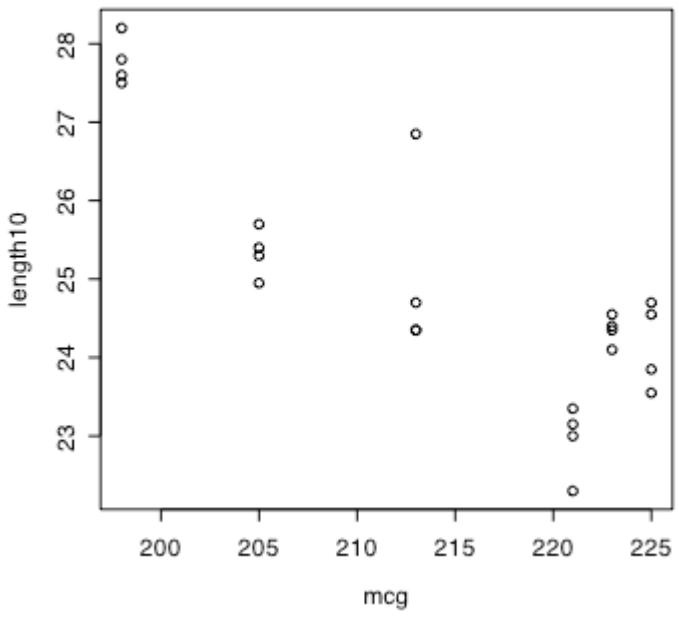
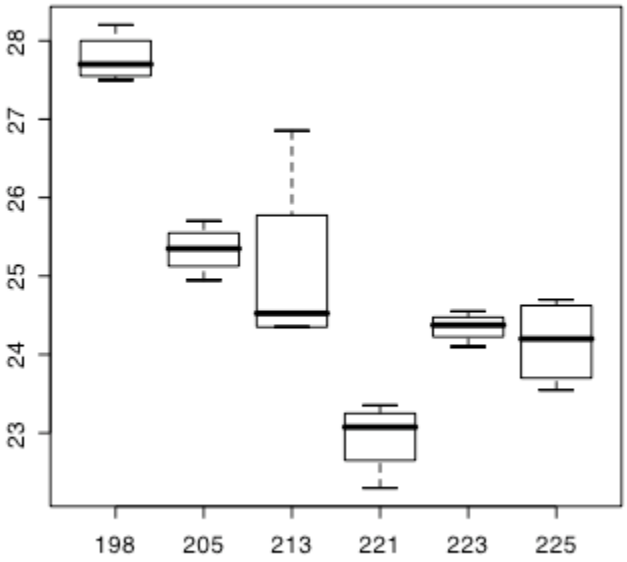
Fungus in test tubes

```
> tubes =
read.table("http://www.utstat.toronto.edu/~brunner/appliedf11/data/littletubes.data
")
> tubes
   mcg length10 weight
1  198    27.80 0.5996
2  198    28.20 0.6040
3  198    27.60 0.6172
4  198    27.50 0.6053
5  205    24.95 0.6769
6  205    25.70 0.7057
7  205    25.40 0.7271
8  205    25.30 0.6029
9  213    26.85 0.6023
10 213    24.35 0.6976
11 213    24.70 0.7154
12 213    24.35 0.6575
13 221    23.35 0.5958
14 221    23.00 0.6789
15 221    22.30 0.6965
16 221    23.15 0.6433
17 223    24.10 0.5479
18 223    24.55 0.5604
19 223    24.35 0.5446
20 223    24.40 0.5398
21 225    23.55 0.5615
22 225    24.55 0.6363
23 225    24.70 0.5753
24 225    23.85 0.6627
> attach(tubes) # mcg, length10 and weight are now available without direct
>              # reference to tubes
> # First, look at the data. Scatterplot might be better than boxplots
> # because there are so few cases in each category
> plot(mcg,length10) # Sizable differences, possible outlier for mcg213
```

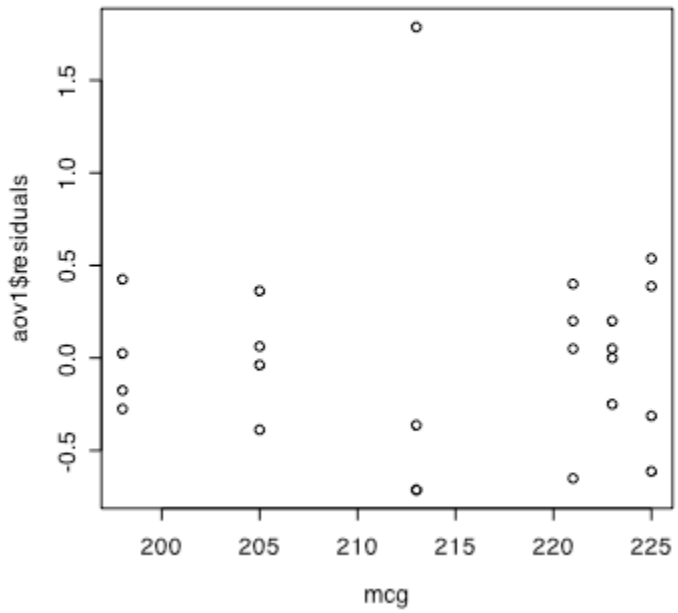


```
> fungus = factor(mcg) # Fungus is a factor (categorical variable) based on mcg
> plot(fungus,length10) # Boxplots! Again, possible outlier for mcg213
```

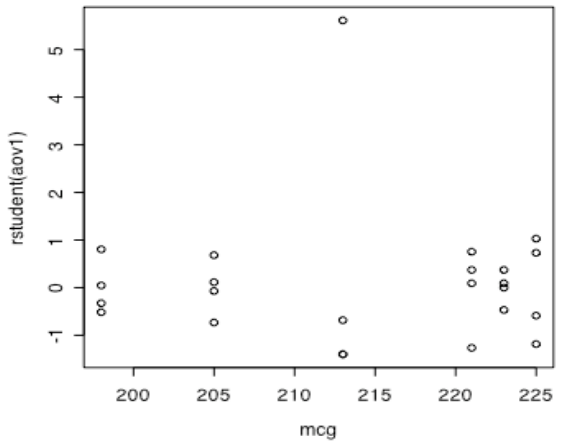



```
> # Make a table of means and SDs. Why is this so hard?!
> name = names(table(fungus)); k = length(name)
> meantable = numeric(3*k); dim(meantable) = c(k,3)
> rownames(meantable) = name; colnames(meantable) = c("Mean","St.Dev.","N")
> for(i in 1:k)
+     {
+     meantable[i,1] = mean(length10[fungus==name[i]])
+     meantable[i,2] = sqrt(var(length10[fungus==name[i]]))
+     meantable[i,3] = length(length10[fungus==name[i]])
+     }
> meantable
       Mean   St.Dev. N
198 27.7750 0.3095696 4
205 25.3375 0.3092329 4
213 25.0625 1.2030344 4
221 22.9500 0.4564355 4
223 24.3500 0.1870829 4
225 24.1625 0.5513242 4
