Handout R14
Psychology 501 Analysis of Covariance
Prof Colleen Moore
Montana State University

This handout describes the basics of carrying out an analysis of covariance in R. The data are from Kirk, 3rd Ed., p. 720. The example illustrates a case in which there are differences among the groups in the covariate.

Overview:

I. Bring in data, create factors if you entered the conditions numerically, use ‘aov’ to fit model, use ‘Anova’ in ‘car’ package to get Type III SS.

New: install the package ‘effects’ which will give us adjusted means.

I.A. Bring in data and do some set up

```
> library(car)  # bring in 'car' package
> options(contrasts=c("contr.helmert","contr.helmert"))
# set options so that you will get correct type III SS
> datafilename <-file.choose() # navigate to your datafile
> kirk=read.table(datafilename,header=TRUE)
> kirk # check the data
# I entered it as a One way design and as a 2x2 just for fun.
# we have 4 cells, 32 total observations, 8 per cell, balanced!

   Oneway A B dv covar
1       1 1 1  3    42
2       1 1 1  6    57
3       1 1 1  3    33
4       1 1 1  3    47
5       1 1 1  1    32
6       1 1 1  2    35
7       1 1 1  2    33
8       1 1 1  2    39
9       2 2 1  4    47
10      2 2 1  5    49
11      2 2 1  4    42
12      2 2 1  3    41
13      2 2 1  2    38
14      2 2 1  3    43
15      2 2 1  4    48
16      2 2 1  3    45
17      3 1 2  7    61
18      3 1 2  8    65
```

```
19       3 1 2  7    64
20       3 1 2  6    56
21       3 1 2  5    52
22       3 1 2  6    58
23       3 1 2  5    53
24       3 1 2  6    54
25       4 2 2  7    65
26       4 2 2  8    74
27       4 2 2  9    80
28       4 2 2  8    73
29       4 2 2 10    85
30       4 2 2 10    82
31       4 2 2  9    78
32       4 2 2 11    89
```

```
> attach(kirk)
> facOne=factor(Oneway) # turns numbers into factor levels
```

I.B. Look over some descriptives and eyeball assumptions

```
# eyeball normal distribution for dv and covar
> qqnorm(dv); qqline(dv)
> qqnorm(covar); qqline(covar)
# how about some boxplots to see a lot at once
> boxplot(dv~facOne)
> boxplot(covar~facOne)  # plot the covariate too
> tapply(dv, facOne, 'mean')  # get cell means

   1    2    3    4
2.75 3.50 6.25 9.00

>  tapply(dv, facOne, 'sd')  # cell s.d.'s
        1         2         3         4
1.4880476 0.9258201 1.0350983 1.3093073
```

Do the same for the covariate.

I.C. First let's do the regular anova of the dv, omitting the covariate.

```
> mod1=aov(dv~facOne)
> summary(mod1,intercept=T)
            Df Sum Sq Mean Sq F value   Pr(>F)
(Intercept)  1  924.5   924.5  631.37  < 2e-16 ***
facOne       3  194.5    64.8   44.28 9.32e-11 ***
Residuals   28   41.0     1.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results should be the same for the Type III SS from 'Anova'. The design is equal-n, and we set the contrasts to 'helmert'.

```
> Anova(mod1, type="III")  # type III ss
Anova Table (Type III tests)

Response: dv
             Sum Sq Df F value    Pr(>F)    
(Intercept)  924.5  1 631.366 < 2.2e-16 ***
facOne       194.5  3  44.276 9.321e-11 ***
Residuals     41.0 28                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## now get cell means
## 'effect' is in package 'effects'

> library(effects)  # activate the package
> effectmeans=effect("facOne",mod1,se=T); summary(effectmeans)

 facOne effect
facOne
   1    2    3    4 
2.75 3.50 6.25 9.00 

 Lower 95 Percent Confidence Limits
facOne
       1        2        3        4 
1.873637 2.623637 5.373637 8.123637 

 Upper 95 Percent Confidence Limits
facOne
       1        2        3        4 
3.626363 4.376363 7.126363 9.876363 

## same results as for 'model.tables'
##
> model.tables(mod1,"means",se=T)
Tables of means
Grand mean
         
5.375 

 facOne
facOne
   1    2    3    4 
2.75 3.50 6.25 9.00 

Standard errors for differences of means
        facOne
         0.605
replic.      8
```

I. D. Now put the covariate in the model. If we use 'Anova' and type III SS, the order we list the covariate and the factor shouldn't matter. In 'aov' by itself, order matters because it is giving type I SS.

```
> mod2=aov(dv~facOne+covar); Anova(mod2, type="III")
Anova Table (Type III tests)

Response: dv
             Sum Sq Df  F value    Pr(>F)
(Intercept)  5.856  1  22.4361 6.181e-05 ***
facOne       1.793  3   2.2898     0.101
covar       33.953  1 130.0916 7.861e-12 ***
Residuals    7.047 27
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(mod2,intercept=T)
# this is the summary of the 'aov' type I SS
## Notice that results don't match !!!

