Lecture 10: General linear model

Myostatin data analysis

Mice were randomized to one of 2*3=6 possible groups in a factorial design. The treatment variable was myostatin (yes or no), and the other variable was time of measurement (24, 48 and 72 hours). The outcome was protein degradation.

```
myostatin <- data.frame(leucine=c(6568 , 6802 , 7198 , 7280 ,
4992 , 5242 , 5285 , 6284 ,
4092 , 4331 , 5135 , 6087 ,
5516 , 6023 , 6334 , 6400 ,
4512 , 4706 , 5175 , 6612 ,
3076 , 3209 , 3462 , 5364),
group=rep(c('control','myostatin'),each=12),
time=rep(rep(c(24,48,72),each=4),2))

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      6.96200
## 2 myostatin   24      6.06825
## 3   control   48      5.45075
## 4 myostatin   48      5.25125
## 5   control   72      4.91125
## 6 myostatin   72      3.77775

plot(subset(myostat.means,group=='control')[,c(2,3)],
     type='b',ylim=range(myostat.means[,3]),pch=19)
lines(subset(myostat.means,group=='myostatin')[,c(2,3)],lty=2,type='b')
legend(x='bottomleft',inset=.025,pch=c(19,21),lty=1:2,legend=c('control','myo
statin'))
```
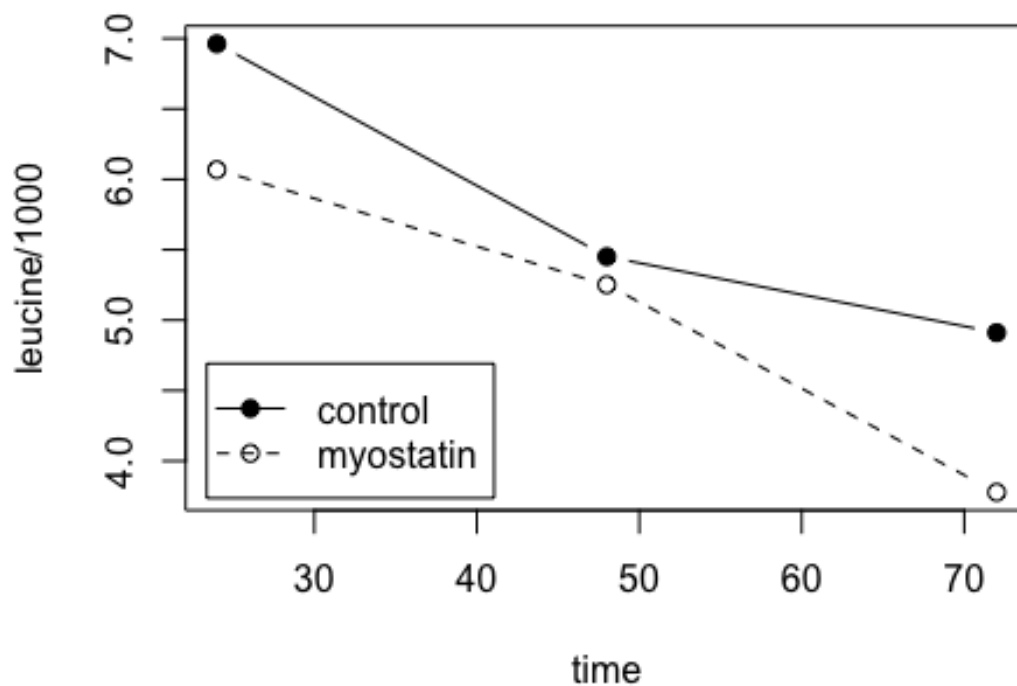


```
mod1 <- lm(leucine/1000 ~ group*as.factor(time),data=myostatin,x=TRUE,y=TRUE)
summary(mod1)

##
## Call:
## lm(formula = leucine/1000 ~ group * as.factor(time), data = myostatin,
```

```
##     x = TRUE, y = TRUE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8193 -0.5470 -0.1629  0.2788  1.5862
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          6.9620     0.3796  18.341 4.27e-13 ***
## groupmyostatin                      -0.8938     0.5368  -1.665  0.11325
## as.factor(time)48                   -1.5113     0.5368  -2.815  0.01146 *
## as.factor(time)72                   -2.0508     0.5368  -3.820  0.00125 **
## groupmyostatin:as.factor(time)48     0.6943     0.7592   0.914  0.37256
## groupmyostatin:as.factor(time)72    -0.2397     0.7592  -0.316  0.75579
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 18 degrees of freedom
## Multiple R-squared:  0.6903, Adjusted R-squared:  0.6043
## F-statistic: 8.025 on 5 and 18 DF,  p-value: 0.000396
```

```
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: leucine/1000
##                       Df  Sum Sq Mean Sq F value    Pr(>F)
## group                  1  3.3056  3.3056  5.7353   0.02772 *
## as.factor(time)        2 18.8796  9.4398 16.3782 8.872e-05 ***
## group:as.factor(time)  2  0.9412  0.4706  0.8165   0.45769
## Residuals             18 10.3745  0.5764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
X <- mod1$x
y <- mod1$y
```

```
# x-transpose x
t(X) %*% X
```

```
##                                  (Intercept) groupmyostatin
## (Intercept)                               24             12
## groupmyostatin                            12             12
## as.factor(time)48                          8              4
## as.factor(time)72                          8              4
## groupmyostatin:as.factor(time)48           4              4
## groupmyostatin:as.factor(time)72           4              4
##                                  as.factor(time)48 as.factor(time)72
## (Intercept)                                      8                 8
## groupmyostatin                                   4                 4
## as.factor(time)48                                8                 0
```

```
## as.factor(time)72                                                   0               8
## groupmyostatin:as.factor(time)48                     4               0
## groupmyostatin:as.factor(time)72                     0               4
##                                           groupmyostatin:as.factor(time)48
## (Intercept)                                                          4
## groupmyostatin                                                       4
## as.factor(time)48                                                    4
## as.factor(time)72                                                    0
## groupmyostatin:as.factor(time)48                                     4
## groupmyostatin:as.factor(time)72                                     0
##                                           groupmyostatin:as.factor(time)72
## (Intercept)                                                          4
## groupmyostatin                                                       4
## as.factor(time)48                                                    0
## as.factor(time)72                                                    4
## groupmyostatin:as.factor(time)48                                     0
## groupmyostatin:as.factor(time)72                                     4
```

