R as a Statistics Package

```
> # Potato data with R
> # Change Working Directory in the Misc menu: Navigate to where the data file is.
> getwd()
[1] "/Users/brunner/Documents/Current_Work/Class/429f07/R_Work"

> spud <- read.table("potato2.data")
> spud
   Bact Temp Rot
1     1    1   7
2     1    1   7
3     1    1   9
```

skipping ...

```
52    3    2  20
53    3    2  24
54    3    2   8
> spud[1:10,]
   Bact Temp Rot
1     1    1   7
2     1    1   7
3     1    1   9
4     1    1   0
5     1    1   0
6     1    1   0
7     1    1   9
8     1    1   0
9     1    1   0
10    1    2  10
> mean(spud$Rot)
[1] 9.407407
> length(spud$Rot)
[1] 54
> # One approach is to extract the variables you want to work with.
> rot <- spud$Rot
> bact <- factor(spud$Bact) # Sets up dummy variables
> contrasts(bact)
  2 3
1 0 0
2 1 0
3 0 1
> # Make 3 the reference category
> contrasts(bact) <- contr.treatment(3, base = 3) ; contrasts(bact)
  1 2
1 1 0
2 0 1
3 0 0
> temp <- factor(spud$Temp)



> # the lm (linear model) function is very powerful. See help(lm)
> lm(rot~bact+temp+bact:temp)
```

```
Call:
lm(formula = rot ~ bact + temp + bact:temp)

Coefficients:
(Intercept)        bact1        bact2        temp2  bact1:temp2  bact2:temp2
      8.000       -4.444       -3.222       11.556       -8.111       -2.778

> # That does not show you much -- just the surface. The lm function creates
> # an lm OBJECT, an elaborate list of matrices containing numbers. Usually
> # you save the object and extract what you want from it.
> taters1 <- lm(rot~bact+temp+bact:temp)
> summary(taters1) # By default it's a regression model

Call:
lm(formula = rot ~ bact + temp + bact:temp)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5556  -3.5556   0.2222   3.4444   9.4444

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.000      1.562   5.121 5.33e-06 ***
bact1         -4.444      2.209  -2.012   0.0499 *
bact2         -3.222      2.209  -1.459   0.1512
temp2         11.556      2.209   5.231 3.66e-06 ***
bact1:temp2   -8.111      3.124  -2.596   0.0125 *
bact2:temp2   -2.778      3.124  -0.889   0.3784
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.686 on 48 degrees of freedom
Multiple R-Squared: 0.6106,   Adjusted R-squared:  0.57
F-statistic: 15.05 on 5 and 48 DF,  p-value: 7.003e-09

> anova(taters1) # R is smart enough to allow for the method of dummy variable coding

Analysis of Variance Table

Response: rot
          Df  Sum Sq Mean Sq F value    Pr(>F)
bact       2  651.81  325.91 14.8390 9.608e-06 ***
temp       1  848.07  848.07 38.6138 1.180e-07 ***
bact:temp  2  152.93   76.46  3.4815   0.03874 *
Residuals 48 1054.22   21.96
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> # That ANOVA summary table is a matrix, and you can get into it
> anova(taters1)[1,4] # F statistic for Bacteria type
[1] 14.83895
> # Calculate the proportion of remaining variation explained by Bacteria
> F <- anova(taters1)[1,4] ; s <- anova(taters1)[1,1]
> nminusp <- anova(taters1)[4,1]
> a <- s*F / (s*F + nminusp) ; print(a)
[1] 0.3820637
```

```
> residuals(taters1) # There's a lot more in there.
