Homework #1 Key Prof. Colleen F Moore
Spring 2014
Psyx 501, Montana State University

Preliminary comments:
The design is a 4x3 factorial between-groups. Non-athletes do aerobic training for 6, 4 or 2 weeks, and then are timed while running one of 4 distances. The data are fake. There are 36 observations, and it is a balanced design.

Key skills here:

- Dummy coding
- Effect coding
- Helmert coding
- Polynomial coding
- Problem in R with Type III SS with dummy and effect coding
- Relationship between regression coefficients and ANOVA tables of means and effects
- That using numerical values for the IV in regression fits the equivalent of the linear part of polynomial contrasts in ANOVA
- Graphs for some of this are found in handout R2-Graphs

Below R commands are preceded by the '>' prompt.
Comments and interpretations are preceded by #

1A. Bring data into R in at least two ways.
I copied from the clipboard.

```
> hmwk1=read.table(pipe("pbpaste"),header=T)
> hmwk1
   dv Dist Train
1   1    1     6
2   2    1     6
3   1    1     6
4   3    1     4
5   2    1     4
## data set truncated here for the handout …

> nrow(hmwk1)   # check that the number of observations is correct
[1] 36
> attach(hmwk1)    # I like to attach the data to avoid the '$' addressing
```

1B. Calculate the grand mean, main effect means, and cell means and standard deviations, plot the Q-Q normal graph. Visually examine homogeneity of variance in the cell sd's and normality in the Q-Q graph. Use functions such as (but not limited to).
summary(variablename), mean(variablename), sd(variablename), boxplot(variablename), boxplot(dv~A, data=datafile), boxplot(dv~A*B, data=datafile), tapply(variablename, IV, mean), tapply(dv, list(A,B), sd), qqnorm(variablename), qqline(variablename)

```
> summary(hmwk1)    # look at descriptives. Helps find data entry errors
       dv               Dist           Train
 Min.   : 1.000   Min.   :1.00   Min.   :2
 1st Qu.: 3.750   1st Qu.:1.75   1st Qu.:2
 Median : 5.000   Median :2.50   Median :4
```

```
 Mean   : 5.861   Mean   :2.50   Mean   :4
 3rd Qu.: 7.000   3rd Qu.:3.25   3rd Qu.:6
 Max.   :16.000   Max.   :4.00   Max.   :6

> tapply(dv, list(Dist, Train), mean)  # table of means arranged by the IVs
          2         4        6
1  5.333333  3.000000 1.333333
2  5.666667  5.000000 2.333333
3  6.000000  4.666667 3.666667
4 15.000000 10.333333 8.000000
> tapply(dv, list(Dist, Train), sd)   # table of sd's arranged by the IVs
          2         4         6
1 0.5773503 1.0000000 0.5773503
2 0.5773503 1.0000000 0.5773503
3 1.0000000 0.5773503 0.5773503
4 1.0000000 1.5275252 1.0000000


# make some boxplots of the main effects, with simple labels
> boxplot(dv~Train, main="dv by Train", xlab="Train", ylab=" dv ")
> boxplot(dv~Dist, main="dv by Dist", xlab="Dist", ylab=" dv ")
# boxplot of cell values
> boxplot(dv~Train*Dist, main="dv by Train*Dist", xlab="Train * Dist")
```

2A. Carry out the ANOVA using the 'aov' function. Get the ANOVA summary including the intercept. Get the means and se's using the 'model.tables' function. Do they match the means you calculated in 1B (they should match)? How did your guess about significance in 1B work out?

```
# First step for anova: convert the numerical condition codes into 'factors'
#  I name my factors differently from the numerical codes so they are separate
# in R's memory

> facTrain=factor(Train)
> facDist=factor(Dist)
> summary(facTrain)  # R will tell you the number of observations per condition
 2  4  6
12 12 12
> summary(facDist)
1 2 3 4
9 9 9 9

#
> mod1=aov(dv~facDist*facTrain)  # the '*' includes the interaction
> summary(mod1, intercept=T)    # these are type I SS's
                 Df Sum Sq Mean Sq  F value   Pr(>F)
(Intercept)       1 1236.7  1236.7 1590.036  < 2e-16 ***
facDist           3  342.3   114.1  146.702 1.44e-15 ***
facTrain          2  104.4    52.2   67.107 1.48e-10 ***
facDist:facTrain  6   22.9     3.8    4.917  0.00208 **
Residuals        24   18.7     0.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> model.tables(mod1, "means", se=T)
# this gives main effect and interaction means, with standard error of diff
Tables of means
```

```
Grand mean

5.861111

 facDist
facDist
     1      2      3      4
 3.222  4.333  4.778 11.111

 facTrain
facTrain
    2     4     6
8.000 5.750 3.833

 facDist:facTrain
       facTrain
facDist 2      4      6
      1  5.333  3.000  1.333
      2  5.667  5.000  2.333
      3  6.000  4.667  3.667
      4 15.000 10.333  8.000

Standard errors for differences of means
        facDist facTrain facDist:facTrain
         0.4157   0.3600           0.7201
replic.       9       12                3
```

