Analysis of Covariance
Introduction to Statistics Using R (Psychology 9041B)

Paul Gribble

Winter, 2016

1 ANCOVA

The analysis of covariance (ANCOVA) model includes a continuous variable X (called a *covariate*) that is typically correlated with the dependent variable.

$$Y_{ijk} = \mu + \beta X_{ij} + \alpha_j + \epsilon_{ijk} \tag{1}$$

Usually values of the covariate are collected, or available, prior to any experimental manipulation. This ensures that values of the covariate are independent of the experimental factors. If they are not independent, the logic of ANCOVA falls apart.

Including a covariate in the model results in two desirable effects:

1. The error term (ϵ_{ijk} in Equation 1 above) is reduced, resulting in a more powerful statistical test. Variance in the dependent variable that can be accounted for by scores on the covariate is removed from the error term.

2. The estimate of the slope of the relationship between the covariate and the dependent variable (β in Equation 1 above) allows you to numerically adjust scores on the dependent variable, and ask the (often useful) question: *What would scores have been on the dependent variable IF the groups had been equal on the covariate?*

To evaluate the effect of the treatment factor we would compare the following restricted and full models:

$$\begin{aligned} full: \quad Y_{ijk} &= \mu + \beta X_{ij} + \alpha_j + \epsilon_{ijk} \qquad (2) \\ restricted: \quad Y_{ijk} &= \mu + \beta X_{ij} + \epsilon_{ijk} \qquad (3) \end{aligned}$$

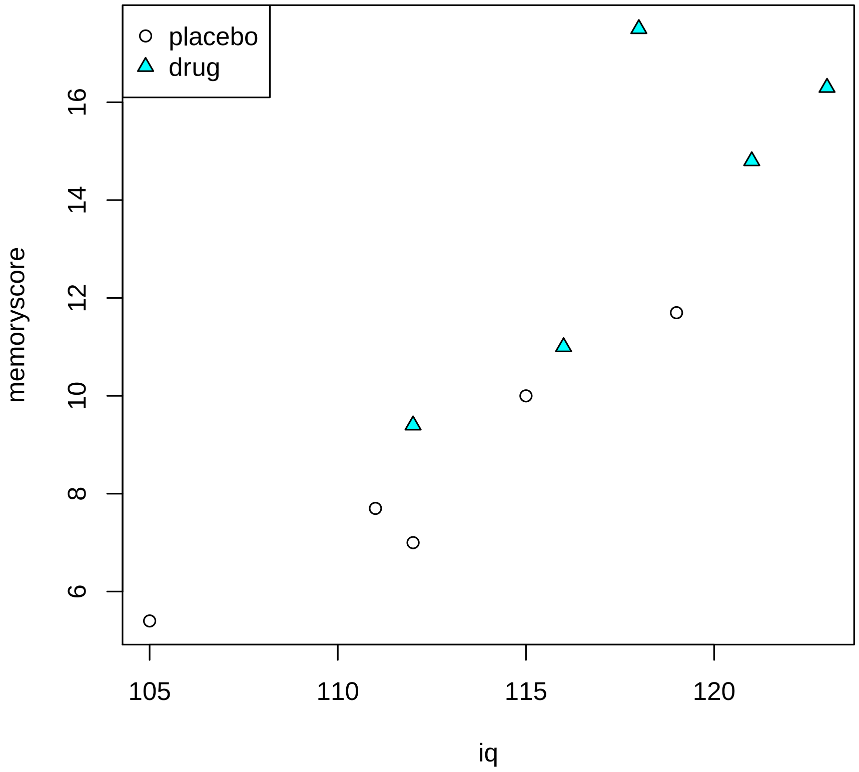
Notice how the covariate term βX_{ij} is present in both models. We are asking: is there an effect of the treatment factor (is the α_j parameter zero) after we have removed the variance accounted for by the covariate?

Notice also how in the equation for the full model (Equation 1) the α_j and β parameters can be regarded as an intercept and slope, respectively, of the straight line relating the covariate X to the dependent variable Y. Let's look at this graphically.

Imagine we have data from 10 subjects, 5 in each of two groups (e.g. control and experimental). We have scores on the dependent variable Y, and we also have, for each subject, scores on a covariate X. Let's say for the sake of an example, we're testing whether a new drug is effective at boosting memory. The two groups are placebo and drug, and the dependent variable is number of items recalled on a memory test. Let's say we know from previous research that IQ scores are predictive (are correlated with) memory, and so we also have IQ scores for each of our 10 subjects.

