Linear Regression Models

STATS 762 – Lecture Slides 6

March 25, 2019

Interactions

Interactions may be useful to model in situations where the impact of one factor may depend on the level of another factor.

- Interactions can involve just factors, just numeric regressors or both factors and numeric regressors.
- Useful when the impact on the response of one regressor depends on the level of another regressor.

ANOVA

Classically, ANOVA was used to analyse data from a designed experiment where:

- All of the explanatory variables were factors,
- The set of covariate patterns consisted of all possible combinations of levels for these factors (this was called a factorial design),
- The same number of observations were taken for each covariate pattern.

This structure simplified both the mathematical calculations and the interpretation of results. It also allowed interactions between the factors to be estimated.

Example: Weight Gain of Rats

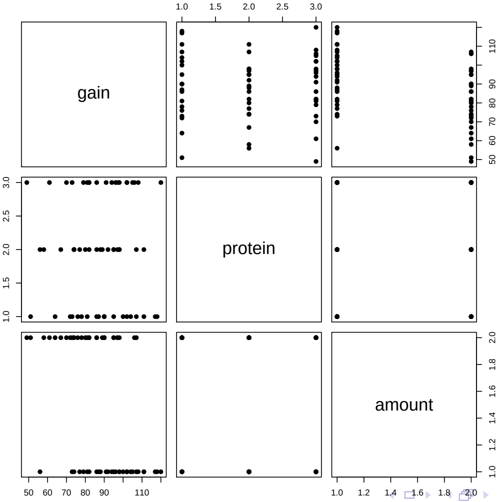
An experiment investigated the effect of diet on the weight gained by rats.

- Two treatment factors:
  1. source of protein: beef, pork, or cereal
  2. level: high or low
- Ten rats were assigned to each combination of “source” and “level”
  - this is called a balanced design – each combination of factors was used for the same number of observations

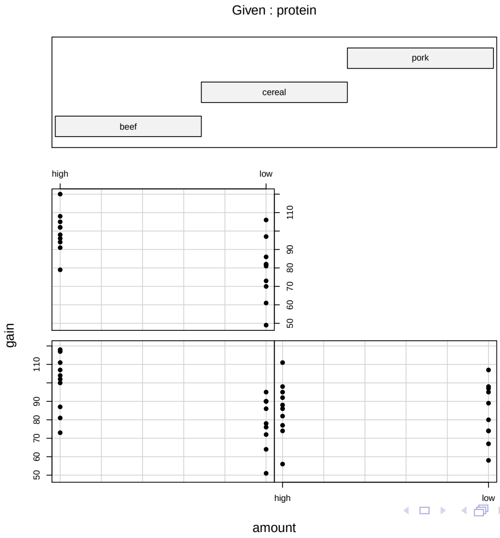
The Data Frame

```
> rats.df
   gain protein amount
1    73    Beef   High
2    98  Cereal   High
3    94    Pork   High
4    90    Beef    Low
5   107  Cereal    Low
6    49    Pork    Low
=======================
56   92  Cereal   High
57  105    Pork   High
58   78    Beef    Low
59   58  Cereal    Low
60   82    Pork    Low
```

Pairs Plot


gain
protein
amount
1.0
1.5
2.0
2.5
3.0
110
90
80
70
60
50
2.0
1.8
1.6
1.4
1.2
1.0

Conditional Plot



Two Factors

Both of the regressors (`protein` and `amount`) are factors. In *R* the `factor` function is used to designate factors:

```
> attach(rats.df)
> proteinA<-factor(protein)
> attributes(proteinA)
$levels
[1] "beef"   "cereal" "pork"
$class
[1] "factor"

> amountA<-factor(amount)
> attributes(amountA)
$levels
[1] "high" "low"
$class
[1] "factor"
```

Indicator Variables

A set of indicator variables is needed for each factor:

- The indicator variables allow the mean response to be adjusted for each different category of the factor individually.
- To accomplish this the number of indicator variables needs to be one fewer than the number of levels for the factor.
- By default *R* creates indicator variables for a "baseline" model – each factor has a baseline level and all of the indicator variables represent comparisons of the other levels to the baseline level.

Indicator Variables (cont.)