2B. Find Type II and Type III SS's for the ANOVA from 2A using the 'Anova' function in the 'car' package for the same model you calculated in part 2A. Compare not just significance but SS's. Which SS are the same?

```
# What contrasts does R use as the default when calculating 'aov' ? Ask.
> contrasts(facDist)
  2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1
> contrasts(facTrain)
  4 6
2 0 0
4 1 0
6 0 1

# these are like 'dummy codes'. The condition labeled 0 is compared with each
# other condition in turn.
# Let's get the Type II and Type III SS's using the 'car' package

> library(car)  # activate the package in memory
> Anova(mod1, type="II")  # notice the capital A on 'Anova'
Anova Table (Type II tests)

Response: dv
                 Sum Sq Df  F value    Pr(>F)
facDist          342.31  3 146.7024 1.438e-15 ***
facTrain         104.39  2  67.1071 1.485e-10 ***
```

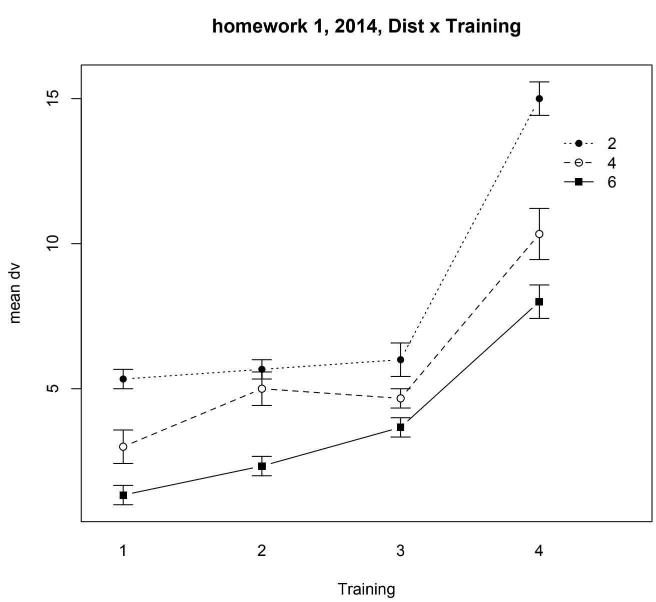
```
facDist:facTrain  22.94  6   4.9167  0.002081 **
Residuals         18.67 24
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> Anova(mod1, type="III", intercept=T)  # ask for Type III SS's
Anova Table (Type III tests)

Response: dv
                  Sum Sq Df  F value    Pr(>F)
(Intercept)       85.333  1 109.7143 1.973e-10 ***
facDist          196.667  3  84.2857 6.960e-13 ***
facTrain          24.222  2  15.5714 4.620e-05 ***
facDist:facTrain  22.944  6   4.9167  0.002081 **
Residuals         18.667 24
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# The Type II SS match the Type I, but Type III does not
# LESSON here is that Type III SS will not match Type I and II when contrasts
#   are R's default.

# Change the contrasts to something orthogonal (helmert or poly), make new model

> options(contrasts=c("contr.helmert","contr.helmert"))
> contrasts(facDist)    # check to see what R has in memory
  [,1] [,2] [,3]
1   -1   -1   -1
2    1   -1   -1
3    0    2   -1
4    0    0    3


> mod2=aov(dv~facDist*facTrain)  # the '*' includes the interaction
> summary(mod2,intercept=T)      # Type I SS's for the Helmert
                 Df Sum Sq Mean Sq  F value   Pr(>F)
(Intercept)       1 1236.7  1236.7 1590.036  < 2e-16 ***
facDist           3  342.3   114.1  146.702 1.44e-15 ***
facTrain          2  104.4    52.2   67.107 1.48e-10 ***
facDist:facTrain  6   22.9     3.8    4.917  0.00208 **
Residuals        24   18.7     0.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> Anova(mod2, type="II")    # Type II SS's for Helmert contrasts match Type I
Anova Table (Type II tests)

Response: dv
                 Sum Sq Df  F value    Pr(>F)
facDist          342.31  3 146.7024 1.438e-15 ***
facTrain         104.39  2  67.1071 1.485e-10 ***
facDist:facTrain  22.94  6   4.9167  0.002081 **
Residuals         18.67 24
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> Anova(mod2, type="III", intercept=T)
Anova Table (Type III tests)

Response: dv
                   Sum Sq Df   F value    Pr(>F)
(Intercept)       1236.69  1 1590.0357 < 2.2e-16 ***
facDist            342.31  3  146.7024 1.438e-15 ***
facTrain           104.39  2   67.1071 1.485e-10 ***
facDist:facTrain    22.94  6    4.9167  0.002081 **
Residuals           18.67 24
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Type III SS's for Helmert contrasts match Type I and Type II
## TAKE-AWAY message: use orthogonal contrasts to get correct Type III SS's
```