We can plot the data in a particular way to illustrate the principles of ANCOVA. We will plot a point for each subject, with their score on the dependent variable on the vertical axis, their score on the covariate on the horizontal axis, and we will use different plot symbols to denote membership in the placebo versus drug group.

```
> memoryscore <- c(7.0, 7.7, 5.4, 10.0, 11.7, 11.0, 9.4, 14.8, 17.5, 16.3)
> group <- factor(c(rep("placebo", 5), rep("drug", 5)),
+         levels=c("placebo", "drug"))
> iq <- c(112,111,105,115,119,116,112,121,118,123)
> plot(iq,memoryscore, pch=c(rep(1,5),rep(24,5)), bg=c(rep(1,5),rep(21,5)))
> legend("topleft", c("placebo","drug"), pch=c(1,24), pt.bg=c(1,21))
```



We can see three important things:

1. There seems to be a difference between groups on the dependent variable. Mean memory score for the placebo group is 8.4 and for the drug group is 13.8.
2. There is clearly a linear relationship between memory score and IQ.
3. The two groups also seem to differ on their IQ scores. Mean IQ for the placebo group is 112.4 and for the drug group is 118.

If we were to run a simple one-way between subjects ANOVA, ignoring the covariate, we would get the following:

```
> options(contrasts = c("contr.sum", "contr.poly"))
> mod.1 <- lm(memoryscore ~ group)
> a <- anova(mod.1)
> pval.1 <- a$`Pr(>F)`[1]
> mse.1 <- a$`Mean Sq`[2]
> print(a)
```

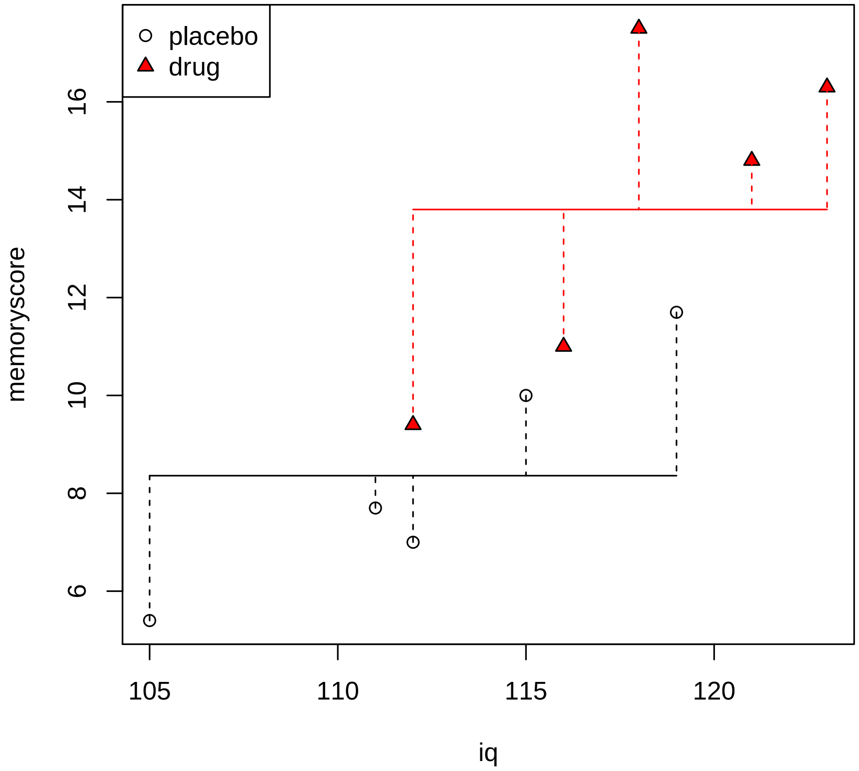
```
Analysis of Variance Table

Response: memoryscore
          Df Sum Sq Mean Sq F value  Pr(>F)
group      1 73.984  73.984  8.1043 0.02158 *
Residuals  8 73.032   9.129
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We would see that the omnibus test of the effect of group is significant at the 0.05 level (p = 0.022). Make a note of the within-group variance. We can see that the error term in the ANOVA output (the `Mean Sq` value for the `Residuals`) is 9.129.

We can visualize it this way (see below): the horizontal line is the mean memory score of each group. The vertical dashed lines represent the error, under the full model.

