

Lecture 11: General linear model part 2

Myostatin data analysis

We will continue with the myostatin example.

```
myostatin
##      leucine      group time
## 1      6568    control   24
## 2      6802    control   24
## 3      7198    control   24
## 4      7280    control   24
## 5      4992    control   48
## 6      5242    control   48
## 7      5285    control   48
## 8      6284    control   48
## 9      4092    control   72
## 10     4331    control   72
## 11     5135    control   72
## 12     6087    control   72
## 13     5516 myostatin   24
## 14     6023 myostatin   24
## 15     6334 myostatin   24
## 16     6400 myostatin   24
## 17     4512 myostatin   48
## 18     4706 myostatin   48
## 19     5175 myostatin   48
## 20     6612 myostatin   48
## 21     3076 myostatin   72
## 22     3209 myostatin   72
## 23     3462 myostatin   72
## 24     5364 myostatin   72

myostat.means <- aggregate(leucine/1000 ~ group*time, FUN=mean,
data=myostatin)
myostat.means
##      group time leucine/1000
## 1  control   24         6.962
## 2 myostatin   24         6.068
## 3  control   48         5.451
## 4 myostatin   48         5.251
## 5  control   72         4.911
## 6 myostatin   72         3.778
```

This code shows how to fit the various forms of the general linear model we have learned about to the myostatin data. The one-way effects model includes an intercept and

parameter for each group and time combination so is not full rank. The **means model is the one-way effects model WITHOUT the intercept.** The two-way model includes an intercept, main effects for time and group, and their interaction.

```
# one-way effects model
mod2 <- lm(leucine/1000 ~ group:time,data=myostatin)
summary(mod2)

##
## Call:
## lm(formula = leucine/1000 ~ group:time, data = myostatin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.819 -0.547 -0.163  0.279  1.586
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.778      0.380    9.95 9.6e-09
## groupcontrol:time24  3.184      0.537    5.93 1.3e-05
## groupmyostatin:time24 2.290      0.537    4.27 0.00046
## groupcontrol:time48  1.673      0.537    3.12 0.00596
## groupmyostatin:time48 1.474      0.537    2.74 0.01332
## groupcontrol:time72  1.133      0.537    2.11 0.04897
## groupmyostatin:time72      NA         NA      NA      NA
##
## Residual standard error: 0.759 on 18 degrees of freedom
## Multiple R-squared:  0.69,    Adjusted R-squared:  0.604
## F-statistic: 8.02 on 5 and 18 DF,  p-value: 0.000396

anova(mod2)

## Analysis of Variance Table
##
## Response: leucine/1000
##              Df Sum Sq Mean Sq F value Pr(>F)
## group:time    5   23.1    4.63    8.02 4e-04
## Residuals    18   10.4    0.58

# means model
mod3 <- lm(leucine/1000 ~ 0+group:time,data=myostatin)
summary(mod3)

##
## Call:
## lm(formula = leucine/1000 ~ 0 + group:time, data = myostatin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.819 -0.547 -0.163  0.279  1.586
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## groupcontrol:time24      6.96      0.38  18.34  4.3e-13
## groupmyostatin:time24     6.07      0.38  15.99  4.4e-12
## groupcontrol:time48      5.45      0.38  14.36  2.7e-11
## groupmyostatin:time48     5.25      0.38  13.83  5.0e-11
## groupcontrol:time72      4.91      0.38  12.94  1.5e-10
## groupmyostatin:time72     3.78      0.38   9.95  9.6e-09
##
## Residual standard error: 0.759 on 18 degrees of freedom
## Multiple R-squared:  0.986, Adjusted R-squared:  0.981
## F-statistic: 209 on 6 and 18 DF, p-value: 1.2e-15

anova(mod3)

## Analysis of Variance Table
##
## Response: leucine/1000
##           Df Sum Sq Mean Sq F value    Pr(>F)
## group:time  6    724   120.6    209 1.2e-15
## Residuals  18     10     0.6

# two-way model with SAS coding
myostatin.relevel <- myostatin
myostatin.relevel$group <- relevel(myostatin.relevel$group,ref='myostatin')
myostatin.relevel$time <- relevel(as.factor(myostatin.relevel$time),ref='72')
mod1 <- lm(leucine/1000 ~ group*time, data=myostatin.relevel)
summary(mod1)

##
## Call:
## lm(formula = leucine/1000 ~ group * time, data = myostatin.relevel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.819 -0.547 -0.163  0.279  1.586
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.778      0.380   9.95  9.6e-09
## groupcontrol      1.133      0.537   2.11  0.04897
## time24           2.291      0.537   4.27  0.00046
## time48           1.473      0.537   2.74  0.01332
## groupcontrol:time24 -0.240      0.759  -0.32  0.75579
## groupcontrol:time48 -0.934      0.759  -1.23  0.23443
##
## Residual standard error: 0.759 on 18 degrees of freedom
## Multiple R-squared:  0.69, Adjusted R-squared:  0.604
## F-statistic: 8.02 on 5 and 18 DF, p-value: 0.000396
```

```

# time as a linear term in the model
myostatin2 <- myostatin
myostatin2$time <- as.numeric(as.character(myostatin2$time))
mod1.lin <- lm(leucine/1000 ~ group*time, data=myostatin2)
summary(mod1.lin)

##
## Call:
## lm(formula = leucine/1000 ~ group * time, data = myostatin2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.811 -0.523 -0.193  0.389  1.580
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.82542     0.57388   13.64  1.4e-11
## groupmyostatin    -0.50250     0.81158   -0.62  0.54280
## time             -0.04272     0.01107   -3.86  0.00098
## groupmyostatin:time -0.00499     0.01565   -0.32  0.75297
##
## Residual standard error: 0.751 on 20 degrees of freedom
## Multiple R-squared:  0.663, Adjusted R-squared:  0.612
## F-statistic: 13.1 on 3 and 20 DF,  p-value: 5.83e-05