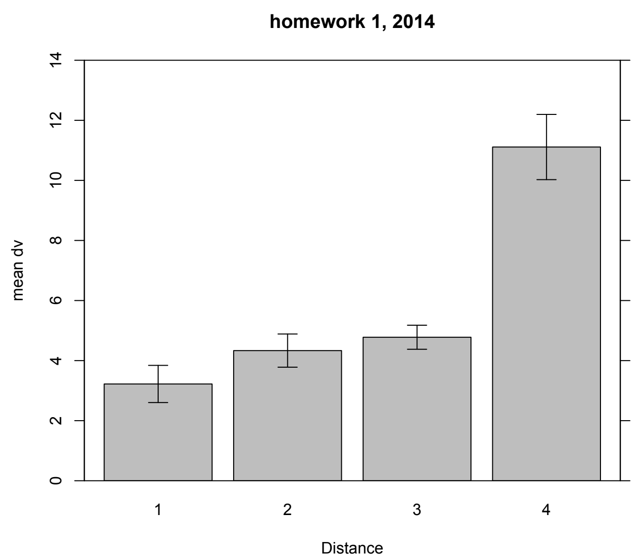
2D. Make separate graphs of the main effects and cell means with standard error bars. No, using 'boxplot' isn't enough. Try installing 'ggplot2' or 'sciplot'

```
# try 'sciplot'. This makes a line graph, which is appropriate for
# numerical IVs as we have in this fake study
> library(sciplot)
> lineplot.CI(x.factor=facDist, response=dv,
group=facTrain,xlab="Training",ylab="mean dv", main="homework 1, 2014, Dist x
Training")

## gives an interaction plot. I made a mistake in labeling the x-axis,
should say Distance
```

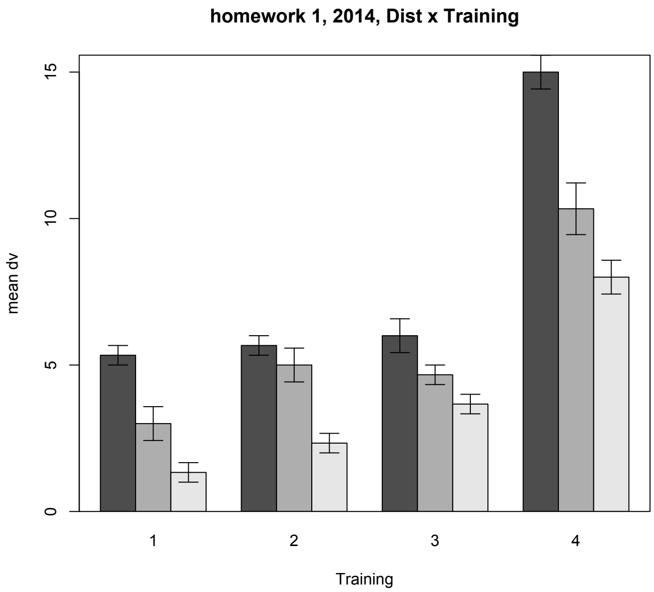



```
# make barplots
(see handout R2-Graphs)
> library(sciplot)
```

```
> bargraph.CI(x.factor=facDist, response=dv, xlab="Distance",ylab="mean
dv", main="homework 1, 2014", ylim=c(0,14))  # main effect of whatever is
called x.factor
> box()

## standard error bars vary by condition in the graph. In a main effect of
a factorial design with equal numbers of observations in the cells, I
normally prefer an estimate of the s.e. based on the pooled error from the
ANOVA. For this example, MSerror = 0.7778, and there are 9 observations in
each main effect mean.
##

> seTrain=sqrt(0.7777917/9)
> seTrain
[1] 0.293975

# compare the pooled s.e. calculated here to the bars below. See handout
R2-Graphs for how to re-make the graph with error bars you calculate
yourself.
```




```
> bargraph.CI(x.factor=Dist, response=dv,
group=Train,xlab="Training",ylab="mean dv", main="homework 1, 2014, Dist x
Training")   # an interaction plot
> box()   # add a box around the plot
```



3. Now you'll do the whole thing over again with multiple regression.
Step 1: Prepare data by creating the 3 kinds of contrast codes for your factors. How many df does your design have? (A: #cells -1). Create a predictor variable for each df. You can do this all in one spreadsheet or text file, or you can generate them in R.