In *R* we can view the indicator variables that have been set up for a factor using the `contrasts` command:

```
> contrasts(proteinA)
       cereal pork
beef        0    0
cereal      1    0
pork        0    1

> contrasts(amountA)
     low
high   0
low    1
```

- Each column defines an indicator variable (a contrast).

The Model Defined by these Indicator Variables

Our model matrix will have four columns consisting of the intercept the two indicator variables for `proteinA` and the indicator for `amountA`. Let the coefficients for these be β_0, β_C, β_P and β_L respectively. Then the model defines the means for the six different combinations of protein source and amount as:

	Protein Source		
amount	beef	cereal	pork
high	β_0	$\beta_0 + \beta_C$	$\beta_0 + \beta_P$
low	$\beta_0 + \beta_L$	$\beta_0 + \beta_C + \beta_L$	$\beta_0 + \beta_P + \beta_L$

Output for Fitted Model

```
> ratsA.lm=lm(gain~proteinA+amountA)
> summary(ratsA.lm)
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      96.867      3.898  24.850  < 2e-16 ***
proteinAcereal   -4.700      4.774  -0.984 0.329126
proteinApork     -0.500      4.774  -0.105 0.916965
amountAlow      -14.533      3.898  -3.728 0.000451 ***
---
Residual standard error: 15.1 on 56 degrees of freedom
Multiple R-squared:  0.212,Adjusted R-squared:  0.1698
F-statistic: 5.023 on 3 and 56 DF,  p-value: 0.003739
```

ANOVA for Fitted Model

```
> anova(ratsA.lm)
Response: gain
          Df  Sum Sq Mean Sq F value    Pr(>F)
proteinA   2   266.5   133.3  0.5847 0.5606496
amountA    1  3168.3  3168.3 13.9001 0.0004511 ***
Residuals 56 12764.1   227.9
```

- The line for `proteinA` tests whether both indicator variables for protein are needed in the model.
- No evidence of a difference between in mean gain between the three different sources of protein (but we'll revisit this).

ANOVA for Fitted Model (cont.)

What if we try adding the variables in the opposite order:

```
> ratsA1.lm=lm(gain~amountA+proteinA)
> anova(ratsA1.lm)
Analysis of Variance Table

Response: gain
          Df  Sum Sq Mean Sq F value    Pr(>F)
amountA    1  3168.3  3168.3 13.9001 0.0004511 ***
proteinA   2   266.5   133.3  0.5847 0.5606496
Residuals 56 12764.1   227.9
```

- Exactly the same results as before – this is a consequence of using a balanced orthogonal design.

Different Indicator Variables

We can adjust the indicator variables that *R* uses for factors.

- We may decide that we would rather have `cereal` be the baseline for `protein` and `low` be the baseline for `amount`.

```
> proteinB<-factor(protein,levels=c("cereal","pork","beef"))
> contrasts(proteinB)
       pork beef
cereal    0    0
pork      1    0
beef      0    1

> amountB<-factor(amount,levels=c("low","high"))
> contrasts(amountB)
     high
low     0
high    1
```

New Version of the Model

The new version of the model defines the means for the combinations of protein source and amount:

	Protein Source		
amount	beef	cereal	pork
high	$\beta_0 + \beta_B + \beta_H$	$\beta_0 + \beta_H$	$\beta_0 + \beta_P + \beta_H$
low	$\beta_0 + \beta_B$	β_0	$\beta_0 + \beta_P$

Fitted Model

The coefficients for this new version of the model are:

```
> ratsB.lm=lm(gain~proteinB+amountB)
> summary(rats2.lm)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    77.633      3.898  19.916  < 2e-16 ***
proteinBpork    4.200      4.774   0.880 0.382767
proteinBbeef    4.700      4.774   0.984 0.329126
amountBhigh    14.533      3.898   3.728 0.000451 ***
---
Residual standard error: 15.1 on 56 degrees of freedom
Multiple R-squared:  0.212,Adjusted R-squared:  0.1698
F-statistic: 5.023 on 3 and 56 DF,  p-value: 0.003739
```

ANOVA

The ANOVA table for the new version of the model is exactly the same as before (changing the way we define indicator variables does not affect this analysis).

```
> anova(rats2.lm)
Analysis of Variance Table

Response: gain
          Df  Sum Sq Mean Sq F value    Pr(>F)
proteinB   2   266.5   133.3  0.5847 0.5606496
amountB    1  3168.3  3168.3 13.9001 0.0004511 ***
Residuals 56 12764.1   227.9
```