> meantable[order(meantable[,1]),] # Rows sorted by mean
       Mean   St.Dev. N
221 22.9500 0.4564355 4
225 24.1625 0.5513242 4
223 24.3500 0.1870829 4
213 25.0625 1.2030344 4
205 25.3375 0.3092329 4
198 27.7750 0.3095696 4
> # MCG 213 has a large standard deviation
```

```
> # Fit a linear model: One-factor ANOVA
>
> aov1 = aov(length10 ~ fungus)
> summary(aov1) # Just ANOVA summary table
            Df Sum Sq Mean Sq F value    Pr(>F)
fungus       5 52.492  10.498  28.815 5.415e-08 ***
Residuals   18  6.558   0.364
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> model.tables(aov1,type = "means") # Easier table of means, but no SD or N
Tables of means
Grand mean

24.93958

 fungus
fungus
   198    205    213    221    223    225
27.775 25.338 25.063 22.950 24.350 24.163

> sort(aov1$residuals)
          10           12           15           21            5           11
-7.12500e-01 -7.12500e-01 -6.50000e-01 -6.12500e-01 -3.87500e-01 -3.62500e-01
          24            4           17            3            8           19
-3.12500e-01 -2.75000e-01 -2.50000e-01 -1.75000e-01 -3.75000e-02  8.85154e-16
           1           20           14            7           16           18
 2.50000e-02  5.00000e-02  5.00000e-02  6.25000e-02  2.00000e-01  2.00000e-01
           6           22           13            2           23            9
 3.62500e-01  3.87500e-01  4.00000e-01  4.25000e-01  5.37500e-01  1.78750e+00
> plot(mcg,aov1$residuals)
```



```
> # Deleted studentized residuals are even more revealing
> plot(mcg,rstudent(aov1))
```




```
> cbind(tubes[,1:2],rstudent(aov1))
   mcg length10 rstudent(aov1)
1  198    27.80     0.04648059
2  198    28.20     0.80503969
3  198    27.60    -0.32636103
4  198    27.50    -0.51523021
5  205    24.95    -0.73165799
6  205    25.70     0.68311253
7  205    25.40     0.11624025
8  205    25.30    -0.06972642
9  213    26.85     5.61400431
10 213    24.35    -1.39876217
11 213    24.70    -0.68311253