3A. Create dummy contrast codes. Select a comparison group and give it zeros for all your predictor variables. Each other group in the design will get 1's in just one of the predictor variables and zeros for all the others.

Do regression using the 'lm' function in R. Ask for the 'summary' and also the 'anova' (all lower case) of your model.

Examine the coefficients (estimates) in the summary, and the SS's in the 'anova' table. Compare the SS's to those from 2A (sum up the SS's in the regression).

Write-up: What do the coefficients mean? Write an interpretation of one of the coefficients. Write the regression equation with the estimates in it and then describe what it means. Compare the standard errors of the estimates to the se's from the 'model.tables' command you ran in 2A. What is the same and why? Now use the 'anova' results of your regression and re-create the ANOVA source table from 2A by adding SS's and df's together appropriately. Describe how you did this in words.

Output to include: The summary and anova.

```
# I created dummy codes in excel and pasted the data with codes into R
# Note: I closed R without saving, and started all over here.

> hmwk1=read.table(pipe("pbpaste"),header=T)
> hmwk1
  dv Dist Train dum1 dum2 dum3 dum4 dum5 dum6 dum7 dum8 dum9 dum10 dum11
1  1    1     6    0    0    0    0    0    0    0    0    0     0     0
2  2    1     6    0    0    0    0    0    0    0    0    0     0     0
3  1    1     6    0    0    0    0    0    0    0    0    0     0     0
4  3    1     4    1    0    0    0    0    0    0    0    0     0     0
5  2    1     4    1    0    0    0    0    0    0    0    0     0     0
6  4    1     4    1    0    0    0    0    0    0    0    0     0     0
```

```
7    5    1    2    0    1    0    0    0    0    0    0    0    0    0
8    6    1    2    0    1    0    0    0    0    0    0    0    0    0
9    5    1    2    0    1    0    0    0    0    0    0    0    0    0
10   2    2    6    0    0    1    0    0    0    0    0    0    0    0
11   2    2    6    0    0    1    0    0    0    0    0    0    0    0
# data truncated here ....

# run a simple regression model
> attach(hmwk1)
> mod1r=lm(dv~dum1+dum2+dum3+dum4+dum5+dum6+dum7+dum8+dum9+dum10+dum11)
> summary(mod1r)

Call:
lm(formula = dv ~ dum1 + dum2 + dum3 + dum4 + dum5 + dum6 + dum7 +
    dum8 + dum9 + dum10 + dum11)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3333 -0.4167  0.0000  0.4167  1.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3333     0.5092   2.619 0.015056 *
dum1          1.6667     0.7201   2.315 0.029510 *
dum2          4.0000     0.7201   5.555 1.03e-05 ***
dum3          1.0000     0.7201   1.389 0.177669
dum4          3.6667     0.7201   5.092 3.29e-05 ***
dum5          4.3333     0.7201   6.018 3.26e-06 ***
dum6          2.3333     0.7201   3.240 0.003483 **
dum7          3.3333     0.7201   4.629 0.000107 ***
dum8          4.6667     0.7201   6.481 1.06e-06 ***
dum9          6.6667     0.7201   9.258 2.17e-09 ***
dum10         9.0000     0.7201  12.499 5.36e-12 ***
dum11        13.6667     0.7201  18.979 5.86e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8819 on 24 degrees of freedom
Multiple R-squared:  0.9618,  Adjusted R-squared:  0.9443
F-statistic: 54.89 on 11 and 24 DF,  p-value: 2.83e-14
```