```
> plot(iq,memoryscore, pch=c(rep(1,5),rep(24,5)), bg=c(rep(1,5),rep("red",5)))
> legend("topleft", c("placebo","drug"), pch=c(1,24), pt.bg=c(1,"red"))
> for (j in 1:2) {
+         if (j==1) {
+                 w <- c(1,2,3,4,5)
+                 cl <- 1
+         }
+         if (j==2) {
+                 w <- c(6,7,8,9,10)
+                 cl <- "red"
+         }
+         lines(c(min(iq[w]),max(iq[w])), c(mean(memoryscore[w]),
+                 mean(memoryscore[w])), col=cl)
+         for (i in 1:length(w)) {
```

```
+                    lines(c(iq[w[i]],iq[w[i]]), c(memoryscore[w[i]],
+                             mean(memoryscore[w])), lty=2, col=cl)
+           }
+ }
```



The idea of ANCOVA then is to account for, or represent, the variation in memory score that can be captured by variation in IQ. This is done by modeling the relationship between memory score and IQ as a straight line. We can run the ANCOVA in the following way (see code snippet below), which is sort of neat, because we can explicitly state the full and restricted models as linear models using the `lm()` function, and then perform an F-test using the `anova()` function to compare each model. I also show below that you can just use the `anova()` function on the full model alone, as well.

```
> mod.full <- lm(memoryscore ~ iq + group)
> mod.rest <- lm(memoryscore ~ iq)
> a2 <- anova(mod.full, mod.rest)
> print(a2)

Analysis of Variance Table
```

```
Model 1: memoryscore ~ iq + group
Model 2: memoryscore ~ iq
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1      7 20.913
2      8 31.273 -1    -10.36 3.4678 0.1049
```

```
> a3 <- anova(mod.full)
> print(a3)
```

```
Analysis of Variance Table

Response: memoryscore
          Df  Sum Sq Mean Sq F value    Pr(>F)
iq         1 115.743 115.743 38.7412 0.0004349 ***
group      1  10.360  10.360  3.4678 0.1048703
Residuals  7  20.913   2.988
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> pval.2 <- a3$`Pr(>F)`[2]
> mse.2 <- a3$`Mean Sq`[3]
```

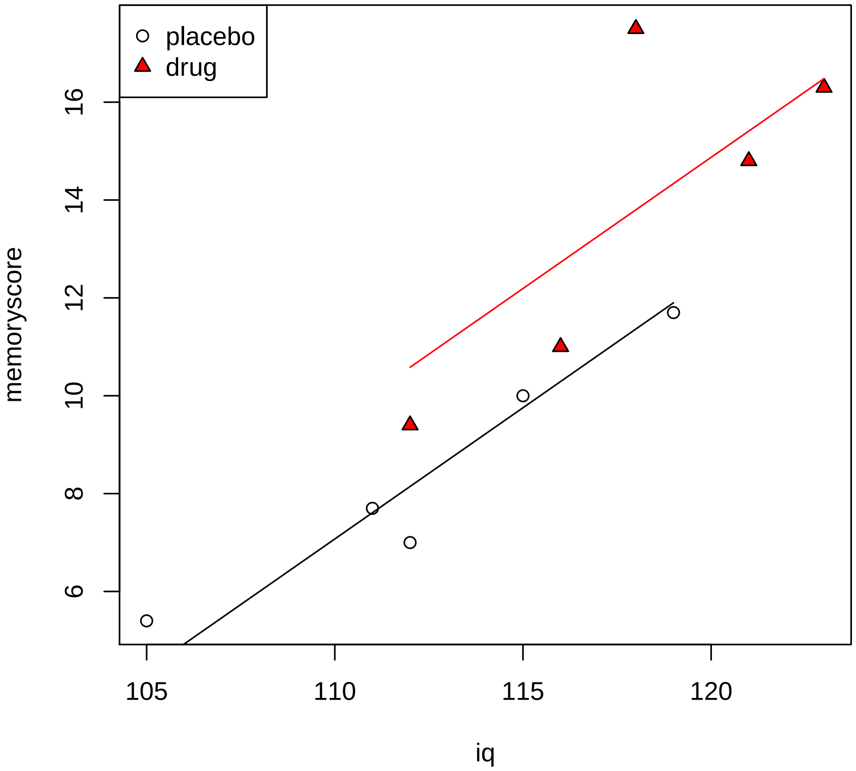
Notice how much lower our error term is (`Mean Sq Residuals =` 2.988) than before when we ran the ANOVA without the covariate. Also notice that the effect of group is no longer significant! For the test of the effect of group, p = 0.105.

Graphically we can represent the ANCOVA full model by plotting the line corresponding to each pair of (α_j, β) for each group. Remember α_j is the intercept and β is the (common) slope.