            Df Sum Sq Mean Sq F value   Pr(>F)
(Intercept)  1  924.5   924.5  3542.2  < 2e-16 ***
facOne       3  194.5    64.8   248.4  < 2e-16 ***
covar        1   34.0    34.0   130.1 7.86e-12 ***
Residuals   27    7.0     0.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##  results will match Type III if we enter our covariate first

## model.tables will give unadjusted means, same as from the
## anova without the covariate


> model.tables(mod2,"means",se=T)
Tables of means
Grand mean

5.375

 facOne
facOne
   1    2    3    4
2.75 3.50 6.25 9.00

 covar
covar
   32    33    35    38    39    41    42    43    45    47    48
49    52    53
```

```
4.074 4.242 4.578 4.347 5.249 4.850 5.385 5.186 5.522 6.225 6.026
6.193 4.389 4.557
   54    56    57    58    61    64    65    73    74    78    80
82    85    89
4.724 5.060 8.271 5.396 5.900 6.403 4.861 4.494 4.662 5.333 5.669
6.005 6.508 7.180

Standard errors for differences of means
        facOne
        0.2554
replic.      8
Warning message:
In replications(paste("~", xx), data = mf) : non-factors ignored:
covar
```

find adjusted means using 'effect' command in 'effects'
package

```
> adjmeans=effect("facOne",mod2,se=T); summary(adjmeans)facOne
effect

facOne
       1        2        3        4
5.310127 5.325664 5.767353 5.096856

 Lower 95 Percent Confidence Limits
facOne
       1        2        3        4
4.718978 4.830475 5.386713 4.302900

 Upper 95 Percent Confidence Limits
facOne
       1        2        3        4
5.901275 5.820853 6.147994 5.890813
```

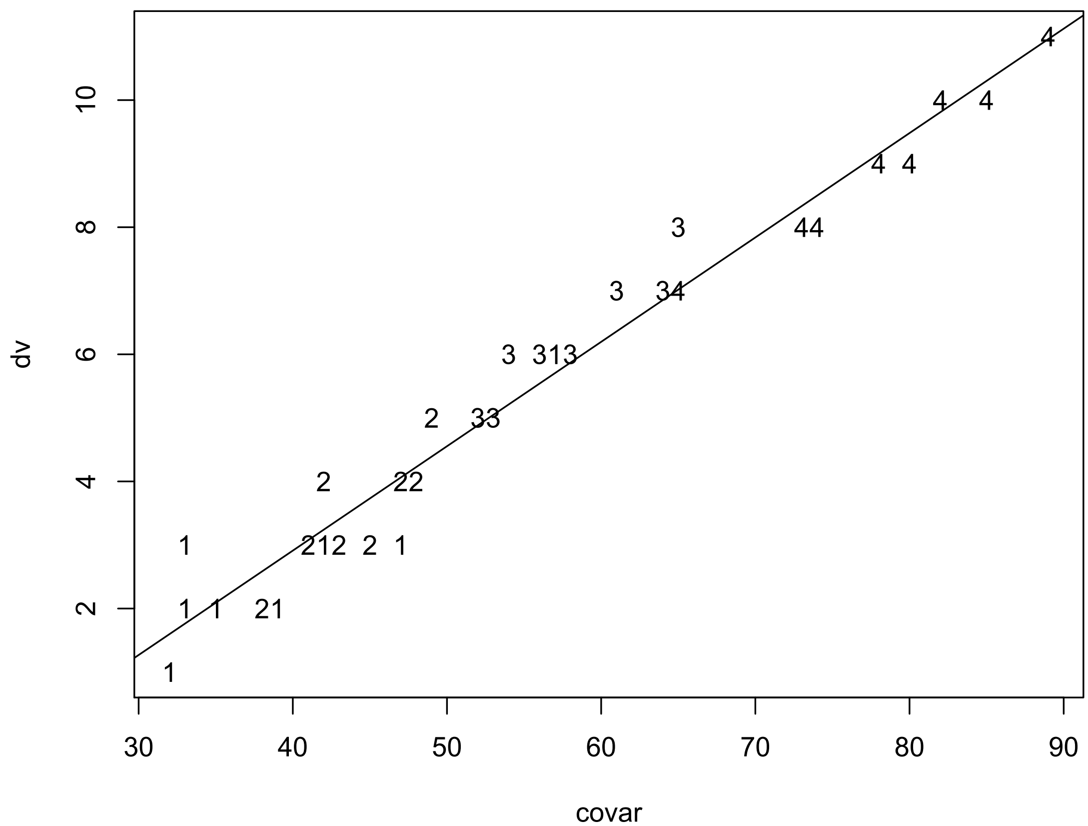
wow, because the covariate is affected by treatment, it really changes the estimated cell means

I. E. Test homogeneity of regression (are regression slopes equivalent across groups?) by entering the interaction of covar x facOne.

First, let's graph some things, look and think

```
# graph the scatter plot of dv and covar with a separate symbol
for each condition
> plot(dv~covar, pch=as.character(facOne))
#  'pch' tells it to use condition as the plot symbol
> abline(lm(dv~covar))  # this adds the regression line
> line(dv~covar)
# this adds a robustly rather than least-squares fitted line
## Neither of these line are the ANCOVA fitted regression, but
```

```
## they are good for eyeballing.
```



Second, statistically test homogeneity of regression by testing the significance of the covar x factor interactions term.

```
## if you have entered your treatments as numbers
##   make sure you use the 'factor' version of your treatments
> mod3=aov(dv~covar*facOne)
#the * says to use main effs and interaction
> Anova(mod3, type="III")
Anova Table (Type III tests)

Response: dv
              Sum Sq Df F value    Pr(>F)
(Intercept)   5.7142  1 20.7312 0.0001294 ***
covar        25.8488  1 93.7796 9.169e-10 ***
facOne        0.3551  3  0.4294 0.7337919
covar:facOne  0.4316  3  0.5220 0.6712657
Residuals     6.6152 24
```

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

## The interaction of the covariate x facOne is not significant.
## This agrees with what we see when looking at the scatter plot.

(things to do: get the estimated 'pooled' regression coefficient
from R and hand-calculate adjusted means to illustrate)
```