A Third Version

There are many different ways of defining indicator variables for factors. *R* has a built-in method of generating indicator variables that produces the “Helmert contrasts”.

```
> proteinC<-factor(protein,levels=c("pork","beef","cereal"))
> contrasts(proteinC)<-contr.helmert(3)
> contrasts(proteinC)
       [,1] [,2]
pork     -1   -1
beef      1   -1
cereal    0    2
>
> amountC<-factor(amount,levels=c("low","high"))
> contrasts(amountC)<-contr.helmert(2)
> contrasts(amountC)
     [,1]
low    -1
high    1
```

Helmert Contrasts

The first Helmert contrast compares the mean response for the second level of a factor to that for the first level. For each subsequent level, the mean response is compared to average mean response for all preceding levels.

For our example:

- The first contrast generated for `protein` compares the mean response for beef to that for pork.
- The second contrast for `protein` compares the mean response for cereal to the average mean response for beef and pork.
- The contrast for `amount` compares the mean response for high to that for low.

Helmert Version of the Model

Using the Helmert contrasts as our indicator variables defines the means for the combinations of protein source and amount quite differently from the baseline models. Let β_0 be intercept coefficient, β_{P1} and β_{P2} the coefficients for the `protein` contrasts and β_{A1} be the coefficients for the `amount` contrast.

	Protein Source		
amount	beef	cereal	pork
high	$\beta_0 + \beta_{A1} + \beta_{P1} - \beta_{P2}$	$\beta_0 + \beta_{A1} + 2\beta_{P2}$	$\beta_0 + \beta_{A1} - \beta_{P1} - \beta_{P2}$
low	$\beta_0 - \beta_{A1} + \beta_{P1} - \beta_{P2}$	$\beta_0 - \beta_{A1} + 2\beta_{P2}$	$\beta_0 - \beta_{A1} - \beta_{P1} - \beta_{P2}$

Fitted Model: Version 3

```
> ratsC.lm=lm(gain~proteinC+amountC)
> summary(ratsC.lm)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   87.867      1.949  45.081  < 2e-16 ***
proteinC1      0.250      2.387   0.105 0.916965
proteinC2     -1.483      1.378  -1.076 0.286414
amountC1       7.267      1.949   3.728 0.000451 ***
---
Residual standard error: 15.1 on 56 degrees of freedom
Multiple R-squared:  0.212,Adjusted R-squared:  0.1698
F-statistic: 5.023 on 3 and 56 DF,  p-value: 0.003739
```

ANOVA: Version 3

The ANOVA table is the same for this version of our model as it was for the first two versions.

```
> anova(ratsC.lm)
Analysis of Variance Table

Response: gain
          Df  Sum Sq Mean Sq F value    Pr(>F)
proteinC   2   266.5   133.3  0.5847 0.5606496
amountC    1  3168.3  3168.3 13.9001 0.0004511 ***
Residuals 56 12764.1   227.9
```

Equivalent Models

Changing the way we designate the indicator variables for factors doesn't change the model. If we look at the table of fitted values for the six combinations of `protein` and `amount` we get the same estimates for all three versions:

	Protein Source		
amount	beef	cereal	pork
high	96.87	92.17	96.37
low	82.33	77.63	81.83

Interaction?

Our model assumes that the impact of `protein` doesn't depend on the level of `amount` and vice versa.

- The difference between the two levels of `amount` is the same for each level of `protein`.
- The difference between any two levels of `protein` is the same for each level of `amount`.

However, if we examine our conditional plot (slide 7) this doesn't seem to be the case.

The Interaction Model

If we include the interaction between `amount` and `protein`, then the impact of `amount` can be different for each level of `protein` (and vice versa).

- The interaction contrasts for the `amount:protein` interaction are obtained by multiplying each `amount` contrast by each `protein` contrast.
- It doesn't matter which version of our model we start with, adding the `amount:protein` interaction will have the same impact although the coefficients will be different and have different interpretations.

The Interaction Contrasts

Adding the interaction to the first version of our model (baseline model with beef and high as the baseline levels) gives the following table of indicator variables.