```
> plot(iq,memoryscore, pch=c(rep(1,5),rep(24,5)), bg=c(rep(1,5),rep("red",5)))
> legend("topleft", c("placebo","drug"), pch=c(1,24), pt.bg=c(1,"red"))
> mod.full.coef <- coef(mod.full)
> print(mod.full.coef)
```

```
(Intercept)          iq      group1
-50.7033113   0.5363135  -1.2183223
```

```
> placebo.pred <- predict(mod.full)
> for (j in 1:2) {
+         if (j==1) {
+                 w <- c(1,2,3,4,5)
+                 cl <- 1
+         }
+         if (j==2) {
+                 w <- c(6,7,8,9,10)
+                 cl <- "red"
+         }
+         lines(c(min(iq[w]),max(iq[w])),
+                 c(min(placebo.pred[w]), max(placebo.pred[w])), col=cl)
+ }
```

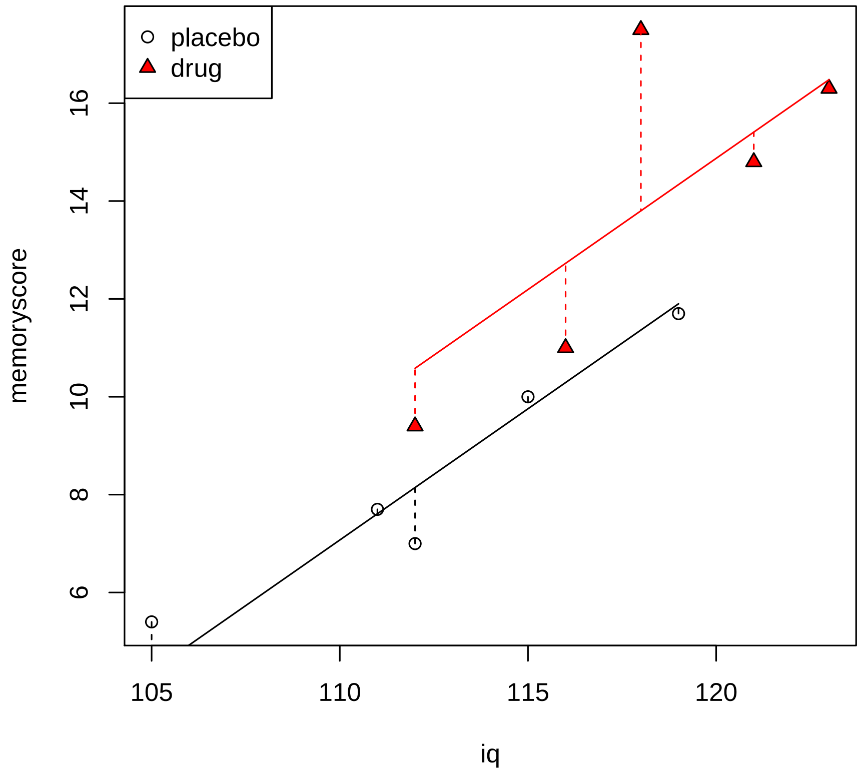


The `coef()` function gives us the model coefficients for `mod.full`, which is our full model (the output of the `lm()` function applied to our full model formula above). The `iq` value (0.536) is the estimate of the slope of the lines, β in our equations above. The `Intercept` value (-50.703) is the value of α_1, the intercept for the *first* level of the `group` factor, i.e. *placebo*. This is tricky and is an **R** gotcha: the `group1` value (-1.218) is *not* equal to α_2, but is what you have to *add* to α_1 in order to get the estimate of α_2. Thus α_2 = -50.703 + 0.536 = -50.167.

Now, the test for the effect of group, is essentially a test of whether a full model that contains *separate* intercepts, $\mu + \alpha_j$, one for each level of the group factor (placebo, drug), is significantly better than a restricted model that only contains *one* intercept (in this case, since α_j is missing from the model, so the intercept is simply μ and since it doesn't have a subscript j, is the same for each level of the group factor).

In the example above, the intercepts of the two lines corresponding to the full model (the values of α_1 and α_2) are similar enough to each other that the restricted model probably fits the data just as well (indeed it does, according to the F-test we performed above).

