**Example Analysis of an Unbalanced Two-Factor Experiment**

An experiment was conducted to study the effect of storage time and storage temperature on the amount of active ingredient present in a drug at the end of storage. A total of 16 vials of the drug, each containing approximately 30 mg/mL of active ingredient were assigned (using a completely randomized design) to the following treatments:

1) Storage for 3 months at $20^{o}$ C
2) Storage for 3 months at $30^{o}$ C
3) Storage for 6 months at $20^{o}$ C
4) Storage for 6 months at $30^{o}$ C

Six of the 16 vials were damaged during shipment to the laboratory where the active ingredient was measured. Accurate measures of the amount of active ingredient could be obtained only for the 10 undamaged vials. The table below shows the amount of active ingredient lost during storage (in tenths of mg/mL) for each of the undamaged vials.

| Storage Time | Storage Temperature 20°C | 30°C |
|---|---|---|
| 3 months | 2 5 | 9 12 15 |
| 6 months | 6 6 7 7 | 16 |

We call an experiment *balanced* if all treatments have the same number of experimental units. Although this experiment was designed to be balanced with 4 experimental units per treatment, it has become *unbalanced* because the number of measured experimental units varies with treatment.

```
time=factor(rep(c(3,6),each=5))
temp=factor(rep(c(20,30,20,30),c(2,3,4,1)))
y=c(2,5,9,12,15,6,6,7,7,16)
d=data.frame(time,temp,y)
d
   time temp  y
1     3   20  2
2     3   20  5
3     3   30  9
4     3   30 12
5     3   30 15
6     6   20  6
7     6   20  6
8     6   20  7
9     6   20  7
10    6   30 16

o=lm(y~time+temp+time:temp,data=d)
```

```
model.matrix(o)
   (Intercept) time6 temp30 time6:temp30
1            1     0      0            0
2            1     0      0            0
3            1     0      1            0
4            1     0      1            0
5            1     0      1            0
6            1     1      0            0
7            1     1      0            0
8            1     1      0            0
9            1     1      0            0
10           1     1      1            1
attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$time
[1] "contr.treatment"

attr(,"contrasts")$temp
[1] "contr.treatment"
```

```
coef(o)
 (Intercept)         time6         temp30 time6:temp30
         3.5           3.0            8.5           1.0
vcov(o)
             (Intercept)     time6    temp30 time6:temp30
(Intercept)     1.958333 -1.958333 -1.958333     1.958333
time6          -1.958333  2.937500  1.958333    -2.937500
temp30         -1.958333  1.958333  3.263889    -3.263889
time6:temp30    1.958333 -2.937500 -3.263889     8.159722
```

```
#Cell means are
#
#           temp 20   temp 30
#           -----------------------
# time 3    mu        mu+temp30
# time 6    mu+time6  mu+time6+temp30+time6:temp30
#
#Time main effects?
#
#    (mu+mu+temp30)/2
#   -(mu+time6+mu+time6+temp30+time6:temp30)/2
#    ---------------------------
#   -time6-time6:temp30/2
#
#   H0:time6+time6:temp30/2=0
```

```
test=function(lmout,C,d=0){
  b=coef(lmout)
  V=vcov(lmout)
  dfn=nrow(C)
  dfd=lmout$df
  Cb.d=C%*%b-d
  Fstat=drop(t(Cb.d)%*%solve(C%*%V%*%t(C))%*%Cb.d/dfn)
  pvalue=1-pf(Fstat,dfn,dfd)
  list(Fstat=Fstat,pvalue=pvalue)
}

o$coe
 (Intercept)         time6        temp30 time6:temp30
         3.5           3.0           8.5           1.0

Ctime=matrix(c(
0,1,0,.5
),nrow=1,byrow=T)
```

```
Ctemp=matrix(c(
0,0,1,.5
),nrow=1,byrow=T)

