

# Analysis of Covariance in R

Philip Dixon

3/23/23

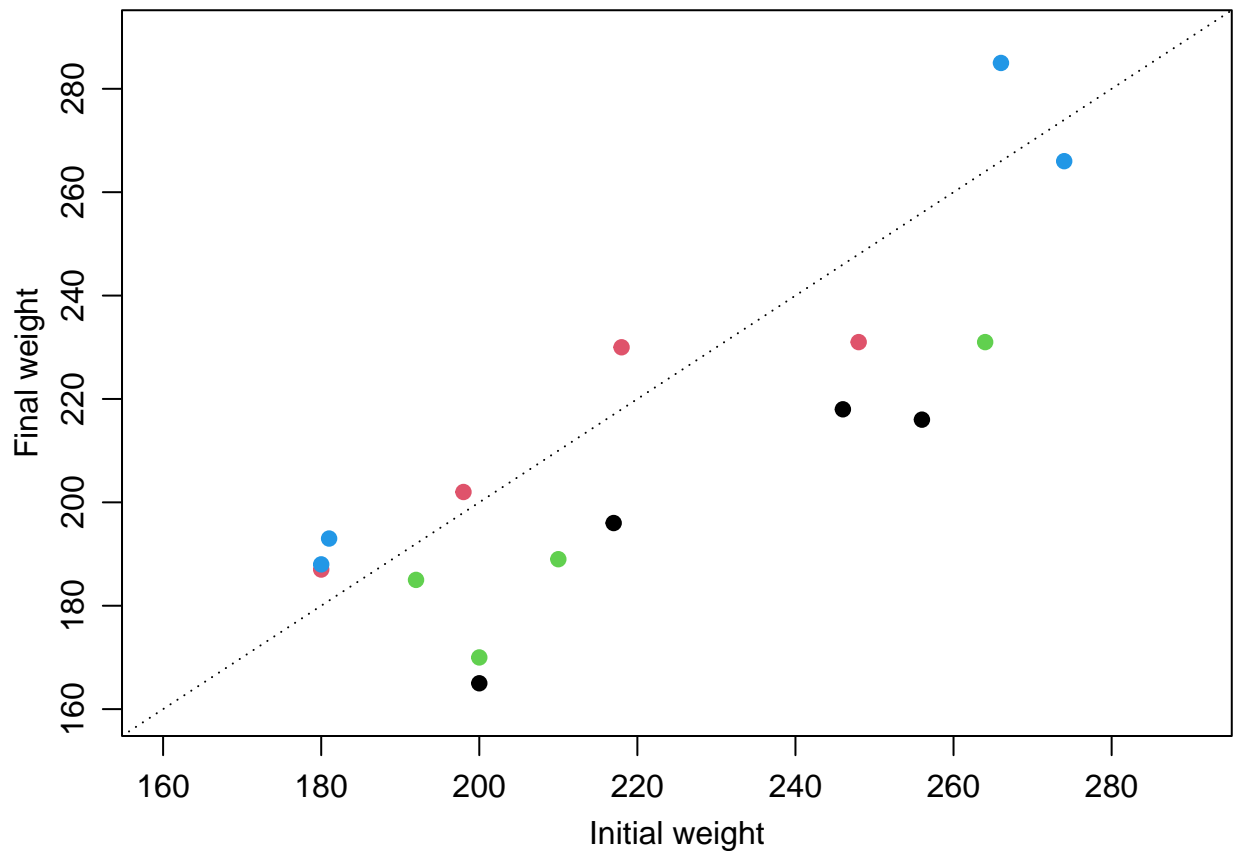
## Analysis of the deLury toxicology data using ANCOVA and related methods

The data are in delury.xlsx. Load libraries, read the data, and define drug as a factor.

```
library(readxl)
library(emmeans)

rat <- read_excel('../..//data/delury.xlsx')
rat$trt.f <- factor(rat$treatment)
```

Look at the data, with colors indicating different treatments. Code in delury.r



**ANCOVA** Fit the ANCOVA model using PRE as the covariate (same slope for all drugs). Test equality of drugs, after adjusting to same value of PRE.

```
rat.lm <- lm(final ~ initial + trt.f, data=rat)
anova(rat.lm)
```

```
## Analysis of Variance Table
##
## Response: final
##           Df Sum Sq Mean Sq F value    Pr(>F)
## initial    1 11237.2 11237.2 106.033 5.513e-07 ***
## trt.f       3  3989.0  1329.7  12.547 0.0007138 ***
## Residuals 11  1165.8   106.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q to think about: If you only want a test of drugs (trt.f) at the same initial value, why is anova() reasonable here?

My answer, because trt.f is last in the model, so type I and type III tests are identical.

Estimate adjusted means. Test equality of adjusted means. Type III test using joint\_tests() gives same result as the ANOVA.