```
> plot(iq,memoryscore, pch=c(rep(1,5),rep(24,5)), bg=c(rep(1,5),rep("red",5)))
> legend("topleft", c("placebo","drug"), pch=c(1,24), pt.bg=c(1,"red"))
> for (j in 1:2) {
+         if (j==1) {
+                 w <- c(1,2,3,4,5)
+                 cl <- 1
+         }
+         if (j==2) {
+                 w <- c(6,7,8,9,10)
+                 cl <- "red"
+         }
+         lines(c(min(iq[w]),max(iq[w])),
+                 c(min(placebo.pred[w]), max(placebo.pred[w])), col=cl)
+         for (i in 1:length(w)) {
+                 lines(c(iq[w[i]],iq[w[i]]),
+                         c(memoryscore[w[i]], placebo.pred[w[i]]), lty=2, col=cl)
+         }
+ }
```



Adjusted Means

Having run an ANCOVA, and having estimated the relationship between IQ and memory score, we are now in a position to answer the interesting question, *What would memory scores have been for placebo and drug groups IF their scores on IQ had been similar?*. Of course their scores on IQ were *not* similar, but ANCOVA allows us to answer the question, *IF* they had been similar, then taking account what we know about how memory score varies with IQ (independent of any experimental manipulation, like placebo vs drug), what would their scores have been?

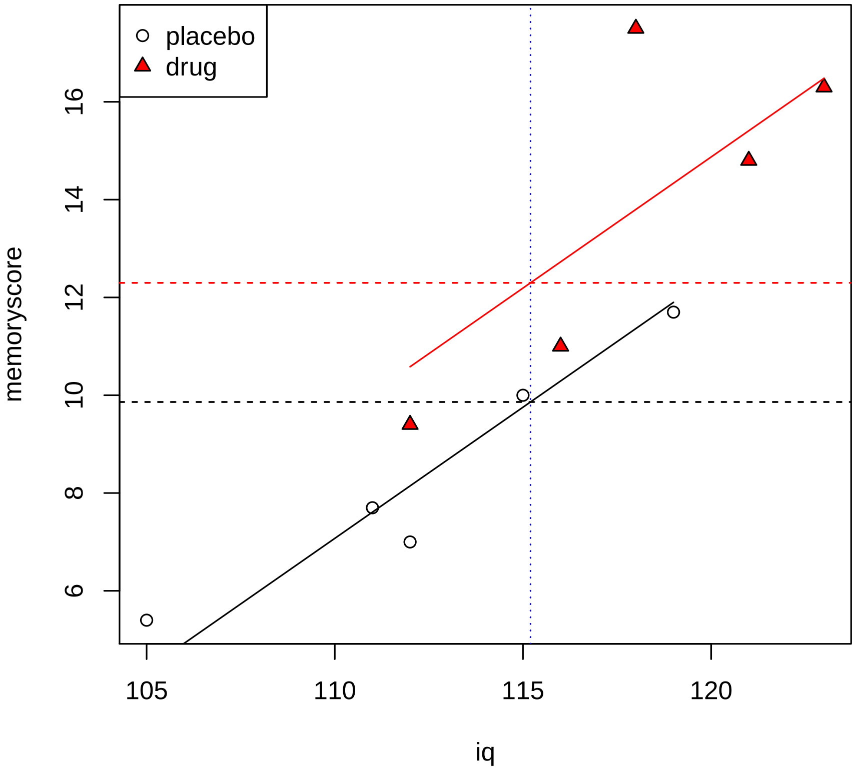
We are going to ask **R** to generate predicted values of memory score given the full model that includes both α_j and β parameters. We will then ask **R** to generate predictions of what memory score would have been, for both placebo and drug groups, *if* their scores on IQ had been the same. We will have to choose some arbitrary value of IQ, the usual is to simply choose the grand mean of IQ across both groups. We will use the `predict()` command to generate predictions from our full model.