		Contrasts				
		Main Effects			Interactions	
protein	amount	cereal	pork	low	cereal:low	pork:low
beef	high	0	0	0	0	0
beef	low	0	0	1	0	0
cereal	high	1	0	0	0	0
cereal	low	1	0	1	1	0
pork	high	0	1	0	0	0
pork	low	0	1	1	0	1

The Model Means

The table of model means (see slide 11) becomes:

		Protein Source	
amount	beef	cereal	pork
high	β_0	$\beta_0 + \beta_C$	$\beta_0 + \beta_P$
low	$\beta_0 + \beta_L$	$\beta_0 + \beta_C + \beta_L + \beta_{C:L}$	$\beta_0 + \beta_P + \beta_L + \beta_{P:L}$

- The difference between the two levels of `amount` is not necessarily the same for each level of `protein`.
- The difference between any two levels of `protein` is not necessarily the same for each level of `amount`.

The Fitted Interaction Model

```
> ratsAI.lm=lm(gain~proteinA*amountA)
> summary(ratsAI.lm)
Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.000e+02  4.632e+00  21.589  < 2e-16 ***
proteinAcereal             -1.410e+01  6.551e+00  -2.152  0.03585 *
proteinApork               -5.000e-01  6.551e+00  -0.076  0.93944
amountAlow                 -2.080e+01  6.551e+00  -3.175  0.00247 **
proteinAcereal:amountAlow   1.880e+01  9.264e+00   2.029  0.04736 *
proteinApork:amountAlow     2.247e-15  9.264e+00   0.000  1.00000
---
Residual standard error: 14.65 on 54 degrees of freedom
Multiple R-squared:  0.2848,Adjusted R-squared:  0.2185
F-statistic:   4.3 on 5 and 54 DF,  p-value: 0.002299
```

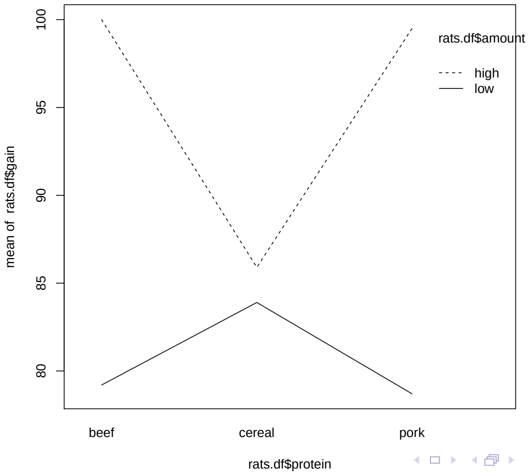
ANOVA Table

```
> anova(ratsAI.lm)
Analysis of Variance Table

Response: gain
                Df  Sum Sq Mean Sq F value    Pr(>F)
proteinA         2   266.5   133.3  0.6211 0.5411319
amountA          1  3168.3  3168.3 14.7666 0.0003224 ***
proteinA:amountA  2  1178.1   589.1  2.7455 0.0731879 .
Residuals       54 11586.0   214.6
```

- Moderate evidence of a `protein:amount` interaction.
- If an interaction is significant it implies that both factors have an impact on the response (keep main effects in the model regardless of whether they show up as significant or not).

Interaction Plot



Binary ANOVA: Plum Root Stock Example

A study was conducted on the reproduction of plum trees by taking cuttings from older trees.

- Half the cuttings were planted immediately while the other half were bedded in sand until spring when they were planted.
- Two lengths of cuttings were used: long (12 cm) and short (6cm).
- A total of 240 cuttings were taken for each of the 4 combinations of planting time and cutting length
- Interested in the survival rate of the cuttings.

Plum Root Stock Data

Length of cutting	Time of planting	Number surviving (out of 240)
short	at once	107
	in spring	31
long	at once	156
	in spring	84

- Note: the model that contains both main effects and their interaction is the maximal model.

Plum Root Stock Analysis

```
> length<-rep(c("short","long"),c(2,2))
> time<-c("at once","spring","at once","spring")
> survive<-c(107,31,156,84)
> time<-C(factor(time),treatment)
> length<-C(factor(length),treatment)
> plum.glm<-glm(cbind(survive,240-survive)~time*length,
                                        family=binomial)
> anova(plum.glm,test="Chisq")
Analysis of Deviance Table
            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                            3    151.019
time         1   97.579         2     53.440 < 2.2e-16 ***
length       1   51.147         1      2.294 8.572e-13 ***
time:length  1    2.294         0      0.000    0.1299
```

Plum Root Stock Analysis (cont.)