12 213    24.35    -1.39876217
13 221    23.35     0.75604063
14 221    23.00     0.09297890
15 221    22.30    -1.26392161
16 221    23.15     0.37334224
17 223    24.10    -0.46775769
18 223    24.55     0.37334224
19 223    24.35     0.00000000
20 223    24.40     0.09297890
21 225    23.55    -1.18478153
22 225    24.55     0.73165799
23 225    24.70     1.02997597
24 225    23.85    -0.58682517

>
> # The scientists decided that Case 9 was the result of contamination,
> # and we discarded it.
>
```

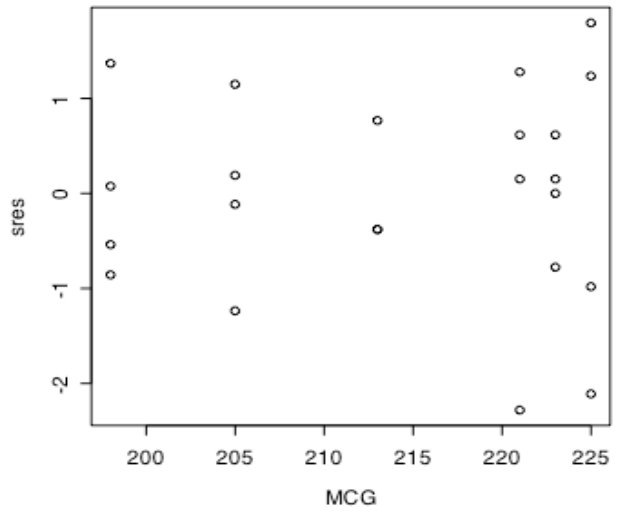
```
> MCG = mcg # Case sensitive
> MCG[9] = NA
> Fungus = factor(MCG)
> table(Fungus)
Fungus
198 205 213 221 223 225
  4   4   3   4   4   4
>
> aov2 = aov(length10 ~ Fungus); summary(aov2)
            Df Sum Sq Mean Sq F value    Pr(>F)
Fungus       5 52.944  10.589  78.335 3.953e-11 ***
Residuals   17  2.298   0.135
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
1 observation deleted due to missingness
> summary(aov1)
            Df Sum Sq Mean Sq F value    Pr(>F)
fungus       5 52.492  10.498  28.815 5.415e-08 ***
Residuals   18  6.558   0.364
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # Another table of means and SDs
> # Unfortunately, must be very careful
> mean(length10[Fungus==225])
[1] NA
> mean(length10[Fungus==225],na.rm=T)
[1] 24.1625

> name = names(table(Fungus)); k = length(name)
> meantable = numeric(3*k); dim(meantable) = c(k,3)
> rownames(meantable) = name; colnames(meantable) = c("Mean","St.Dev.","N")
> for(i in 1:k)
+     {
+     meantable[i,1] = mean(length10[Fungus==name[i]],na.rm=T)
+     meantable[i,2] = sqrt( var(length10[Fungus==name[i]],na.rm=T))
+     meantable[i,3] = length(length10[Fungus==name[i]])
+     }
> meantable
        Mean   St.Dev. N
198 27.77500 0.3095696 5
205 25.33750 0.3092329 5
213 24.46667 0.2020726 4
221 22.95000 0.4564355 5
223 24.35000 0.1870829 5
225 24.16250 0.5513242 5
> # Look at this carefully. What is wroNg?