 # what do the coefficients mean?

```
# The intercept is the 'comparison group', the group with all zeros.
# In the coding here, that is 6 wks Training, and Distance =1. The mean is 1.33.
# dum1 compares 4 wks Training and Dist=1 to 6wks Training and Dist=1.
    and 3.00 - 1.333 = 1.667.
# dum11 compares 2wks Training and Dist=4 to 6wks Training and Dist=1. 15 -
1.333 = 13.667.
# The coefficients tell us how much the mean of each grouop differs from the
comparison group.



> anova(mod1r)    # get SS's for the regression model
Analysis of Variance Table
```

```
Response: dv
          Df  Sum Sq Mean Sq  F value    Pr(>F)
dum1       1  26.790  26.790  34.4448 4.707e-06 ***
dum2       1   2.048   2.048   2.6338  0.117677
dum3       1  49.837  49.837  64.0762 3.123e-08 ***
dum4       1   8.963   8.963  11.5238  0.002389 **
dum5       1   4.667   4.667   6.0000  0.021983 *
dum6       1  38.889  38.889  50.0000 2.609e-07 ***
dum7       1  30.044  30.044  38.6286 2.012e-06 ***
dum8       1  17.067  17.067  21.9429 9.267e-05 ***
dum9       1   1.778   1.778   2.2857  0.143627
dum10      1   9.389   9.389  12.0714  0.001963 **
dum11      1 280.167 280.167 360.2143 5.860e-16 ***
Residuals 24  18.667   0.778
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# the SS residual matches the residual from the ANOVA earlier
# the sum of the SS will add up, see below for effect coding, 3C.
```

3B. Repeat 3A, but for orthogonal contrast coding. Start by creating an orthogonal set of contrasts on factor A and B of your design (I suggest using Helmert or polynomial contrasts), and then generate the AxB contrasts. Then fill these into your predictor variables appropriately. Then do the regression, etc.

```
##
## Note: I closed R without saving and started all over here.
## now Helmert coding, and we will reconstruct the SS of the 'aov'
##  I constructed helmert codes in excel, pasted the data again into R
# hel1, hel2, hel3 are the main effect of Dist
# hel4, hel5 represent Train
# the others are pieces of the interaction

> hmwk1
   dv Dist Train hel1 hel2 hel3 hel4 hel5 hel1x4 hel1x5 hel2x4 hel2x5 hel3x4
1   1    1     6    3    0    0    2    0      6      0      0      0      0
2   2    1     6    3    0    0    2    0      6      0      0      0      0
3   1    1     6    3    0    0    2    0      6      0      0      0      0
4   3    1     4    3    0    0   -1    1     -3      3      0      0      0
5   2    1     4    3    0    0   -1    1     -3      3      0      0      0
6   4    1     4    3    0    0   -1    1     -3      3      0      0      0
7   5    1     2    3    0    0   -1   -1     -3     -3      0      0      0
8   6    1     2    3    0    0   -1   -1     -3     -3      0      0      0
9   5    1     2    3    0    0   -1   -1     -3     -3      0      0      0
10  2    2     6   -1    2    0    2    0     -2      0      4      0      0
11  2    2     6   -1    2    0    2    0     -2      0      4      0      0
# data truncated here...
   hel3x5
1       0
2       0
3       0
4       0
5       0
6       0
```

```
7        0
8        0
9        0
10       0
11       0
# data truncated here...

> attach(hmwk1)
>
mod3r=lm(dv~hel1+hel2+hel3+hel4+hel5+hel1x4+hel1x5+hel2x4+hel2x5+hel3x4+hel3x5)
> summary(mod3r)

Call:
lm(formula = dv ~ hel1 + hel2 + hel3 + hel4 + hel5 + hel1x4 +
    hel1x5 + hel2x4 + hel2x5 + hel3x4 + hel3x5)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3333 -0.4167  0.0000  0.4167  1.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.86111    0.14699  39.875  < 2e-16 ***
hel1        -0.87963    0.08486 -10.365 2.43e-10 ***
hel2        -1.20370    0.12001 -10.030 4.64e-10 ***
hel3        -3.16667    0.20787 -15.234 7.75e-14 ***
hel4        -1.01389    0.10393  -9.755 7.96e-10 ***
hel5        -1.12500    0.18002  -6.249 1.85e-06 ***
hel1x4       0.02315    0.06001   0.386  0.70307
hel1x5      -0.01389    0.10393  -0.134  0.89481
hel2x4       0.01852    0.08486   0.218  0.82910
hel2x5       0.38889    0.14699   2.646  0.01416 *
hel3x4       0.50000    0.14699   3.402  0.00235 **
hel3x5       0.83333    0.25459   3.273  0.00321 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8819 on 24 degrees of freedom
Multiple R-squared:  0.9618,  Adjusted R-squared:  0.9443
F-statistic: 54.89 on 11 and 24 DF,  p-value: 2.83e-14

> anova(mod3r)
Analysis of Variance Table

Response: dv
          Df  Sum Sq Mean Sq  F value    Pr(>F)
hel1       1  83.565  83.565 107.4405 2.428e-10 ***
hel2       1  78.241  78.241 100.5952 4.639e-10 ***
hel3       1 180.500 180.500 232.0714 7.748e-14 ***
hel4       1  74.014  74.014  95.1607 7.961e-10 ***
hel5       1  30.375  30.375  39.0536 1.852e-06 ***
hel1x4     1   0.116   0.116   0.1488  0.703074
hel1x5     1   0.014   0.014   0.0179  0.894809
hel2x4     1   0.037   0.037   0.0476  0.829104
hel2x5     1   5.444   5.444   7.0000  0.014157 *
hel3x4     1   9.000   9.000  11.5714  0.002348 **
hel3x5     1   8.333   8.333  10.7143  0.003215 **
```

```
Residuals 24  18.667   0.778
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


# add up SS's to create an anova table that matches the one from 'aov'
# these are just 'hand calculations' below

> SSDist=83.565+78.241+180.500
> SSTrain=74.014+30.375
> SSDxT=0.116+0.014+0.037+5.444+9.000+8.333
> SSDist; SSTrain; SSDxT
[1] 342.306
[1] 104.389
[1] 22.944

# matches the SS's from the aov.
```