The previous analysis suggests that the interaction can be dropped (only a hint of evidence that it is needed). The output for the no interaction model is:

```
> plum2.glm<-glm(cbind(survive,240-survive)~time+length,
                                        family=binomial)
> summary(plum2.glm)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.7138     0.1217   5.867 4.45e-09 ***
timespring   -1.4275     0.1465  -9.747  < 2e-16 ***
lengthshort  -1.0177     0.1455  -6.995 2.64e-12 ***
---
    Null deviance: 151.0193  on 3  degrees of freedom
Residual deviance:   2.2938  on 1  degrees of freedom
```

Fitted Values

The fitted values for the main effects only model are:

```
> plum2.glm$fitted.values
        1         2         3         4
0.4245994 0.1504006 0.6712339 0.3287661
```

Which are very similar to those for the maximal model:

```
> plum.glm$fitted.values
        1         2         3         4
0.4458333 0.1291667 0.6500000 0.3500000
```

Fitted Values (cont.)

To summarize:

Length of cutting	Time of planting	Survival probability observed	Survival probability fitted
short	at once	0.446	0.425
	in spring	0.129	0.150
long	at once	0.650	0.671
	in spring	0.350	0.329

Confidence Intervals

To calculate 95% confidence intervals for the fitted values:

```
> preds<-predict(plum2.glm,se=TRUE)
> endpts<-cbind(preds$fit -1.96*preds$se.fit,
                preds$fit +1.96*preds$se.fit)
> round(exp(endpts)/(1+exp(endpts)),3)
   [,1]  [,2]
1 0.370 0.481
2 0.118 0.190
3 0.617 0.722
4 0.278 0.383
```

Chick Weight Gain Example

An experiment was conducted to compare different diets for feeding chickens. The diets depended on three variables:

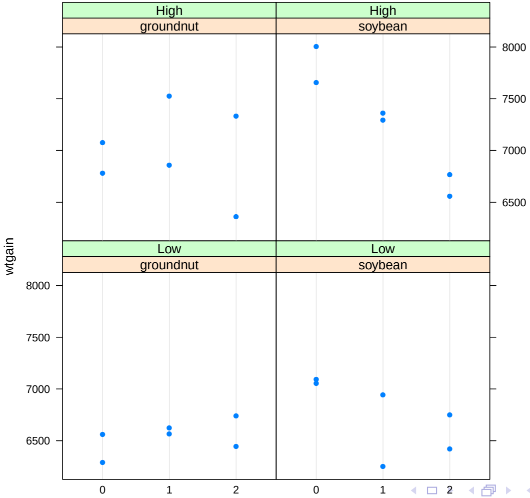
protein source of protein – either groundnut or soybean;

protlev level of protein – either 0, 1 or 2;

fish level of fish solubles – either low or high.

Response variable is `wtgain` and measures the amount of weight gained.

Dotpot of the Data



Three-Way ANOVA

In this example we have three factors. As a result we have the following possible terms for our model:

- The three main effect terms.
- Three two-factor interactions.
- The three-factor interaction.

The interpretation of a k-factor interaction is that the effect of any one of the factors depends on the combination of levels used for the remaining $k-1$ factors.

Specifying Factors

First, we want to designate all the regressors in our data frame as factors and take the opportunity to adjust the order of levels for `fish`.

```
> chickwts.df<-read.table("chickwts.txt",header=TRUE)
> summary(chickwts.df)
     wtgain          protein       protlev     fish
 Min.   :6249   groundnut:12   Min.   :0   High:12
 1st Qu.:6560   soybean  :12   1st Qu.:0   Low :12
 Median :6772                  Median :1
 Mean   :6887                  Mean   :1
 3rd Qu.:7144                  3rd Qu.:2
 Max.   :8005                  Max.   :2
> chickwts.df$protein<-factor(chickwts.df$protein)
> chickwts.df$protlev<-factor(chickwts.df$protlev)
> chickwts.df$fish<-factor(chickwts.df$fish,
                           levels=c("Low","High"))
```