```r
# inverse
solve(t(X) %*% X)
```

```
##                                           (Intercept) groupmyostatin
## (Intercept)                                      0.25          -0.25
## groupmyostatin                                  -0.25           0.50
## as.factor(time)48                               -0.25           0.25
## as.factor(time)72                               -0.25           0.25
## groupmyostatin:as.factor(time)48                 0.25          -0.50
## groupmyostatin:as.factor(time)72                 0.25          -0.50
##                                           as.factor(time)48 as.factor(time)72
## (Intercept)                                           -0.25             -0.25
## groupmyostatin                                         0.25              0.25
## as.factor(time)48                                      0.50              0.25
## as.factor(time)72                                      0.25              0.50
## groupmyostatin:as.factor(time)48                      -0.50             -0.25
## groupmyostatin:as.factor(time)72                      -0.25             -0.50
##                                           groupmyostatin:as.factor(time)48
## (Intercept)                                                          0.25
## groupmyostatin                                                      -0.50
## as.factor(time)48                                                  -0.50
## as.factor(time)72                                                  -0.25
## groupmyostatin:as.factor(time)48                                    1.00
## groupmyostatin:as.factor(time)72                                    0.50
##                                           groupmyostatin:as.factor(time)72
## (Intercept)                                                          0.25
## groupmyostatin                                                      -0.50
## as.factor(time)48                                                  -0.25
## as.factor(time)72                                                  -0.50
## groupmyostatin:as.factor(time)48                                    0.50
## groupmyostatin:as.factor(time)72                                    1.00
```

```r
# projection
P <- X %*% solve(t(X) %*% X) %*% t(X)
# fitted values
P %*% y
```

```
##        [,1]
## 1   6.96200
## 2   6.96200
## 3   6.96200
## 4   6.96200
## 5   5.45075
## 6   5.45075
## 7   5.45075
## 8   5.45075
## 9   4.91125
## 10  4.91125
## 11  4.91125
## 12  4.91125
## 13  6.06825
## 14  6.06825
## 15  6.06825
## 16  6.06825
## 17  5.25125
## 18  5.25125
## 19  5.25125
## 20  5.25125
## 21  3.77775
## 22  3.77775
## 23  3.77775
## 24  3.77775
```

```r
# from lm()
mod1$fitted.values
```

```
##       1       2       3       4       5       6       7       8       9
## 6.96200 6.96200 6.96200 6.96200 5.45075 5.45075 5.45075 5.45075 4.91125
##      10      11      12      13      14      15      16      17      18
## 4.91125 4.91125 4.91125 6.06825 6.06825 6.06825 6.06825 5.25125 5.25125
##      19      20      21      22      23      24
## 5.25125 5.25125 3.77775 3.77775 3.77775 3.77775
```

```r
# solution for beta.hat
solve(t(X) %*% X) %*% t(X) %*% y
```

```
##                                      [,1]
## (Intercept)                       6.96200
## groupmyostatin                   -0.89375
## as.factor(time)48                -1.51125
## as.factor(time)72                -2.05075
## groupmyostatin:as.factor(time)48  0.69425
## groupmyostatin:as.factor(time)72 -0.23975
```

```
# from lm()
coef(mod1)