3C. Repeat, but for effect coding. You can generate the effect codes by substituting -1 for the 0's of the comparison group.

```
# Note: I closed R and started all over again
# Next, Effect codes. Comparison group is coded -1.

> hmwk1=read.table(pipe("pbpaste"),header=T)
> hmwk1
> hmwk1
   dv Dist Train eff1 eff2 eff3 eff4 eff5 eff6 eff7 eff8 eff9 eff10 eff11
1   1    1     6   -1   -1   -1   -1   -1   -1   -1   -1   -1    -1    -1
2   2    1     6   -1   -1   -1   -1   -1   -1   -1   -1   -1    -1    -1
3   1    1     6   -1   -1   -1   -1   -1   -1   -1   -1   -1    -1    -1
4   3    1     4    1    0    0    0    0    0    0    0    0     0     0
5   2    1     4    1    0    0    0    0    0    0    0    0     0     0
6   4    1     4    1    0    0    0    0    0    0    0    0     0     0
7   5    1     2    0    1    0    0    0    0    0    0    0     0     0
8   6    1     2    0    1    0    0    0    0    0    0    0     0     0
9   5    1     2    0    1    0    0    0    0    0    0    0     0     0
10  2    2     6    0    0    1    0    0    0    0    0    0     0     0
11  2    2     6    0    0    1    0    0    0    0    0    0     0     0
# data truncated here ....

> mod2r=lm(dv~eff1+eff2+eff3+eff4+eff5+eff6+eff7+eff8+eff9+eff10+eff11)

> summary(mod2r)

Call:
lm(formula = dv ~ eff1 + eff2 + eff3 + eff4 + eff5 + eff6 + eff7 +
    eff8 + eff9 + eff10 + eff11)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3333 -0.4167  0.0000  0.4167  1.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)   5.8611     0.1470  39.875  < 2e-16 ***
eff1         -2.8611     0.4875  -5.869 4.71e-06 ***
eff2         -0.5278     0.4875  -1.083 0.289732
eff3         -3.5278     0.4875  -7.236 1.78e-07 ***
eff4         -0.8611     0.4875  -1.766 0.090049 .
eff5         -0.1944     0.4875  -0.399 0.693522
eff6         -2.1944     0.4875  -4.501 0.000148 ***
eff7         -1.1944     0.4875  -2.450 0.021951 *
eff8          0.1389     0.4875   0.285 0.778164
eff9          2.1389     0.4875   4.387 0.000197 ***
eff10         4.4722     0.4875   9.174 2.58e-09 ***
eff11         9.1389     0.4875  18.747 7.74e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8819 on 24 degrees of freedom
Multiple R-squared:  0.9618,  Adjusted R-squared:  0.9443
F-statistic: 54.89 on 11 and 24 DF,  p-value: 2.83e-14


#  In effect coding, what do coefficients mean?
#  Intercept is the grand mean !
# Each coefficient is comparison of the the condition coded 1 to the grand mean

# find SS's
> anova(mod2r)
Analysis of Variance Table

Response: dv
          Df  Sum Sq Mean Sq  F value    Pr(>F)
eff1       1   4.167   4.167   5.3571 0.0295104 *
eff2       1  20.056  20.056  25.7857 3.412e-05 ***
eff3       1   1.778   1.778   2.2857 0.1436269
eff4       1   9.600   9.600  12.3429 0.0017830 **
eff5       1  12.844  12.844  16.5143 0.0004484 ***
eff6       1   0.032   0.032   0.0408 0.8415956
eff7       1   2.149   2.149   2.7628 0.1094897
eff8       1  12.042  12.042  15.4821 0.0006213 ***
eff9       1  40.833  40.833  52.5000 1.739e-07 ***
eff10      1  92.803  92.803 119.3182 8.526e-11 ***
eff11      1 273.336 273.336 351.4318 7.739e-16 ***
Residuals 24  18.667   0.778
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Sum of the SS for eff1 to eff11 = sum of SS(Dist, Train and DistxTrain) in
aov. Again, these are 'hand calculations done in R.

> sumaov=104.39+342.31+22.94; sumaov
[1] 469.64
> sumss=4.167+20.056+1.778+9.6+12.844+.032+2.149+12.042+40.833+92.803+273.336
> sumss
[1] 469.64
```