```

```
> # R's Missing Value Problem
>
> cbind(mcg,fungus,MCG,Fungus)
      mcg fungus MCG Fungus
 [1,] 198      1 198      1
 [2,] 198      1 198      1
 [3,] 198      1 198      1
 [4,] 198      1 198      1
 [5,] 205      2 205      2
 [6,] 205      2 205      2
 [7,] 205      2 205      2
 [8,] 205      2 205      2
 [9,] 213      3  NA     NA
[10,] 213      3 213      3
[11,] 213      3 213      3
[12,] 213      3 213      3
[13,] 221      4 221      4
[14,] 221      4 221      4
[15,] 221      4 221      4
[16,] 221      4 221      4
[17,] 223      5 223      5
[18,] 223      5 223      5
[19,] 223      5 223      5
[20,] 223      5 223      5
[21,] 225      6 225      6
[22,] 225      6 225      6
[23,] 225      6 225      6
[24,] 225      6 225      6
> mcg[fungus==225]
[1] 225 225 225 225
> MCG[Fungus==225]
[1]  NA 225 225 225 225
> # NA matches EVERYTHING.
> length(mcg[mcg==225]); length(MCG[MCG==225])
[1] 4
[1] 5

>
> # Work-around is ugly
> cbind(Fungus==225,!is.na(Fungus))
      [,1]  [,2]
 [1,] FALSE  TRUE
 [2,] FALSE  TRUE
 [3,] FALSE  TRUE
 [4,] FALSE  TRUE
 [5,] FALSE  TRUE
 [6,] FALSE  TRUE
 [7,] FALSE  TRUE
 [8,] FALSE  TRUE
 [9,]    NA FALSE
[10,] FALSE  TRUE
[11,] FALSE  TRUE
[12,] FALSE  TRUE
[13,] FALSE  TRUE
[14,] FALSE  TRUE
[15,] FALSE  TRUE
[16,] FALSE  TRUE
[17,] FALSE  TRUE
[18,] FALSE  TRUE
[19,] FALSE  TRUE
[20,] FALSE  TRUE
[21,]  TRUE  TRUE
[22,]  TRUE  TRUE
[23,]  TRUE  TRUE
[24,]  TRUE  TRUE
```

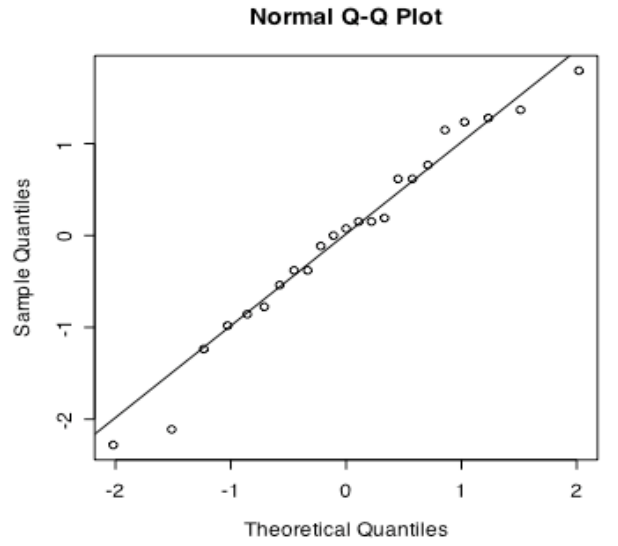
```
> cbind(Fungus==225,!is.na(Fungus),(Fungus==225)&&(!is.na(Fungus))) # ??!