##                     (Intercept)                       groupmyostatin
##                         6.96200                              -0.89375
##               as.factor(time)48                     as.factor(time)72
##                        -1.51125                              -2.05075
## groupmyostatin:as.factor(time)48 groupmyostatin:as.factor(time)72
##                         0.69425                              -0.23975

# two-way effects model
summary(mod1)

##
## Call:
## lm(formula = leucine/1000 ~ group * as.factor(time), data = myostatin,
##     x = TRUE, y = TRUE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8193 -0.5470 -0.1629  0.2788  1.5862
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       6.9620     0.3796  18.341 4.27e-13 ***
## groupmyostatin                   -0.8938     0.5368  -1.665  0.11325
## as.factor(time)48                -1.5113     0.5368  -2.815  0.01146 *
## as.factor(time)72                -2.0508     0.5368  -3.820  0.00125 **
## groupmyostatin:as.factor(time)48  0.6943     0.7592   0.914  0.37256
## groupmyostatin:as.factor(time)72 -0.2397     0.7592  -0.316  0.75579
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 18 degrees of freedom
## Multiple R-squared:  0.6903, Adjusted R-squared:  0.6043
## F-statistic: 8.025 on 5 and 18 DF,  p-value: 0.000396

anova(mod1)

## Analysis of Variance Table
##
## Response: leucine/1000
##                      Df  Sum Sq Mean Sq F value     Pr(>F)
## group                 1  3.3056  3.3056  5.7353    0.02772 *
## as.factor(time)       2 18.8796  9.4398 16.3782 8.872e-05 ***
## group:as.factor(time) 2  0.9412  0.4706  0.8165    0.45769
## Residuals            18 10.3745  0.5764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

##
## Call:
## lm(formula = leucine/1000 ~ group:as.factor(time), data = myostatin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8193 -0.5470 -0.1629  0.2788  1.5862
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        3.7778     0.3796   9.952 9.62e-09 ***
## groupcontrol:as.factor(time)24     3.1843     0.5368   5.932 1.30e-05 ***
## groupmyostatin:as.factor(time)24   2.2905     0.5368   4.267 0.000464 ***
## groupcontrol:as.factor(time)48     1.6730     0.5368   3.116 0.005961 **
## groupmyostatin:as.factor(time)48   1.4735     0.5368   2.745 0.013318 *
## groupcontrol:as.factor(time)72     1.1335     0.5368   2.111 0.048975 *
## groupmyostatin:as.factor(time)72       NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 18 degrees of freedom
## Multiple R-squared:  0.6903, Adjusted R-squared:  0.6043
## F-statistic: 8.025 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:as.factor(time)  5 23.126  4.6253  8.0249 0.000396 ***
## Residuals             18 10.374  0.5764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

##
## Call:
## lm(formula = leucine/1000 ~ 0 + group:as.factor(time), data = myostatin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8193 -0.5470 -0.1629  0.2788  1.5862
##
```

```
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## groupcontrol:as.factor(time)24       6.9620     0.3796  18.341 4.27e-13 ***
## groupmyostatin:as.factor(time)24     6.0682     0.3796  15.986 4.42e-12 ***
## groupcontrol:as.factor(time)48       5.4507     0.3796  14.359 2.67e-11 ***
## groupmyostatin:as.factor(time)48     5.2512     0.3796  13.834 4.95e-11 ***
## groupcontrol:as.factor(time)72       4.9112     0.3796  12.938 1.49e-10 ***
## groupmyostatin:as.factor(time)72     3.7777     0.3796   9.952 9.62e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 18 degrees of freedom
## Multiple R-squared:  0.9859, Adjusted R-squared:  0.9812
## F-statistic: 209.3 on 6 and 18 DF,  p-value: 1.203e-15

anova(mod3)

## Analysis of Variance Table
##
## Response: leucine/1000
##                      Df Sum Sq Mean Sq F value     Pr(>F)
## group:as.factor(time)  6 723.88 120.647  209.33 1.203e-15 ***
## Residuals             18  10.37   0.576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

A <- rbind(c(1,1,2),
           c(1,2,4),
           c(1,3,6))
B <- rbind(c(1,0,2),
           c(1,1,4),
           c(1,2,6))
try(solve(A))
try(solve(B))

# column 3 of A is 2*column 2
2*A[,2]

## [1] 2 4 6

A[,3]

## [1] 2 4 6

# B is a little trickier:
# column 3 = 1 + column 1 + 2*column 2
B[,1]+2*B[,2]+1

## [1] 2 4 6

# or just 2 + 2*column 2
2*B[,2]+2
```

```
## [1] 2 4 6
B[,3]
## [1] 2 4 6
```