          1           2           3           4           5           6
  3.4444444   3.4444444   5.4444444  -3.5555556  -3.5555556  -3.5555556
          7           8           9          10          11          12
  5.4444444  -3.5555556  -3.5555556   3.0000000  -1.0000000   3.0000000
         13          14          15          16          17          18
 -3.0000000   3.0000000  -2.0000000   1.0000000  -7.0000000   3.0000000
         19          20          21          22          23          24
 -2.7777778  -0.7777778   4.2222222  -0.7777778   0.2222222   5.2222222
         25          26          27          28          29          30
 -0.7777778   0.2222222  -4.7777778   3.4444444   4.4444444  -5.5555556
         31          32          33          34          35          36
-10.5555556   9.4444444  -6.5555556   1.4444444   0.4444444   3.4444444
         37          38          39          40          41          42
  5.0000000   3.0000000  -5.0000000   2.0000000  -4.0000000  -1.0000000
         43          44          45          46          47          48
  7.0000000  -6.0000000  -1.0000000   6.4444444  -0.5555556   4.4444444
         49          50          51          52          53          54
 -4.5555556   2.4444444  -1.5555556   0.4444444   4.4444444 -11.5555556

> # More sophisticated ...
> sumtable <- anova(lm(Rot~factor(Bact)*factor(Temp),data=spud))
> sumtable
Analysis of Variance Table

Response: Rot
                          Df  Sum Sq Mean Sq F value    Pr(>F)
factor(Bact)               2  651.81  325.91 14.8390 9.608e-06 ***
factor(Temp)               1  848.07  848.07 38.6138 1.180e-07 ***
factor(Bact):factor(Temp)  2  152.93   76.46  3.4815   0.03874 *
Residuals                 48 1054.22   21.96
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # Simple descriptive statistics are available, but it's clumsy
> aggregate(rot,by=list(temp,bact),mean)
  Group.1 Group.2         x
1       1       1  3.555556
2       2       1  7.000000
3       1       2  4.777778
4       2       2 13.555556
5       1       3  8.000000
6       2       3 19.555556
> aggregate(rot,by=list(Temp=temp,Bact=bact),mean) # Better Labels
  Temp Bact         x
1    1    1  3.555556
2    2    1  7.000000
3    1    2  4.777778
4    2    2 13.555556
5    1    3  8.000000
6    2    3 19.555556
>
```

```
> Meanz <- aggregate(rot,by=list(Temp=temp,Bact=bact),mean)
> Meanz
  Temp Bact         x
1    1    1  3.555556
2    2    1  7.000000
3    1    2  4.777778
4    2    2 13.555556
5    1    3  8.000000
6    2    3 19.555556
> dimnames(Meanz)
[[1]]
[1] "1" "2" "3" "4" "5" "6"

[[2]]
[1] "Temp" "Bact" "x"

> dimnames(Meanz)[[2]][3] <- "Mean" ; Meanz
  Temp Bact      Mean
1    1    1  3.555556
2    2    1  7.000000
3    1    2  4.777778
4    2    2 13.555556
5    1    3  8.000000
6    2    3 19.555556
>
> Varz <- aggregate(rot,by=list(temp,bact),var)
> SummaryStats <- cbind(Meanz,sqrt(Varz[,3])) ; SummaryStats
  Temp Bact      Mean sqrt(Varz[, 3])
1    1    1  3.555556        4.275252
2    2    1  7.000000        3.535534
3    1    2  4.777778        3.113590
4    2    2 13.555556        6.326751
5    1    3  8.000000        4.555217
6    2    3 19.555556        5.525195
> dimnames(SummaryStats)[[2]][4] <- "St Dev"  ; SummaryStats
  Temp Bact      Mean   St Dev
1    1    1  3.555556 4.275252
2    2    1  7.000000 3.535534
3    1    2  4.777778 3.113590
4    2    2 13.555556 6.326751
5    1    3  8.000000 4.555217
6    2    3 19.555556 5.525195
>
> # That shows the process. It could be done more directly. First remove what
> # we've done.
> rm(Meanz,Varz,SummaryStats)
```

```
> # Here's how to get a table of sample sizes, means and standard deviations.