       [,1]  [,2]  [,3]
 [1,] FALSE  TRUE FALSE
 [2,] FALSE  TRUE FALSE
 [3,] FALSE  TRUE FALSE
 [4,] FALSE  TRUE FALSE
 [5,] FALSE  TRUE FALSE
 [6,] FALSE  TRUE FALSE
 [7,] FALSE  TRUE FALSE
 [8,] FALSE  TRUE FALSE
 [9,]    NA FALSE FALSE
[10,] FALSE  TRUE FALSE
[11,] FALSE  TRUE FALSE
[12,] FALSE  TRUE FALSE
[13,] FALSE  TRUE FALSE
[14,] FALSE  TRUE FALSE
[15,] FALSE  TRUE FALSE
[16,] FALSE  TRUE FALSE
[17,] FALSE  TRUE FALSE
[18,] FALSE  TRUE FALSE
[19,] FALSE  TRUE FALSE
[20,] FALSE  TRUE FALSE
[21,]  TRUE  TRUE FALSE
[22,]  TRUE  TRUE FALSE
[23,]  TRUE  TRUE FALSE
[24,]  TRUE  TRUE FALSE
>
> both = NULL
> for(i in 1:length(Fungus))
+     {both = c(both,(Fungus==225)[i] && (!is.na(Fungus))[i])}
> cbind(Fungus==225,!is.na(Fungus),both)
                   both
 [1,] FALSE  TRUE FALSE
 [2,] FALSE  TRUE FALSE
 [3,] FALSE  TRUE FALSE
 [4,] FALSE  TRUE FALSE
 [5,] FALSE  TRUE FALSE
 [6,] FALSE  TRUE FALSE
 [7,] FALSE  TRUE FALSE
 [8,] FALSE  TRUE FALSE
 [9,]    NA FALSE FALSE
[10,] FALSE  TRUE FALSE
[11,] FALSE  TRUE FALSE
[12,] FALSE  TRUE FALSE
[13,] FALSE  TRUE FALSE
[14,] FALSE  TRUE FALSE
[15,] FALSE  TRUE FALSE
[16,] FALSE  TRUE FALSE
[17,] FALSE  TRUE FALSE
[18,] FALSE  TRUE FALSE
[19,] FALSE  TRUE FALSE
[20,] FALSE  TRUE FALSE
[21,]  TRUE  TRUE  TRUE
[22,]  TRUE  TRUE  TRUE
[23,]  TRUE  TRUE  TRUE
[24,]  TRUE  TRUE  TRUE
> sum(both)
[1] 4
> sum(Fungus==225*!is.na(Fungus),na.rm=T)
[1] 4
> # It's a nightmare
>
> # Here we will proceed by removing the entire case instead of using NA.
> # This is NOT recommended for data sets with lots of variables
> # and lots of missing values.
```

```
> MCG = c(mcg[1:8],mcg[10:24]); Fungus=factor(MCG);
> Growth = c(length10[1:8],length10[10:24])
>
> msd <- function(fac,variab,sorted=F)
+     { # Table of means, Ns and standard deviations
+     if(prod(!is.na(fac))* prod(!is.na(variab))==0)
+         {stop("Function msd does not work when there are NAs.")}
+     if(!is.factor(fac)) Stop("First argument must be a factor.")
+     name = names(table(fac)); k = length(name)
+     msd = numeric(3*k); dim(msd) = c(k,3)
+     rownames(msd) = name
+     colnames(msd) = c("Mean","St.Dev.","N")
+     for(i in 1:k)
+         {
+         msd[i,1] = mean(variab[fac==name[i]],na.rm=T)
+         msd[i,2] = sqrt( var(variab[fac==name[i]],na.rm=T))
+         msd[i,3] = length(variab[fac==name[i]])
+         }
+     if(sorted) msd = msd[order(msd[,1]),] # Rows sorted by mean
+     msd # Return the value
+     } # End of function msd
> msd(Fungus,Growth)
        Mean   St.Dev. N
198 27.77500 0.3095696 4
205 25.33750 0.3092329 4
213 24.46667 0.2020726 3
221 22.95000 0.4564355 4
223 24.35000 0.1870829 4
225 24.16250 0.5513242 4
>
> # Re-do Model 2
> aov2 = aov(Growth ~ Fungus); model.tables(aov2,type = "means")
Tables of means
Grand mean

24.85652

 Fungus
      198   205   213   221   223   225
    27.78 25.34 24.47 22.95 24.35 24.16
rep  4.00  4.00  3.00  4.00  4.00  4.00
> summary(aov2)
            Df Sum Sq Mean Sq F value    Pr(>F)
Fungus       5 52.944  10.589  78.335 3.953e-11 ***
Residuals   17  2.298   0.135
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> sres = rstudent(aov2); sort(sres)
         14          20           5          23           4          16
-2.27952905 -2.10996784 -1.23574401 -0.98034504 -0.85691706 -0.77593145
          3           9          11           8          18           1
-0.53801404 -0.37872434 -0.37872434 -0.11430645  0.00000000  0.07618702
         19          13           7          15          17          10
 0.15245703  0.15245703  0.19064919  0.61658292  0.61658292  0.76784407
          6          21          12           2          22
 1.14919098  1.23574401  1.27961617  1.36864983  1.79507853
```

```
> plot(MCG,sres)
```




```
> # What is prob of getting 2 or more greater than 2 in abs value? t, 20 df
> qt(0.975,20)
[1] 2.085963
> 1 - (.95^23 + 23*.05*.95^22)
[1] 0.3205796

> qqnorm(sres)
> qqline(sres)
```




```
> # Jim and Linda would have blamed their grad students
> # What should be done about possible non-normality? Permutation tests?