Fitting the Full Model

First we'll fit the model that contains all the possible interactions and look at the anova output.

```
> model1 <- lm(wtgain˜protein*protlev*fish,data=chickwts.df)
> anova(model1)
Response: wtgain
                     Df  Sum Sq Mean Sq F value   Pr(>F)
protein               1  373751  373751  3.7346 0.077244 .
protlev               2  636283  318141  3.1789 0.078011 .
fish                  1 1421553 1421553 14.2044 0.002677 **
protein:protlev       2  858158  429079  4.2874 0.039361 *
protein:fish          1    7176    7176  0.0717 0.793418
protlev:fish          2  308888  154444  1.5432 0.253258
protein:protlev:fish  2   50128   25064  0.2504 0.782426
Residuals            12 1200938  100078
```

Interpreting the ANOVA Table

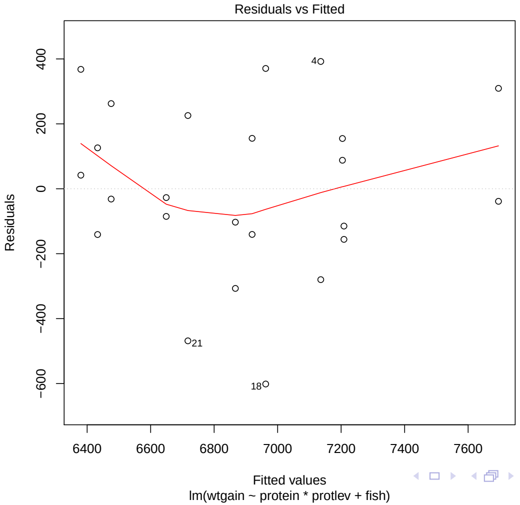
To interpret this ANOVA table at the bottom (with the highest order interaction) and work your way up.

- For an ordinary regression with a balanced design the order that terms are added is not going to affect the hypothesis tests.
- For this table we see that there is no evidence that `protein:protlev:fish`, `protlev:fish` or `protein:fish` are needed in the model.
- Empirical evidence suggests that as the order of an interaction increases it becomes less likely to be important. If you do see a lot of significant high order interactions, try transforming the response.

The Reduced Model

```
> model2 <- lm(wtgain˜protein*protlev+fish,data=chickwts.df)
> summary(model2)
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              6432.88     163.97  39.231  < 2e-16 ***
proteinsoybean            776.00     214.69   3.615  0.00214 **
protlev1                  216.25     214.69   1.007  0.32793
protlev2                   42.75     214.69   0.199  0.84453
fishHigh                  486.75     123.95   3.927  0.00109 **
proteinsoybean:protlev1  -707.75     303.62  -2.331  0.03232 *
proteinsoybean:protlev2  -871.50     303.62  -2.870  0.01061 *
---
Residual standard error: 303.6 on 17 degrees of freedom
Multiple R-squared:  0.6773,Adjusted R-squared:  0.5635
F-statistic: 5.948 on 6 and 17 DF,  p-value: 0.001687
```

Diagnostic Plots



Interpreting the Model

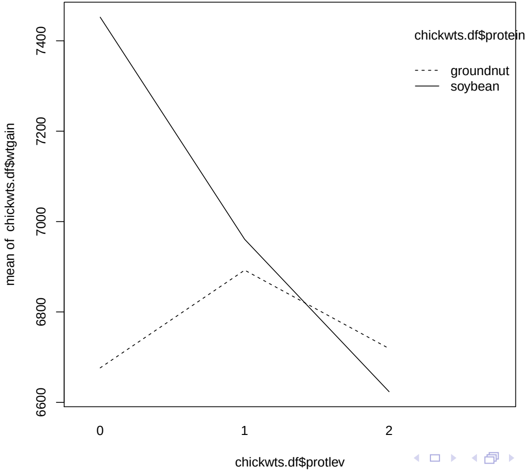
As `fish` doesn't interact with either of the other regressors we can interpret its impact separately.

- A chicken raised on a high fish solubles diet will gain on average 486.75 g(?) more than one on a low fish solubles diet for any combination of the other two factors.

As the `protein:protlev` interaction is significant we should evaluate the impact of these two factors together.

- An interaction plot is very useful in this regard.

Interaction Plot



Conclusions?

Interactions between Factors and Numeric Regressors

Interactions can exist between factors and quantitative regressors.