> Nz <- aggregate(rot,by=list(Temp=temp,Bact=bact),length)
> Meanz <- aggregate(rot,by=list(temp,bact),mean)
> Varz <- aggregate(rot,by=list(temp,bact),var)
> SummaryStats <- cbind(Nz,Meanz[,3],sqrt(Varz[,3]))
> dimnames(SummaryStats)[[2]][3] <- "N"
> dimnames(SummaryStats)[[2]][4] <- "Mean"
> dimnames(SummaryStats)[[2]][5] <- "St Dev"  ; SummaryStats
  Temp Bact N      Mean   St Dev
1    1    1 9  3.555556 4.275252
2    2    1 9  7.000000 3.535534
3    1    2 9  4.777778 3.113590
4    2    2 9 13.555556 6.326751
5    1    3 9  8.000000 4.555217
6    2    3 9 19.555556 5.525195


> # Finally, it's important to see how R handles unequal sample sizes.
>
> length(rot)
[1] 54
>  spud2 <- spud[1:50,] # spud2 gets rows 1:50 of spud, and all the columns
> anova(lm(Rot~factor(Bact)*factor(Temp),data=spud2))
Analysis of Variance Table

Response: Rot
                          Df Sum Sq Mean Sq F value    Pr(>F)
factor(Bact)               2 440.15  220.08 10.9441 0.0001387 ***
factor(Temp)               1 777.15  777.15 38.6467 1.620e-07 ***
factor(Bact):factor(Temp)  2 183.02   91.51  4.5506 0.0159826 *
Residuals                 44 884.80   20.11
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Compare output from SAS proc glm

Source	DF	Type I SS	Mean Square	F Value	Pr > F
bact	2	440.1517460	220.0758730	10.94	0.0001
temp	1	777.1495405	777.1495405	38.65	<.0001
bact*temp	2	183.0187135	91.5093567	4.55	0.0160

Source	DF	Type III SS	Mean Square	F Value	Pr > F
bact	2	651.7789474	325.8894737	16.21	<.0001
temp	1	855.3830065	855.3830065	42.54	<.0001
bact*temp	2	183.0187135	91.5093567	4.55	0.0160

Oops! It's Type I: For Type III, need to do it the hard way with regression. Or maybe there's an add-on package I don't know about.

Often, you want work from a program, though you develop it interactively. Here is the plain text file potato.R.txt

```
# potato.R.txt
spud <- read.table("potato2.data")
rot <- spud$Rot ; bact <- spud$Bact ; temp <- spud$Temp
# Means and standard deviations
Nz <- aggregate(rot,by=list(Temp=temp,Bact=bact),length)
Meanz <- aggregate(rot,by=list(temp,bact),mean)
Varz <- aggregate(rot,by=list(temp,bact),var)
SummaryStats <- cbind(Nz,Meanz[,3],sqrt(Varz[,3]))
dimnames(SummaryStats)[[2]][3] <- "N"
dimnames(SummaryStats)[[2]][4] <- "Mean"
dimnames(SummaryStats)[[2]][5] <- "St Dev"
print(SummaryStats)
print(anova(lm(Rot~factor(Bact)*factor(Temp),data=spud)))
```

Here is how it is used:

```
>
> source("potato.R.txt")
  Temp Bact N      Mean   St Dev
1    1    1 9  3.555556 4.275252
2    2    1 9  7.000000 3.535534
3    1    2 9  4.777778 3.113590
4    2    2 9 13.555556 6.326751
5    1    3 9  8.000000 4.555217
6    2    3 9 19.555556 5.525195
Analysis of Variance Table

Response: Rot
                          Df  Sum Sq Mean Sq F value    Pr(>F)
factor(Bact)               2  651.81  325.91 14.8390 9.608e-06 ***
factor(Temp)               1  848.07  848.07 38.6138 1.180e-07 ***
factor(Bact):factor(Temp)  2  152.93   76.46  3.4815   0.03874 *
Residuals                 48 1054.22   21.96
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```