```

Suppose we want to estimate mean leucine levels in the myostatin group at 36 hours. We can do this with the linear form of time in the model because it allows for interpolation between the observed values for the time variable.

```

L <- c(1,1,36,36)
# estimate is
L %>% coef(mod1.lin)

##           [,1]
## [1,] 5.605

# standard error is
sqrt(t(L) %>% vcov(mod1.lin) %>% L)

##           [,1]
## [1,] 0.2543

# compare with
predict(mod1.lin, se.fit=TRUE, newdata=data.frame(group='myostatin', time=36))

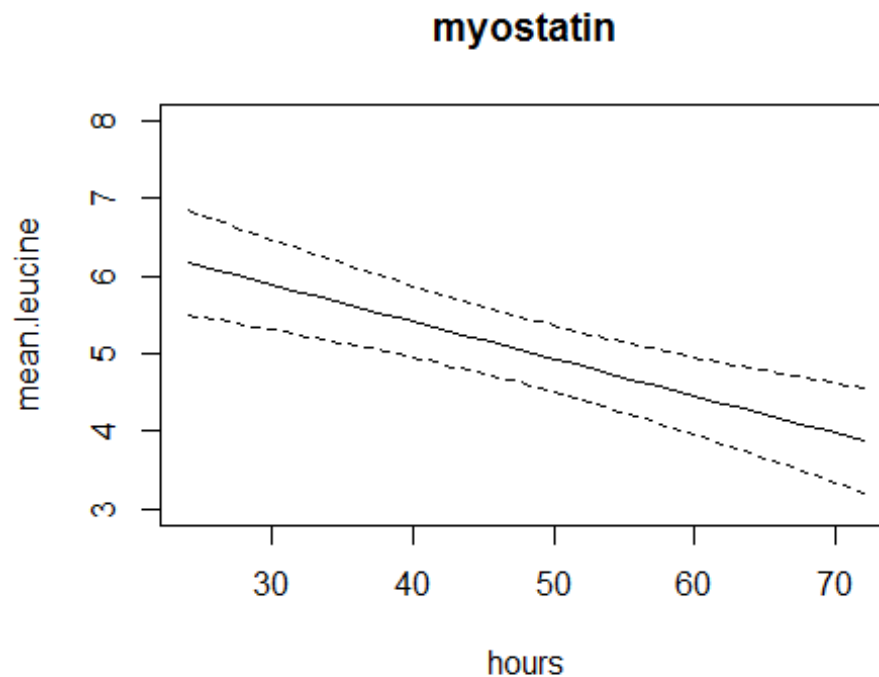
## $fit
##      1
## 5.605
##
## $se.fit
## [1] 0.2543

```

```
##
## $df
## [1] 20
##
## $residual.scale
## [1] 0.7514
```

What if we wanted to produce a curve describing this relationship?

```
hours <- 24:72
# for myostatin group
L <- as.matrix(data.frame(int=1,group=1,time=hours,group.time=hours))
mean.leucine <- L %>% coef(mod1.lin)
plot(hours,mean.leucine,type='l',ylim=c(3,8),main='myostatin')
se.mean.leucine <- sqrt(diag(L %>% vcov(mod1.lin) %>% t(L)))
lines(hours,mean.leucine + qnorm(1-.05/2)*se.mean.leucine,lty=2)
lines(hours,mean.leucine - qnorm(1-.05/2)*se.mean.leucine,lty=2)
```



```
# for control group
L <- as.matrix(data.frame(int=1,group=0,time=hours,group.time=0))
mean.leucine <- L %>% coef(mod1.lin)
plot(hours,mean.leucine,type='l',ylim=c(3,8),main='control')
se.mean.leucine <- sqrt(diag(L %>% vcov(mod1.lin) %>% t(L)))
lines(hours,mean.leucine + qnorm(1-.05/2)*se.mean.leucine,lty=2)
lines(hours,mean.leucine - qnorm(1-.05/2)*se.mean.leucine,lty=2)
```

control