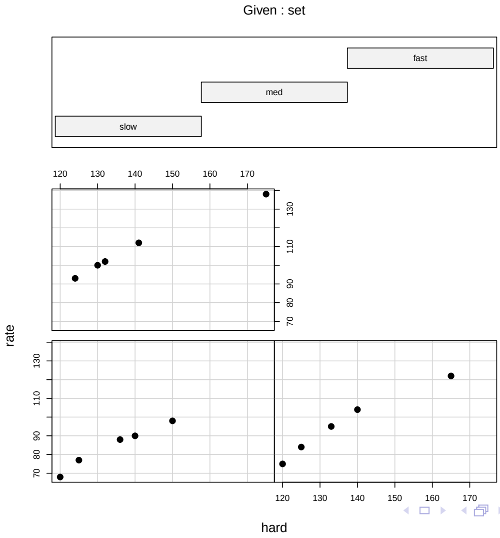
- The coefficient for a factor represents the impact on the response if we change from one level to another. If the factor interacts with a quantitative regressors, this size of the impact depends on the level of this regressor.
- The coefficient of a numeric regressor represents the change in the response per unit change in that regressor (i.e. a slope). If this regressor interacts with a factor, it means that this slope is different for different levels of the factor.

Metal Lathe Example

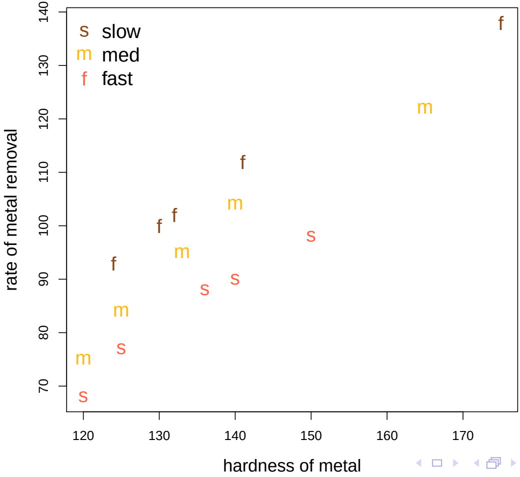
Consider an experiment to measure the rate of metal removal in a machining process on a lathe. The rate depends on:

- the speed setting of the lathe (fast, med or slow): a categorical measurement;
- the hardness of the material being machined: a continuous measurement.

Conditional Plot



Scatter Plot



Same Slopes?

Does the relationship between `rate` and `hardness` have the same slope for each level of `setting`?

- If the slopes are different then we should have the `rate:hardness` interaction in the model.
- If the slopes are the same then this interaction isn't needed.

The No Interaction Model

The baseline contrasts were used to create the indicator variables for `setting`:

```
> contrasts(metal.df$set)
     med fast
slow   0    0
med    1    0
fast   0    1
```

The Model Matrix

```
> metal1.lm <- lm(rate~hard+set,data=metal.df)
> model.matrix(metal1.lm)
   (Intercept) hard setmed setfast
1            1  120      0       0
2            1  140      0       0
3            1  150      0       0
4            1  125      0       0
5            1  136      0       0
6            1  165      1       0
7            1  140      1       0
8            1  120      1       0
9            1  125      1       0
10           1  133      1       0
11           1  175      0       1
12           1  132      0       1
13           1  124      0       1
14           1  141      0       1
15           1  130      0       1
```

The Model Coefficients

So for the baseline model, the relationship between `rate` and `hard` for each level of `set` has been parameterized as:

$$\begin{aligned} \text{set} &= \text{slow} & \qquad \text{rate} &= \beta_0 + \beta_H \text{ hard} \\ \text{set} &= \text{med} & \qquad \text{rate} &= \beta_0 + \beta_M + \beta_H \text{ hard} \\ \text{set} &= \text{fast} & \qquad \text{rate} &= \beta_0 + \beta_F + \beta_H \text{ hard} \end{aligned}$$

The fitted lines are parallel (all have slope β_H) but different intercepts.