```
rat.emm <- emmeans(rat.lm, 'trt.f')
rat.emm
```

```
## trt.f emmean SE df lower.CL upper.CL
## 1      191 5.20 11      179      202
## 2      221 5.21 11      209      232
## 3      197 5.16 11      186      209
## 4      229 5.16 11      218      240
##
## Confidence level used: 0.95
```

```
# or summary(rat.emm, infer=c(T,F))
joint_tests(rat.emm)
```

```
## model term df1 df2 F.ratio p.value
## trt.f      3  11 12.547 0.0007
```

Can follow up with contrasts, pairwise comparisons, or other “after the ANOVA” analyses, just as if there were no covariate.

By default, the emmeans are for initial = overall average. Here that’s initial = 220.625. You can change that by add at=list( variable = value, variable = value) to the emmeans() call.

```
mean(rat$initial)
```

```
## [1] 220.625
```

```
rat.emm <- emmeans(rat.lm, 'trt.f',
  at=list(initial = 220.625) )
# same as default
rat.emm
```

```
## trt.f emmean SE df lower.CL upper.CL
## 1      191 5.20 11      179      202
## 2      221 5.21 11      209      232
## 3      197 5.16 11      186      209
## 4      229 5.16 11      218      240
##
## Confidence level used: 0.95
```

```

rat.emm2 <- emmeans(rat.lm, 'trt.f',
  at=list(initial = 180) )
rat.emm2

## trt.f emmean SE df lower.CL upper.CL
## 1 156 6.57 11 142 171
## 2 186 5.74 11 173 199
## 3 163 5.95 11 149 176
## 4 194 6.35 11 180 208
##
## Confidence level used: 0.95

```

```
joint_tests(rat.emm2)
```

```

## model term df1 df2 F.ratio p.value
## trt.f 3 11 12.547 0.0007

```

Because the lines are parallel, the test of drug effects gives the same result for any choice of covariate value. Notice the test result at initial=180 is the same as the default, for initial = 220.625.

**Heterogeneous regression lines** You specify different slopes as an interaction between the covariate and the treatment factor.

```
rat.lm2 <- lm(final ~ initial + trt.f + initial:trt.f, data=rat)
```

The easiest way to get a type III test of drug effects is to use emmeans. anova() no longer gives you that, even if you try to write the model as initial + initial:trt.f + trt.f, to put the factor of interest last. R will reorder the terms to put the interaction after the component main effects. anova() does give you the type III test of initial:trt.f, i.e. equality of slopes.

```
rat.emm2 <- emmeans(rat.lm2, 'trt.f')
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
rat.emm2
```

```

## trt.f emmean SE df lower.CL upper.CL
## 1 191 6.04 8 177 204
## 2 219 5.99 8 205 233
## 3 197 5.65 8 184 210
## 4 229 5.62 8 216 242
##
## Confidence level used: 0.95

```

```
anova(rat.lm2)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: final
```

```

##          Df Sum Sq Mean Sq F value    Pr(>F)
## initial    1 11237.2 11237.2  89.8295 1.265e-05 ***
## trt.f       3  3989.0  1329.7  10.6294 0.003648 **
## initial:trt.f 3   165.0    55.0   0.4397 0.730901
## Residuals  8  1000.8   125.1

```

```
## ---
```

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

```
joint_tests(rat.emm2)
```

```
## model term df1 df2 F.ratio p.value
## trt.f      3   8   9.672 0.0049
```

You can explicitly see what value of initial is being used to compute these emmeans by adding that variable to the reference grid. Note that emmeans will round the value when reporting the table. The actual value is 220.625.

