value.grammar<-1-pf(F.grammar,df1=3-1,df2=36)
print(p.value.grammar)
```

```
## [1] 0.0002906327
```

```
TukeyHSD(grammar.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Y ~ Diagnosis, data = task.grammar)
##
## $Diagnosis
##                       diff        lwr       upr     p adj
## Huntingtons-Amnesic  -25 -44.238238 -5.761762 0.0119893
## Control-Amnesic       10  -9.238238 29.238238 0.3780273
## Control-Huntingtons   35  15.761762 54.238238 0.0010645
```

Approach #2 - Linear Contrasts

However, given our *a priori* hypotheses about how each patient group ought to perform on tasks of implicit and explicit memory, a better approach would be to use planned contrasts to specifically evaluate whether the pattern of means is consistent with our hypotheses. Note that here too we should use the MS_W from the full 3 X 3 factorial ANOVA as it pools the error variance over all 9 groups (I'm assuming equal variance across groups here).

For the Recognition task, we might choose the weights (-2, 1, 1) to capture the hypothesis that Amnesics in particular should perform poorly on tests of explicit memory.

For the Classification task, we might choose the weights, (1, -2, 1) to capture the hypothesis that Huntingtons patients, but not Amnesics or Controls are impaired on implicit memory tasks.

Given that the Grammar task also measures implicit memory, we can use the same weights as for the Classification task.

```
recog.weights<-c(-2,1,1) # Amnesics, Huntingtons, Controls
classification.weights<-c(1,-2,1) # Amnesics, Huntingtons, Controls
grammar.weights<-c(1,-2,1) # Amnesics, Huntingtons, Controls

# Get condition means for each Task separately
recog.means<-patient.descriptives %>% filter(Task=="Recognition") %>% ungroup() %>% select(Mean)

classification.means<-patient.descriptives %>% filter(Task=="Classification") %>%
  ungroup() %>% select(Mean)

grammar.means<-patient.descriptives %>% filter(Task=="Grammar") %>% ungroup() %>% select(Mean)
```

Now that we have the weights for each Task as well as the means, let's do three linear contrasts:

```
n.pergroup<-5

# Contrast for Recognition Task
recog.psi<-sum(recog.weights*recog.means$Mean)
recog.F<-((recog.psi^2)/(sum((recog.weights^2)/n.pergroup)))/MS.w
recog.p.value<-1-pf(recog.F,df1=1,df2=45-3)
print(recog.p.value)
```

```
## [1] 8.238931e-05
```

```
# Contrast for Classification Task
classification.psi<-sum(classification.weights*classification.means$Mean)
classification.F<-((classification.psi^2)/(sum((classification.weights^2)/n.pergroup)))/MS.w
classification.p.value<-1-pf(classification.F,df1=1,df2=45-3)
print(classification.p.value)
```

```
## [1] 8.238931e-05
```

```
# Contrast for Grammar Task
grammar.psi<-sum(grammar.weights*grammar.means$Mean)
grammar.F<-((grammar.psi^2)/(sum((grammar.weights^2)/n.pergroup)))/MS.w
grammar.p.value<-1-pf(grammar.F,df1=1,df2=45-3)
print(grammar.p.value)
```

```
## [1] 8.238931e-05
```

All three contrasts are significant, which supports our hypotheses concerning the relationship between Amnesia, Huntingtons, and implicit/explicit memory.

Note that I did not use any corrections here to maintain $\alpha_{FW} = .05$. As I alluded to in the previous lab tutorial, use of the Bonferroni adjustment with *planned* contrasts is somewhat controversial. Because I did the omnibus test first and found a significant interaction, my own approach would be to go ahead with the planned contrasts without using a correction.

Effect size and Power

The appropriate effect size measure for a two-way Factorial ANOVA is *partial eta-squared*, η_p^2. Note that your MDK textbook refers to η_p^2 as $R_{partial}^2$. When there are two IVs, η_p^2 can be calculated for each IV as well as the interaction term. Let's use the data from MDK Chapter 7 Table 1 to calculate η_p^2 for the main effects of Drug and BioFeedback. Let's print the output from the ANOVA to get the SS for each term.

```
print(summary(sbp.aov))
```

```
##                  Df Sum Sq Mean Sq F value  Pr(>F)
## Drug              1    720   720.0   11.52 0.00371 **
## BioFeedback       1    500   500.0    8.00 0.01211 *
## Drug:BioFeedback  1    320   320.0    5.12 0.03792 *
## Residuals        16   1000    62.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's store these SS values as new variables, as we'll need them shortly.

```
SS.drug<-720
SS.biofeedback<-500
SS.int<-320
SS.within<-1000
```

The general formula for η_p^2 in a between-subjects factorial design is:

$$\eta_p^2 = \frac{SS_{Effect}}{SS_{Effect} + SS_{Within}}$$

For the main effect of Drug, η_p^2 is:

$$\eta_p^2 = \frac{SS_{Drug}}{SS_{Drug} + SS_{Within}}$$

```
p.etasq.drug<-SS.drug/(SS.drug + SS.within)
print(round(p.etasq.drug,digits=2))
```

```
## [1] 0.42
```

We can also calculate η_p^2 for the main effect of BioFeedback as well as the Drug:BioFeedback interaction similarly:

```
p.etasq.biofeedback<-SS.biofeedback/(SS.biofeedback + SS.within)
print(round(p.etasq.biofeedback,digits=2))
```

```
## [1] 0.33
```

```
p.etasq.int<-SS.int/(SS.int + SS.within)
print(round(p.etasq.int,digits=2))
```

```
## [1] 0.24
```

These are very large effects!