Ctimetempint=matrix(c(
0,0,0,1
),nrow=1,byrow=T)

test(o,Ctime)
$Fstat
[1] 6.005106
$pvalue
[1] 0.04975481

test(o,Ctemp)
$Fstat
[1] 39.70723
$pvalue
[1] 0.0007447007
```

```
test(o,Ctimetempint)
$Fstat
[1] 0.1225532

$pvalue
[1] 0.7382431
```

```
#The R function anova will produce tests for
#the presence of time main effects,
#temp main effects, and time-by-temp interaction.
#However, these are "Type I Tests" that do not
#in general match the "Type III Tests" above.

anova(o)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value   Pr(>F)
time       1   0.10   0.100  0.0255 0.878292
temp       1 158.42 158.420 40.4477 0.000709 ***
time:temp  1   0.48   0.480  0.1226 0.738243
Residuals  6  23.50   3.917
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’
0.1 ‘ ’ 1
```

```
#Any difference among the four treatment means?
#
#           temp 20   temp 30
#           ------------------------
# time 3    mu        mu+temp30
# time 6    mu+time6  mu+time6+temp30+time6:temp30

Coverall=matrix(c(
0,1,0,0,
0,0,1,0,
0,0,0,1
),nrow=3,byrow=T)

test(o,Coverall)
$Fstat
[1] 13.53191

$pvalue
[1] 0.004438826
```

```
#Other choices for C can provide the same test.

Coverall=matrix(c(
0,0,1,0,
0,0,1,1,
0,1,0,0
),nrow=3,byrow=T)

test(o,Coverall)
$Fstat
[1] 13.53191

$pvalue
[1] 0.004438826
```

```
#Any difference between the 3 month, 20 degree mean
#and the average of the other three treatment means?
#
#           temp 20   temp 30
#           -------------------------
# time 3    mu        mu+temp30
# time 6    mu+time6  mu+time6+temp30+time6:temp30
#
#
# 2*time6/3+2*temp30/3+time6:temp30

C=matrix(c(
0,2/3,2/3,1/3
),nrow=1,byrow=T)

test(o,C)
$Fstat
[1] 24.17488
$pvalue
[1] 0.002665384
```

```
#The following function can be used to obtain
#confidence intervals for each element of an
#estimable C*beta.
ci=function(lmout,C,a=0.05)
{
  b=coef(lmout)
  V=vcov(lmout)
  df=lmout$df
  Cb=C%*%b
  se=sqrt(diag(C%*%V%*%t(C)))
  tval=qt(1-a/2,df)
  low=Cb-tval*se
  up=Cb+tval*se
  m=cbind(C,Cb,se,low,up)
  dimnames(m)[[2]]=c(paste("c",1:ncol(C),sep=""),
              "estimate","se",
              paste(100*(1-a),"% Conf.",sep=""),
              "limits")
  m
}
```

```
ci(o,Ctime)
     c1 c2 c3  c4 estimate       se   95% Conf.  limits
[1,] 0  1  0 0.5      3.5 1.428261 0.005170315 6.99483

ci(o,Coverall)
      c1 c2 c3 c4 estimate       se 95% Conf.    limits
[1,]  0  0  1  0      8.5 1.806624  4.079351 12.920649
[2,]  0  0  1  1      9.5 2.212653  4.085833 14.914167
[3,]  0  1  0  0      3.0 1.713914 -1.193796  7.193796
```

```
summary(o)
Call:
lm(formula = y ~ time temp time:temp, data = d)

Residuals:
       Min          1Q      Median          3Q         Max
-3.000e+00 -5.000e-01 -7.606e-17  5.000e-01  3.000e+00

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.500      1.399   2.501  0.04646 *
time6           3.000      1.714   1.750  0.13062
temp30          8.500      1.807   4.705  0.00331 **
time6:temp30    1.000      2.857   0.350  0.73824
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.979 on 6 degrees of freedom
Multiple R-squared: 0.8712, Adjusted R-squared: 0.8068
F-statistic: 13.53 on 3 and 6 DF,  p-value: 0.004439
```

```
#############################################################
#
#           temp 20   temp 30
#           ------------------------
#
# time 3    mu          mu+temp30
#
# time 6    mu+time6    mu+time6+temp30+time6:temp30
#
#############################################################
```