```
emmeans(rat.lm2, c('trt.f', 'initial'))
```

```
## trt.f initial emmean SE df lower.CL upper.CL
## 1      221    191 6.04 8     177     204
## 2      221    219 5.99 8     205     233
## 3      221    197 5.65 8     184     210
## 4      221    229 5.62 8     216     242
##
## Confidence level used: 0.95
```

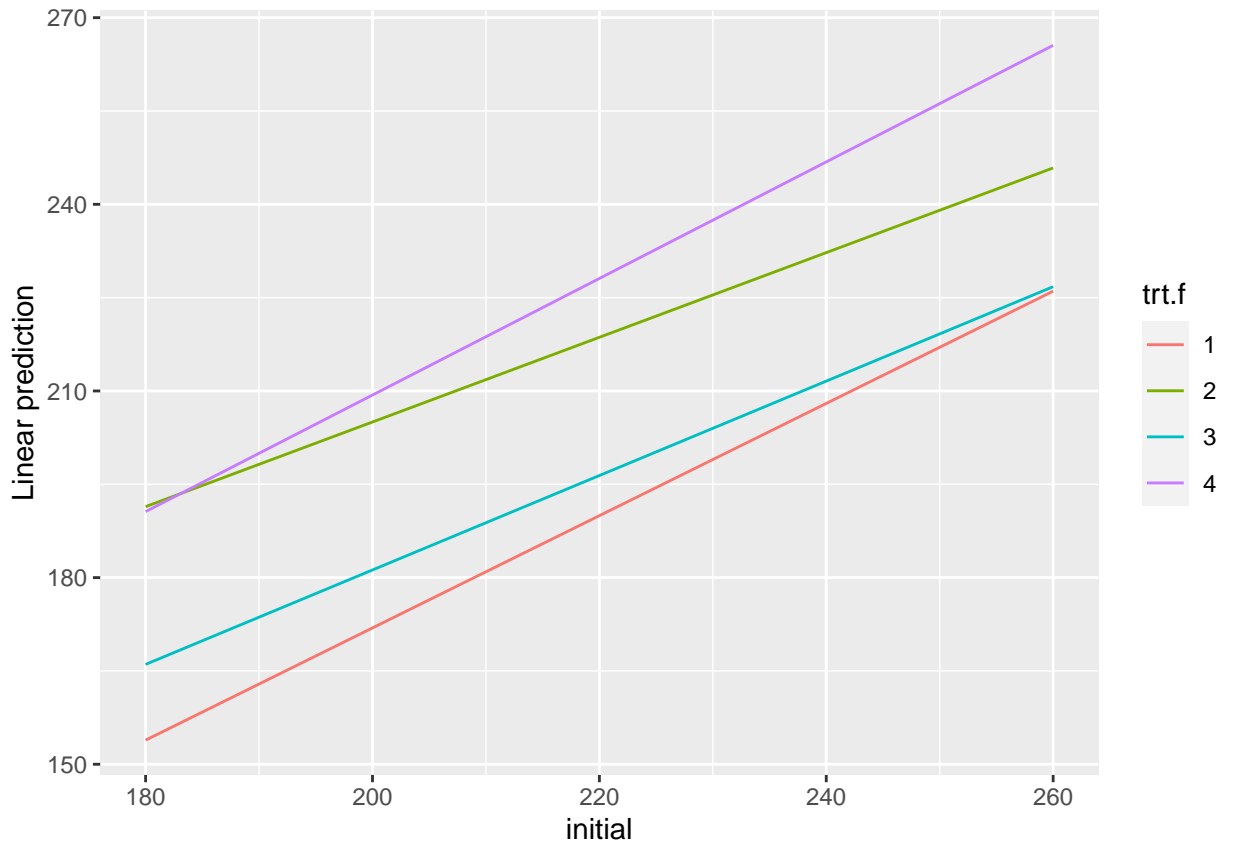
As before, you specify the desired value of the covariate by adding at=. If you give a vector of values, you get predictions at each of the specified values.

```
rat.emm3 <- emmeans(rat.lm2, c('trt.f', 'initial'), at=list(initial=c(180,200, 220, 240, 260)))
rat.emm3
```

```
## trt.f initial emmean SE df lower.CL upper.CL
## 1      180    154 13.64 8     122     185
## 2      180    191  8.86 8     171     212
## 3      180    166  9.16 8     145     187
## 4      180    191  7.94 8     172     209
## 1      200    172  9.31 8     150     193
## 2      200    205  6.10 8     191     219
## 3      200    181  6.48 8     166     196
## 4      200    209  6.42 8     195     224
## 1      220    190  6.10 8     176     204
## 2      220    219  5.94 8     205     232
## 3      220    196  5.64 8     183     209
## 4      220    228  5.63 8     215     241
## 1      240    208  6.15 8     194     222
## 2      240    232  8.52 8     213     252
## 3      240    212  7.28 8     195     228
## 4      240    247  5.89 8     233     260
## 1      260    226  9.41 8     204     248
## 2      260    246 12.21 8     218     274
## 3      260    227 10.29 8     203     250
## 4      260    266  7.08 8     249     282
##
## Confidence level used: 0.95
```

This set of combinations of initial values and treatment levels is especially useful if you want to plot the curves or test equality of drugs at each of the specified initial values:

```
emmip(rat.emm3, trt.f ~ initial)
```



het-1.pdf

```
joint_tests(rat.emm3, by='initial')
```

```
## initial = 180:
## model term df1 df2 F.ratio p.value
## trt.f      3  8  3.144 0.0867
##
## initial = 200:
## model term df1 df2 F.ratio p.value
## trt.f      3  8  6.118 0.0182
##
## initial = 220:
## model term df1 df2 F.ratio p.value
## trt.f      3  8  9.615 0.0050
##
## initial = 240:
## model term df1 df2 F.ratio p.value
## trt.f      3  8  8.493 0.0072
##
## initial = 260:
## model term df1 df2 F.ratio p.value
## trt.f      3  8  5.209 0.0276
```

To estimate the slopes, use the `emtrends()` function:

```
rat.emm4 <- emtrends(rat.lm2, 'trt.f', var='initial')
rat.emm4
```

```
## trt.f initial.trend SE df lower.CL upper.CL
```

```
## 1          0.902 0.250  8    0.325    1.48
## 2          0.681 0.222  8    0.170    1.19
## 3          0.759 0.199  8    0.301    1.22
## 4          0.937 0.125  8    0.649    1.22
##
## Confidence level used: 0.95
```

The first two arguments are the lm model and the name of the factor variable (or variables). var= specifies the name of the covariate.

In the most recent versions of emmeans, you can test equality of slopes easily once you have the slopes for each group (the emtrends result):

```
joint_tests(rat.emm4)
```

```
## model term df1 df2 F.ratio p.value
## trt.f      3   8   0.440 0.7309
```

**Fitting quadratic models** This can be done two ways, by calculating a variable containing the squared term or by specifying a quadratic effect in the lm() model. If you want to specify a computation in a model formula, you need to protect it'' by enclosing inI( )'', e.g., I(initial^2). Since R doesn't easily give type III tests of regression parameters, the best way to test a quadratic term is to do a t-test on the estimate, using summary().

```
rat$initial2 <- rat$initial^2
```

```
quad.lm1 <- lm(final ~ trt.f + initial + initial2, data=rat)
summary(quad.lm1)
```

```
##
## Call:
## lm(formula = final ~ trt.f + initial + initial2, data = rat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.609  -6.371  -1.259   6.163  15.559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.350e+02  2.692e+02  -0.873  0.40304
## trt.f2       3.127e+01  7.700e+00   4.061  0.00228 **
## trt.f3       7.961e+00  7.660e+00   1.039  0.32311
## trt.f4       4.514e+01  1.085e+01   4.161  0.00195 **
## initial      2.986e+00  2.411e+00   1.238  0.24394
## initial2    -4.733e-03  5.354e-03  -0.884  0.39743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 10 degrees of freedom
## Multiple R-squared:  0.934, Adjusted R-squared:  0.9011
## F-statistic: 28.32 on 5 and 10 DF,  p-value: 1.346e-05
```

```
quad.lm2 <- lm(final ~ trt.f + initial + I(initial^2), data=rat)
summary(quad.lm2)
```

```
##
## Call:
```

```

## lm(formula = final ~ trt.f + initial + I(initial^2), data = rat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.609  -6.371  -1.259   6.163  15.559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.350e+02  2.692e+02  -0.873  0.40304
## trt.f2       3.127e+01  7.700e+00   4.061  0.00228 **
## trt.f3       7.961e+00  7.660e+00   1.039  0.32311
## trt.f4       4.514e+01  1.085e+01   4.161  0.00195 **
## initial      2.986e+00  2.411e+00   1.238  0.24394
## I(initial^2) -4.733e-03  5.354e-03  -0.884  0.39743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 10 degrees of freedom
## Multiple R-squared:  0.934, Adjusted R-squared:  0.9011
## F-statistic: 28.32 on 5 and 10 DF,  p-value: 1.346e-05

```

**Differences** Compute the difference then use ANOVA methods from previous weeks.

```

rat$diff <- rat$final - rat$initial
rat.lmdiff <- lm(diff ~ trt.f, data=rat)

```