3D. This example is begging for us to use the I.V.'s as numerical variables, instead of categorical variables. Run the analysis that way, compare the results to your other 'lm' analyses.

```
# Note: I closed R and started all over again here
# First I did this with the 'raw' values of Train and Dist
# Then I 'centered' the variables by subtracting their means

# Raw numbers
> modnumerical=lm(dv~Train*Dist)
> summary(modnumerical)

Call:
lm(formula = dv ~ Train * Dist)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3500 -0.7833  0.2167  1.2722  3.8389 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.0000     2.0718   0.965 0.341612    
Train        -0.5417     0.4795  -1.130 0.267049    
Dist          3.2111     0.7565   4.245 0.000175 ***
Train:Dist   -0.2000     0.1751  -1.142 0.261842    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 1.918 on 32 degrees of freedom
Multiple R-squared:  0.7589,  Adjusted R-squared:  0.7363 
F-statistic: 33.57 on 3 and 32 DF,  p-value: 5.295e-10

> anova(modnumerical)
Analysis of Variance Table

Response: dv
           Df Sum Sq Mean Sq F value    Pr(>F)    
Train       1 104.17 104.167 28.3126 7.809e-06 ***
Dist        1 261.61 261.606 71.1046 1.233e-09 ***
Train:Dist  1   4.80   4.800  1.3046    0.2618    
Residuals  32 117.73   3.679                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

# The SS Residual exploded compared to all other models
# Each variable has 1 df rather than the (number of levels - 1)


# Transform the variables by subtracting mean-- this centers them
#
> TrainNew=Train-4
> DistNew=Dist-2.5

> modNew=lm(dv~TrainNew*DistNew)
> summary(modNew)

Call:
lm(formula = dv ~ TrainNew * DistNew)

Residuals:
```

```
    Min      1Q  Median      3Q     Max 
-4.3500 -0.7833  0.2167  1.2722  3.8389 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)        5.8611     0.3197  18.334  < 2e-16 ***
TrainNew          -1.0417     0.1958  -5.321 7.81e-06 ***
DistNew            2.4111     0.2859   8.432 1.23e-09 ***
TrainNew:DistNew  -0.2000     0.1751  -1.142    0.262    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 1.918 on 32 degrees of freedom
Multiple R-squared:  0.7589,  Adjusted R-squared:  0.7363 
F-statistic: 33.57 on 3 and 32 DF,  p-value: 5.295e-10

# Now the Intercept is the grand mean.
# The interpretation of the coefficients is:
#  For each 1 unit difference from the average Training, the dv decreases
# by 1.04.  For each 1 unit increase in Distance, the dv increases by 2.4.


> anova(modNew)
Analysis of Variance Table

Response: dv
                 Df Sum Sq Mean Sq F value    Pr(>F)    
TrainNew          1 104.17 104.167 28.3126 7.809e-06 ***
DistNew           1 261.61 261.606 71.1046 1.233e-09 ***
TrainNew:DistNew  1   4.80   4.800  1.3046    0.2618    
Residuals        32 117.73   3.679                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 


# the predictions and residuals from the two numerical models are identical. # The SS are identical for the 'centered' variables or not.
# The interpretation of the coefficients differs, and the intercept differs.
```