```

```
> # Mean growth on day 10 depends on fungus type
> summary(aov2)
            Df Sum Sq Mean Sq F value    Pr(>F)
Fungus       5 52.944  10.589  78.335 3.953e-11 ***
Residuals   17  2.298   0.135
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> # Which means are different?
> TukeyHSD(aov2,ordered=T)
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = Growth ~ Fungus)

$Fungus
             diff         lwr      upr     p adj
225-221 1.2125000  0.38092404 2.044076 0.0025656
223-221 1.4000000  0.56842404 2.231576 0.0005949
213-221 1.5166667  0.61846197 2.414871 0.0005762
205-221 2.3875000  1.55592404 3.219076 0.0000007
198-221 4.8250000  3.99342404 5.656576 0.0000000
223-225 0.1875000 -0.64407596 1.019076 0.9765713
213-225 0.3041667 -0.59403803 1.202371 0.8814129
205-225 1.1750000  0.34342404 2.006576 0.0034487
198-225 3.6125000  2.78092404 4.444076 0.0000000
213-223 0.1166667 -0.78153803 1.014871 0.9981172
205-223 0.9875000  0.15592404 1.819076 0.0151272
198-223 3.4250000  2.59342404 4.256576 0.0000000
205-213 0.8708333 -0.02737136 1.769038 0.0603187
198-213 3.3083333  2.41012864 4.206538 0.0000000
198-205 2.4375000  1.60592404 3.269076 0.0000005

> msd(Fungus,Growth,sorted=T)
        Mean   St.Dev. N
221 22.95000 0.4564355 4
225 24.16250 0.5513242 4
223 24.35000 0.1870829 4
213 24.46667 0.2020726 3
205 25.33750 0.3092329 4
198 27.77500 0.3095696 4
> lm2 = lm(Growth ~ Fungus); summary(lm2)

Call:
lm(formula = Growth ~ Fungus)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6500 -0.2125  0.0250  0.2167  0.5375

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.7750     0.1838 151.092  < 2e-16 ***
Fungus205    -2.4375     0.2600  -9.376 3.95e-08 ***
Fungus213    -3.3083     0.2808 -11.782 1.33e-09 ***
Fungus221    -4.8250     0.2600 -18.560 1.01e-12 ***
Fungus223    -3.4250     0.2600 -13.174 2.38e-10 ***
Fungus225    -3.6125     0.2600 -13.896 1.04e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3677 on 17 degrees of freedom
Multiple R-Squared: 0.9584,   Adjusted R-squared: 0.9462
F-statistic: 78.34 on 5 and 17 DF,  p-value: 3.953e-11
```

```
> # Note aov is a wrapper for lm
> contrasts(Fungus)
    205 213 221 223 225
198   0   0   0   0   0
205   1   0   0   0   0
213   0   1   0   0   0
221   0   0   1   0   0
223   0   0   0   1   0
225   0   0   0   0   1

> # Now experiment with contrasts (Dummy variable coding)
> fun <- Fungus
> contrasts(fun) <- contr.treatment(n=6,base=4)
> # n is the number of categories, base is the reference category
> contrasts(fun)
    1 2 3 5 6
198 1 0 0 0 0
205 0 1 0 0 0
213 0 0 1 0 0
221 0 0 0 0 0
223 0 0 0 1 0
225 0 0 0 0 1
> contrasts(fun) <- contr.sum;  contrasts(fun) # Effect coding
    [,1] [,2] [,3] [,4] [,5]
198    1    0    0    0    0
205    0    1    0    0    0
213    0    0    1    0    0
221    0    0    0    1    0
223    0    0    0    0    1
225   -1   -1   -1   -1   -1
> contrasts(fun) <- contr.poly;  contrasts(fun)
            .L         .Q         .C         ^4          ^5
198 -0.5976143  0.5455447 -0.3726780  0.1889822 -0.06299408
205 -0.3585686 -0.1091089  0.5217492 -0.5669467  0.31497039
213 -0.1195229 -0.4364358  0.2981424  0.3779645 -0.62994079
221  0.1195229 -0.4364358 -0.2981424  0.3779645  0.62994079
223  0.3585686 -0.1091089 -0.5217492 -0.5669467 -0.31497039
225  0.5976143  0.5455447  0.3726780  0.1889822  0.06299408
> sum(contrasts(fun)[,1]*contrasts(fun)[,2])
[1] 9.020562e-17
> # ORTHOGANAL polynomials
```

For Scheffé and Bonferroni multiple comparisons (and for other purposes), we would like to be able to test general linear hypotheses of the form **Lβ**=**h**. The R package `gmodels` has a function `glh.test` which does this.