```
> group.predict <- factor(c("placebo","drug"), levels=c("placebo","drug"))
> iq.predict <- rep(mean(iq), 2)
> data.predict <- data.frame(group=group.predict, iq=iq.predict)
> print(data.predict)

    group    iq
1 placebo 115.2
2    drug 115.2

> adjmeans <- predict(mod.full, data.predict)
> data.predict = data.frame(data.predict, memoryscore=adjmeans)
> print(data.predict)

    group    iq memoryscore
1 placebo 115.2    9.861678
2    drug 115.2   12.298322
```

So we see that our full model predicts that if the placebo and drug groups had been equal on IQ (for both groups, IQ = 115.2), then the mean memory score for the placebo group would have been 9.86 and for the drug group would have been 12.3. Compare that to the original unadjusted means (placebo = 8.36 and drug = 13.8) and you see that in this case, arithmetically adjusting for the group differences in IQ, resulted in *less* of a difference in memory score.

Graphically we can see how the prediction works, by drawing a vertical line at IQ = 115.2 and determining the value of memoryscore that intercepts that line for each group.

```
> plot(iq,memoryscore, pch=c(rep(1,5),rep(24,5)), bg=c(rep(1,5),rep("red",5)))
> legend("topleft", c("placebo","drug"), pch=c(1,24), pt.bg=c(1,"red"))
> for (j in 1:2) {
+         if (j==1) {
+                 w <- c(1,2,3,4,5)
+                 cl <- 1
+         }
+         if (j==2) {
+                 w <- c(6,7,8,9,10)
+                 cl <- "red"
+         }
+         lines(c(min(iq[w]),max(iq[w])),
+                 c(min(placebo.pred[w]), max(placebo.pred[w])), col=cl)
+         abline(v=mean(iq), lty=3, col="blue")
+         abline(h=adjmeans[1], lty=2, col="black")
+         abline(h=adjmeans[2], lty=2, col="red")
+ }
```



By including a covariate, you can change your conclusions in many ways (or fail to change them at all). An apparent effect can become weaker, or reversed, or be nullified altogether. An effect can become much stronger. Think about the conditions that must exist in your dataset for these different kinds of scenarios to happen.

ANCOVA Assumptions

Scores on the dependent variable must be normally distributed, and they must be so at all values of the covariate.

The *homogeneity of regression* assumption is that the slope of the relationship between the dependent variable and the covariate is the same for all groups. In the equation for our full model this is apparent as β does not have a subscript (indicating different values for each group, i.e. each level of the factor).

Having said that, there is nothing stopping you from fitting a full model that has different slopes for each group. It would look like this:

$$Y_{ijk} = \mu + \beta_j X_{ij} + \alpha_j + \epsilon_{ijk} \tag{4}$$

The cost is, you lose a degree of freedom for every extra slope you have to estimate. In addition however, it raises the troublesome question, *why* do different groups have different relationships between the dependent variable and the covariate? This may not be a problem. Then again it may be. It depends on the situation and the meaning of the dependent variable, the covariate, and the experimental design.

You can perform a test to see if the slopes are different very easily, by comparing the following two models:

$$\begin{aligned} full: \quad Y_{ijk} &= \mu + \beta_j X_{ij} + \alpha_j + \epsilon_{ijk} \qquad (5) \\ restricted: \quad Y_{ijk} &= \mu + \beta X_{ij} + \alpha_j + \epsilon_{ijk} \qquad (6) \end{aligned}$$

In **R** you can simply compare a model that includes an interaction term with one that does not. In other words, in the full model, the effect of IQ on memory score (the slope) varies depending on the level of group (this is the definition of an interaction). In the restricted model, it does not.

```
> mod.F <- lm(memoryscore ~ iq + group + iq:group)
> mod.R <- lm(memoryscore ~ iq + group)
> anova(mod.F, mod.R)

Analysis of Variance Table

Model 1: memoryscore ~ iq + group + iq:group
Model 2: memoryscore ~ iq + group
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1      6 19.519
2      7 20.913 -1   -1.3943 0.4286  0.537
```

Or we could simply list the anova table for the full model:

```
> anova(mod.F)
```

```
Analysis of Variance Table

Response: memoryscore
          Df  Sum Sq Mean Sq F value    Pr(>F)
iq         1 115.743 115.743 35.5788 0.0009948 ***
group      1  10.360  10.360  3.1848 0.1245828
iq:group   1   1.394   1.394  0.4286 0.5369540
Residuals  6  19.519   3.253
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

What is the Meaning of Adjusted Means?