 # Here's a good puzzler: what is the relationship between the coefficients in these analyses and the Linear component of a polynomial contrast??

```
# Close R and start all over again
## Figure out relationship to 'orthogonal polynomial' contrasts
## bring in data with polynomial codes created in excel

> hmwk1=read.table(pipe("pbpaste"),header=T)
> hmwk1
   dv Dist Train poly1 poly2 poly3 poly4 poly5 p1x4 p1x5 p2x4 p2x5 p3x4 p3x5
1   1    1     6    -3     1    -1     1     1   -3   -3    1    1   -1   -1
2   2    1     6    -3     1    -1     1     1   -3   -3    1    1   -1   -1
3   1    1     6    -3     1    -1     1     1   -3   -3    1    1   -1   -1
4   3    1     4    -3     1    -1     0    -2    0    6    0   -2    0    2
5   2    1     4    -3     1    -1     0    -2    0    6    0   -2    0    2
6   4    1     4    -3     1    -1     0    -2    0    6    0   -2    0    2
7   5    1     2    -3     1    -1    -1     1    3   -3   -1    1    1   -1
8   6    1     2    -3     1    -1    -1     1    3   -3   -1    1    1   -1
```

```
9    5    1     2    -3    1    -1    -1    1    3   -3   -1    1    1   -1
10   2    2     6    -1   -1     3     1    1   -1   -1   -1   -1    3    3
11   2    2     6    -1   -1     3     1    1   -1   -1   -1   -1    3    3


> polyall=lm(dv~poly1+poly2+poly3+poly4+poly5+p1x4+p1x5+p2x4+p2x5+p3x4+p3x5)
> summary(polyall)

Call:
lm(formula = dv ~ poly1 + poly2 + poly3 + poly4 + poly5 + p1x4 +
    p1x5 + p2x4 + p2x5 + p3x4 + p3x5)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3333 -0.4167  0.0000  0.4167  1.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.86111    0.14699  39.875  < 2e-16 ***
poly1        1.20556    0.06573  18.340 1.27e-15 ***
poly2        1.30556    0.14699   8.882 4.72e-09 ***
poly3        0.32778    0.06573   4.986 4.30e-05 ***
poly4       -2.08333    0.18002 -11.573 2.64e-11 ***
poly5        0.05556    0.10393   0.535  0.59790
p1x4        -0.20000    0.08051  -2.484  0.02036 *
p1x5         0.06111    0.04648   1.315  0.20102
p2x4        -0.66667    0.18002  -3.703  0.00111 **
p2x5         0.19444    0.10393   1.871  0.07361 .
p3x4        -0.15000    0.08051  -1.863  0.07472 .
p3x5        -0.04444    0.04648  -0.956  0.34851
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8819 on 24 degrees of freedom
Multiple R-squared:  0.9618,   Adjusted R-squared:  0.9443
F-statistic: 54.89 on 11 and 24 DF,  p-value: 2.83e-14

> anova(polyall)
Analysis of Variance Table

Response: dv
          Df  Sum Sq Mean Sq  F value    Pr(>F)
poly1      1 261.606 261.606 336.3500 1.267e-15 ***
poly2      1  61.361  61.361  78.8929 4.720e-09 ***
poly3      1  19.339  19.339  24.8643 4.303e-05 ***
poly4      1 104.167 104.167 133.9286 2.637e-11 ***
poly5      1   0.222   0.222   0.2857  0.597898
p1x4       1   4.800   4.800   6.1714  0.020355 *
p1x5       1   1.344   1.344   1.7286  0.201019
p2x4       1  10.667  10.667  13.7143  0.001111 **
p2x5       1   2.722   2.722   3.5000  0.073611 .
p3x4       1   2.700   2.700   3.4714  0.074720 .
p3x5       1   0.711   0.711   0.9143  0.348513
Residuals 24  18.667   0.778
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# poly1 is linear contrast on Dist. It matches the SS for Dist for the numerical
analysis
# poly4 is the linear contrast on Training. It matches the SS for the analysis
of the Train numerical variable.
# p1x4 is the linear x linear interaction contrast for Dist x Train. It matches
the SS for Dist x Train in the numerical analysis

# To show that using the numerical variables calculates the linear trends only,
now we will calculate a partial model using only the linear contrast variables

> polylin=lm(dv~poly1+poly4+p1x4)
> summary(polylin)

Call:
lm(formula = dv ~ poly1 + poly4 + p1x4)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3500 -0.7833  0.2167  1.2722  3.8389

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.8611     0.3197  18.334  < 2e-16 ***
poly1         1.2056     0.1430   8.432 1.23e-09 ***
poly4        -2.0833     0.3915  -5.321 7.81e-06 ***
p1x4         -0.2000     0.1751  -1.142    0.262
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.918 on 32 degrees of freedom
Multiple R-squared:  0.7589,	Adjusted R-squared:  0.7363
F-statistic: 33.57 on 3 and 32 DF,  p-value: 5.295e-10

# Note: the coefficients differ because the contrasts are coded differently from
the original numbers

> anova(polylin)
Analysis of Variance Table

Response: dv
          Df Sum Sq Mean Sq F value    Pr(>F)
poly1      1 261.61 261.606 71.1046 1.233e-09 ***
poly4      1 104.17 104.167 28.3126 7.809e-06 ***
p1x4       1   4.80   4.800  1.3046    0.2618
Residuals 32 117.73   3.679
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# the SS match those from the numerical analysis.
# notice that the SS residual has ballooned
```