From the Packages and Data menu, select Package Installer
from CRAN (binaries), Get List
select gmodels
Install Selected.

This worked on a Windows machine. For the Mac, needed to also instal the `gtools` and `gdata` packages from the Packages and Data menu.

```
> summary(lm2) # Again

Call:
lm(formula = Growth ~ Fungus)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6500 -0.2125  0.0250  0.2167  0.5375

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.7750     0.1838 151.092  < 2e-16 ***
Fungus205    -2.4375     0.2600  -9.376 3.95e-08 ***
Fungus213    -3.3083     0.2808 -11.782 1.33e-09 ***
Fungus221    -4.8250     0.2600 -18.560 1.01e-12 ***
Fungus223    -3.4250     0.2600 -13.174 2.38e-10 ***
Fungus225    -3.6125     0.2600 -13.896 1.04e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3677 on 17 degrees of freedom
Multiple R-Squared: 0.9584,   Adjusted R-squared: 0.9462
F-statistic: 78.34 on 5 and 17 DF,  p-value: 3.953e-11

> # Use glh.test to reproduce test for Fungus213 vs 198
> # Should get F = t^2 = (-11.782)^2 = 138.81
> # H0: beta2 = 0
> c1 = t(as.matrix(c(0,0,1,0,0,0))); c1
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    1    0    0    0
> glh.test(lm2,c1)

       Test of General Linear Hypothesis
Call:
glh.test(reg = lm2, cm = c1)
F = 138.8086, df1 =  1, df2 = 17, p-value = 1.332e-09

> glh.test(lm2,c(0,0,1,0,0,0)) # Works!

       Test of General Linear Hypothesis
Call:
glh.test(reg = lm2, cm = c(0, 0, 1, 0, 0, 0))
F = 138.8086, df1 =  1, df2 = 17, p-value = 1.332e-09

>
> # Now test mcg205 vs mcg213 with a Bonferroni correction.
> # Expect p > 0.06, because Tukey should be more powerful.
> # H0: beta1=beta2
> glh.test(lm2,c(0,1,-1,0,0,0))

       Test of General Linear Hypothesis
Call:
glh.test(reg = lm2, cm = c(0, 1, -1, 0, 0, 0))
F = 9.6176, df1 =  1, df2 = 17, p-value = 0.006487

> 0.006487*15 # Why 15?
[1] 0.097305
>
> # There is just one critical value for the Scheffe tests
> scrit = 5*qf(0.95,5,17); scrit
[1] 14.04998
> # Note 9.6176 < 14.04998
>
```

```
> # 205 vs. average of 225,223,213 -- As a Scheffe test
> mcg205VSaver = glh.test(lm2,c(0,3,-1,0,-1,-1)); mcg205VSaver

       Test of General Linear Hypothesis
Call:
glh.test(reg = lm2, cm = c(0, 3, -1, 0, -1, -1))
F = 22.0767, df1 =  1, df2 = 17, p-value = 0.0002068

> mcg205VSaver$statistic > scrit
   F
TRUE
>
> # Try reproducing overall F = 78.34
> # H0: beta1 = ... = beta5 = 0
> c2 = rbind(c(0,1,0,0,0,0),
+            c(0,0,1,0,0,0),
+            c(0,0,0,1,0,0),
+            c(0,0,0,0,1,0),
+            c(0,0,0,0,0,1)
+           )
> glh.test(lm2,c2)

       Test of General Linear Hypothesis
Call:
glh.test(reg = lm2, cm = c2)
F = 78.3354, df1 =  5, df2 = 17, p-value = 3.953e-11
```