Although ANCOVA allows you to arithmetically adjust scores to predict what they would be *if* groups were equal on a covariate, if the fact of the matter is, in your experiment, they are *not* equal on the covariate, then it may be difficult to interpret the adjusted scores, in particular when your experiment involves a treatment of some kind. For example, in the fake data we have presented here on memory, drug vs placebo, and IQ, what if the drug in question actually has different effects on memory in low IQ versus high IQ individuals? The only way to truly deal with this problem is to equate the subjects on the covariate in the first place, prior to the treatment. You also may want to consider turning your covariate into a real factor (e.g. crossing your design with levels of IQ, e.g. low, mid, high).

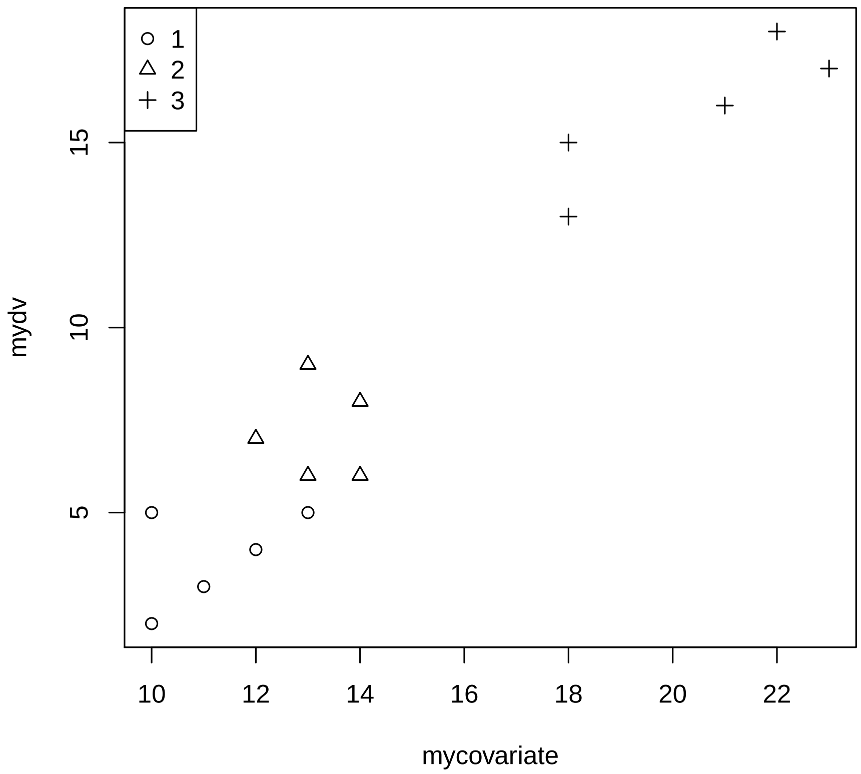
Followup tests on individual means

After a significant omnibus test on the ANCOVA, you may wish to perform follow-up tests on individual means. Tests are done on *adjusted* means. Unfortunately there is no button nor is there a simple **R** command to perform followup tests after an ANCOVA.

One way of doing followup tests, and in many ways the most flexible, is to compute an F-ratio as described in the Maxwell & Delaney text. In my hardcover 2nd edition — for pairwise comparisons, pg. 435, equations 37 and 38; and for any arbitrary contrasts: pg. 436, equations 39, 41 and 42. The denominator of the F-ratio is no longer just read off of the ANOVA output table as the mean-squared error, it's mean-squared error multiplied by a rather ugly looking term that you need to compute. It wouldn't be a big deal for you to write an **R** function to compute the denominator. Note that these are tests uncorrected for Type-I error. You can do Tukey corrections by converting the F-value you get to a q using $q = \sqrt{2F}$, and then compute a probability based on the q distribution using $df_{num} = 1$ and $df_{denom} = N - a - 1$ where a is the number of groups.

