

AMS 162: Lab 3

Analyzing Factorial Experiments

Objectives:

1. To fit effects of full factorial designs
2. To evaluate significance of fitted effects

Note that there is a text file of all the R commands on the class web page, so that if you wish, you can cut and paste those in instead of retyping them.

Problem 8.6

This lab will work through problem 8.6 from the text, to show the relevant commands in R. The data for this problem are available on the class web site in the file `chromatograph.txt`.

A 2^3 experiment was performed to determine the effects of the CO concentration (0.1 vs 0.2), H_2 concentration (0.2 vs. 0.4) and volumetric flowrate (100 cc/min vs 200 cc/min) upon the output of a gas chromatograph used to measure the concentration of H_2 .

(a) Means and Standard Deviations

First we need to read the file in. Open up the file in a text editor and take a look at the format. The first three columns are the factors, coded as “-1” for the lower level and “1” for the higher level. The last three columns are measurements from three different runs at the same factor levels. Since all of the columns are labeled, use `read.table` with `header=T`:

```
> chroma = read.table("chromatograph.txt",header=T)
```

We are now asked to compute the mean and variance or standard deviation for each of the groups of runs, as well as the pooled standard deviation. We can use the `apply` command to do all of the means (or standard deviations or variances) at once. `apply` has three arguments. The first is the matrix (or data.frame) with the relevant data. Here we only want to compute the means in the last three columns, so we'll only feed `apply` those three columns. The second argument is whether we want to operate on rows (1) or columns (2). Here we want to take the averages of rows, so we will enter 1. The third argument is the function to apply to each row (or column). First we will use `mean`, then we will use `sd`:

```
> chroma$means=apply(chroma[,4:6],1,mean)
> chroma$vari=apply(chroma[,4:6],1,var)
> chroma$stdv=apply(chroma[,4:6],1,sd)
```

Two notes. First, by storing the means in `chroma$means` I have created a new column in the `chroma` data.frame that now contains the mean values. Second note is that I picked names for these new columns that are different from the names of existing R functions (so I didn't call them `mean`, `var`, and `sd`). By using `attach` after creating these new columns, we can access them by their column names as well. If we just print these out, we get lots of extra decimals. To display them better, we only need to look at two or so decimal places (but we keep all the precision in the object), so we will use the `round` command. Since we have the same number of observations at each set of factors, we compute the pooled standard deviation by taking the square root of the mean of the variances.

```
> attach(chroma)
> round(means,2)
> round(vari,2)
> round(stdv,2)
> chroma.sp=sqrt(mean(vari))
```

(b) Check for equal variances

We should check the assumption that the variances are equal.

```
> plot(means,stdv)
```

Well, they're not equal, but there isn't any particular pattern to them either, so we'll just go ahead and pretend they're close enough to equal, since we haven't learned how to deal with different variances yet.

(c) Calculate main effects and interaction effects

To compute the main effect for CO level, we could do it by hand with

```
> mean(means [CO==1]) - mean(means [CO== -1])
```

But we also need to specify the levels for the interactions to compute those:

```
> chroma$CO.H=CO*H
> chroma$CO.F1=CO*F1
> chroma$H.F1=H*F1
> chroma$CO.H.F1=CO*H*F1
```

This could be a lot of work to do by hand, especially if it were larger than a 2^3 experiment. Fortunately, we don't have to do all this by hand. We can take advantage of the `lm` command, which fits a linear model. We will learn more about this command when we get to the regression section. For now, we'll just need the fitted coefficients, which are obtained from the `coef` function applied to `lm`. Inside the `lm` command, we put the response variable on the left of a tilde (`~`) and the factors on the right connected by the multiplication sign. As we'll see in Chapter nine, we need to multiply these coefficients by 2 to get the effects:

```

> 2*coef(lm(means~C0*H*F1))
(Intercept)          C0              H              F1          C0:H          C0:F1
8.663000000 -0.246166667  2.133000000 -0.158666667 -0.005166667  0.156833333
          H:F1          C0:H:F1
0.237333333  0.053166667

```

This gives us the fitted main effects and all fitted interactions. Note that this should match what we got by hand above. We ignore the (Intercept) column for now.

(d) Boxplots of main effects

We are asked to make boxplots of the main effects similar to Figure 8.17 and comment on them. We can do this with the `boxplot` command by specifying that we are plotting the means with respect to factors by using a tilde. I'll also set this up to put all three plots on one page.

```

> par(mfrow=c(2,2))
> boxplot(means~C0,ylab="Finish",xlab="C0")
> boxplot(means~H,ylab="Finish",xlab="H")
> boxplot(means~F1,ylab="Finish",xlab="F1")
> par(mfrow=c(1,1))

```

Note that hydrogen level obviously has a large effect. The other factors don't seem to have much effect.

(e) Compute significance

The standard error of an effect is given by $s_E = \frac{2s_p}{\sqrt{n}}$. This has the same number of degrees of freedom as s_p , which is the number of total experimental runs (here $n = 24$) minus the number of means estimated (8). An effect is significant if it is larger in absolute value than a standard t at the 95% significance level times the standard error of an effect.

```

> chroma.se=2*chroma.sp/sqrt(24)
> chroma.se*qt(.975,16)
[1] 0.4862631

```

Thus we see that H is the only significant effect (again, we ignore the intercept).