As you might have noticed by now, η_p^2 is closely related to η^2, but the composition of the denominator differs. In a two-factor design, η_p^2 for Factor A is calculated by expressing SS_A as a proportion of the sum of SS_A and SS_{Within}, whereas for η^2, SS_A is expressed as a proportion of the total variability, SS_{Total}, which includes the variability associated with SS_B and the interaction between A and B SS_{AB}. Therefore, when more than one IV is present, η_p^2 will always be larger than η^2.

Like η^2, we can re-express η_p^2 as Cohen's *f*, which we'll need for power and sample size calculations.

$$Cohen's f = \sqrt{\frac{\eta_p^2}{1-\eta_p^2}}$$

We can use Cohen's f values to conduct a power analysis in which we can estimate the number of participants needed to reach a power = .80. Previously, we used the **pwr.anova.test** function when analyzing a one-way between-subjects ANOVA. Here, we'll use the **pwr.f2.test** function, which allows us to calculate power or n for any F-test.

Suppose we have a two-factor design: factors A and B have three and two levels (3 X 2 design), respectively. Also, let's assume that the effect sizes for the main effects of A and B are $f = 0.1$ (i.e., small) and $f = 0.25$ (i.e., medium), respectively. The following commands calculates the n needed to detect a main effect of B when $\alpha = 05$.

```
library(pwr) # make sure the pwr package is installed first - install.packages("pwr")

pwr.f2.test(u=2-1,f2=(.25^2),power=.80,sig.level=.05) # note that the f2 argument is squared (f^2)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 125.5312
##              f2 = 0.0625
##       sig.level = 0.05
##           power = 0.8
```

The arguments u and v correspond to the degrees of freedom (df) in the numerator and the denominator of the F-test, respectively. The df in the denominator = (num levels A) X (num levels B) X (n - 1), where n refers to the number of subjects in each condition. If we know the df, we can re-arrange the equation to solve for n. In this example, $n = 1 + [126/(3 \text{ X } 2)] = 22$.

In other words, in a 3 X 2 design, we would need 22 subjects per condition to detect a main effect of factor B with an effect size, $f = .25$, and power = .80.

Unbalanced (Non-Orthogonal) Designs

To this point, we have only discussed designs where there was an equal n in each group. When n's are unequal, things become considerably more complicated. Recall that when n's are equal, a two-factor factorial ANOVA is really just a set of three orthogonal contrasts. With unequal n's, however, this no longer holds. That is, the contrasts are no longer orthogonal. This raises serious problems in interpreting the main effects in such designs, because they are no longer independent.

The solution is to use what is known as Type III sums-of-squares together with sum-to-zero coding (which we've been doing all along) in order to interpret each main effect independently. This is a complicated issue conceptually, but the implementation in *R* is fairly straightforward. To use Type III *SS* rather than Type I *SS* (which is the default in *R*), we need to use the **lm** command in conjunction with the **drop1** function. The following example will illustrate how to do this in *R* using the data from MDK Chapter 7 Table 15 that lists salaries (in thousands) of males and females with and without college degrees.

```
salary.data<-read.table(file="Ch7T15.txt",header=T)
str(salary.data)
```

```
## 'data.frame':    22 obs. of  3 variables:
##  $ Sex      : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Education: Factor w/ 2 levels "degree","nodegree": 1 1 1 1 1 1 1 1 2 2 ...
##  $ Salary   : int  24 26 25 24 27 24 27 23 15 17 ...
```

Let's get the number of participants in each condition along with the means:

```
salary.descriptives<-salary.data %>% group_by(Sex,Education) %>%
  summarize(Mean=mean(Salary),n=length(Salary))

print(salary.descriptives)
```

```
## # A tibble: 4 x 4
## # Groups:   Sex [?]
##   Sex    Education  Mean     n
##   <fct>  <fct>     <dbl> <int>
## 1 female degree       25     8
## 2 female nodegree     17     4
## 3 male   degree       27     3
## 4 male   nodegree     20     7
```

You can see that each cell does not have the same *n*. To analyze these data, we first need to make sure that we're using sum-to-zero effect coding using the **options** command. Next, we use the **lm** command to build the full model. Then we evaluate the full model against the appropriate restricted models using the **drop1** function:

```
options(contrasts=c("contr.sum","contr.poly")) # Set sum-to-zero effect coding

salary.lm<-lm(Salary~Sex*Education,data=salary.data)
drop1(salary.lm,.~.,test="F")
```

```
## Single term deletions
##
## Model:
## Salary ~ Sex * Education
##               Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>                      50.000 26.062
## Sex            1    29.371  79.371 34.228 10.5734  0.004429 **
## Education      1   264.336 314.336 64.507 95.1608 1.306e-08 ***
## Sex:Education  1     1.175  51.175 24.573  0.4229  0.523690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```