Below is an example of how you would do this, for some made-up data. First, here are the fake data:

```
> mydv <- c(2,3,5,4,5,6,6,8,9,7,13,15,16,17,18)
> mygroup <- c(1,1,1,1,1,2,2,2,2,2,3,3,3,3,3)
> mygroupf <- factor(mygroup)
> mycovariate <- c(10,11,13,12,10,13,14,14,13,12,18,18,21,23,22)
> rawmeans <- tapply(mydv, list(mygroupf), mean)
> print(rawmeans)

   1    2    3
 3.8  7.2 15.8

> plot(mycovariate, mydv, pch=mygroup)
> legend("topleft", legend=levels(mygroupf), pch=unique(mygroup))
```



Now let's run the ANCOVA model.

```
> mymod <- lm(mydv ~ mycovariate + mygroupf)
> mymod.anova <- anova(mymod)
> print(mymod.anova)

Analysis of Variance Table

Response: mydv
            Df Sum Sq Mean Sq  F value    Pr(>F)
mycovariate  1 379.14  379.14 239.5787 8.186e-09 ***
```

```
mygroupf      2  14.39    7.19   4.5452   0.03641 *
Residuals    11  17.41    1.58
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Now let’s calculate the adjusted means.

```
> # generate adjusted means
> mygroup.predict <- factor(c(1,2,3))
> mycovariate.predict <- rep(mean(mycovariate), 3)
> mydata.predict <- data.frame(mygroupf=mygroup.predict,
+                  mycovariate=mycovariate.predict)
> print(mydata.predict)

  mygroupf mycovariate
1        1    14.93333
2        2    14.93333
3        3    14.93333

> adjmeans <- predict(mymod, mydata.predict)
> print(adjmeans)

        1         2         3
 6.030303  8.235498 12.534199
```

Now let’s do a contrast comparing adjusted means 1 and 2.

```
> # compare adjusted mean 1 and adjusted mean 2
> Y1 <- adjmeans[1]        # adjusted mean 1
> Y2 <- adjmeans[2]        # adjusted mean 2
> a <- length(levels(mygroupf))        # total number of groups
> N <- length(mydv)        # total number of subjects in whole experiment
> n <- N/a # number of subjects in each group
> mse <- mymod.anova$`Mean Sq`[3]        # error term from the ANOVA output table
> n1 <- n
> n2 <- n
> # the extra junk in equation 37, Maxwell & Delaney
> i1 <- which(mygroup==1)
> i2 <- which(mygroup==2)
> i3 <- which(mygroup==3)
> sse1 <- sum((mydv[i1]-mean(mydv[i1]))^2)
> sse2 <- sum((mydv[i2]-mean(mydv[i2]))^2)
> sse3 <- sum((mydv[i3]-mean(mydv[i3]))^2)
> denom_mult <- (1/n1) + (1/n2) + (((Y1-Y2)^2)/(sse1+sse2+sse3))
> s2y1y2 <- mse * denom_mult
> dfnum <- 1
> dfdenom <- N-a-1
> Fobs <- ((Y1-Y2)^2)/s2y1y2
> p.value.uncorrected = pf(Fobs, dfnum, dfdenom, lower.tail=FALSE)
> print(p.value.uncorrected)

         1
0.04062419

> qobs = sqrt(2*Fobs)
> p.value.tukey = ptukey(q=qobs, nmeans=a, df=dfdenom, lower.tail=FALSE)
> print(p.value.tukey)

         1
0.09477965
```

Another possibility is to use the `multcomp` package in `R` (which you will have to download and install). It can perform pairwise Tukey tests on adjusted means, in the following way. Note how the results are presented as a t-test, and the probability is not exactly the same as when using the Maxwell & Delaney equations. I don't yet know why this is, I am investigating it. My guess is that the internals of `glht()` are doing something slightly different than what Maxwell & Delaney suggest.

```
> library(multcomp)
> summary(glht(aov(mydv ~ mycovariate + mygroupf), linfct=mcp(mygroupf="Tukey")))

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = mydv ~ mycovariate + mygroupf)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0   2.2052     0.9157   2.408   0.0714 .
3 - 1 == 0   6.5039     2.2320   2.914   0.0300 *
3 - 2 == 0   4.2987     1.8157   2.368   0.0762 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```