Model Matrix for the Full Model

```
> metal2.lm <- lm(rate~hard*set,data=metal.df)
> model.matrix(metal2.lm)
   (Intercept) hard setmed setfast hard:setmed hard:setfast
1            1  120      0       0           0            0
2            1  140      0       0           0            0
3            1  150      0       0           0            0
4            1  125      0       0           0            0
5            1  136      0       0           0            0
6            1  165      1       0         165            0
7            1  140      1       0         140            0
8            1  120      1       0         120            0
9            1  125      1       0         125            0
10           1  133      1       0         133            0
11           1  175      0       1           0          175
12           1  132      0       1           0          132
13           1  124      0       1           0          124
14           1  141      0       1           0          141
15           1  130      0       1           0          130
```

Coefficients for the Full Model

Now the relationship between `rate` and `hard` for each level of `set` has been parameterized as:

$$
\begin{aligned}
\text{set} &= \text{slow} & \text{rate} &= \beta_0 + \beta_H \text{ hard} \\
\text{set} &= \text{med} & \text{rate} &= \beta_0 + \beta_M + (\beta_H + \beta_{H:M}) \text{ hard} \\
\text{set} &= \text{fast} & \text{rate} &= \beta_0 + \beta_F + (\beta_H + \beta_{H:F}) \text{ hard}
\end{aligned}
$$

The fitted lines have different slopes and different intercepts.

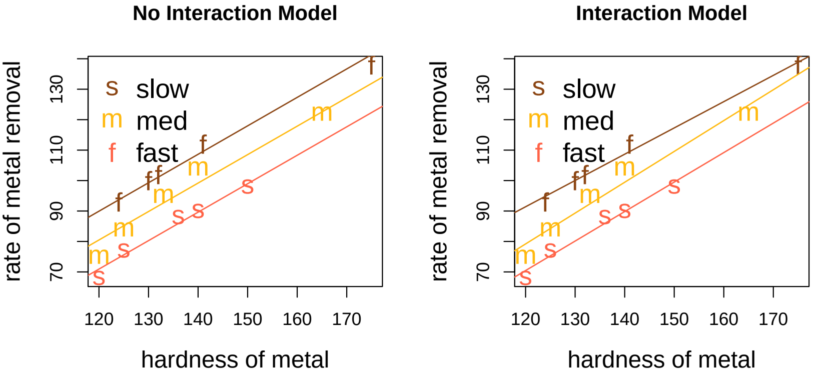
Fitted No Interaction Model

```
> summary(metal1.lm)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -41.17799    6.84927  -6.012 8.77e-05 ***
hard          0.93426    0.05008  18.654 1.13e-09 ***
setmed        9.55777    1.86692   5.120 0.000334 ***
setfast      19.00757    1.88875  10.064 6.94e-07 ***
---
Residual standard error: 2.946 on 11 degrees of freedom
Multiple R-squared:  0.9795,Adjusted R-squared:  0.9739
F-statistic: 175.1 on 3 and 11 DF,  p-value: 1.455e-09
```

Fitted Full Model

```
> summary(metal2.lm)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -45.78282   16.64522  -2.751   0.0225 *
hard          0.96858    0.12364   7.834 2.62e-05 ***
setmed        3.44396   20.25964   0.170   0.8688
setfast      33.60120   19.58902   1.715   0.1204
hard:setmed   0.04415    0.14947   0.295   0.7744
hard:setfast -0.10546    0.14356  -0.735   0.4813
---
Residual standard error: 2.959 on 9 degrees of freedom
Multiple R-squared:  0.9831,Adjusted R-squared:  0.9737
F-statistic: 104.5 on 5 and 9 DF,  p-value: 1.085e-07
```

Which Fit?


No Interaction Model
Interaction Model
rate of metal removal
hardness of metal
70
90
110
130
120
130
140
150
160
170
s slow
m med
f fast

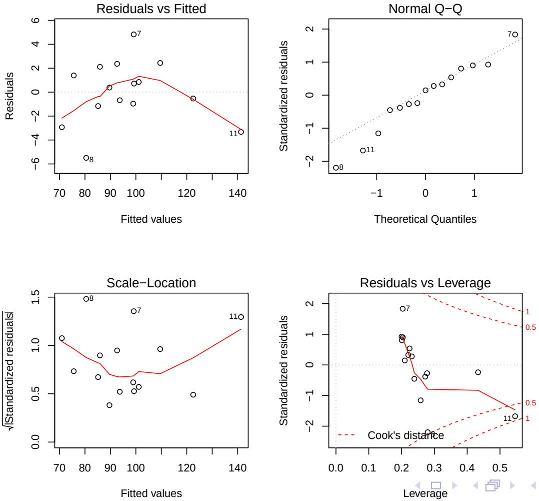
ANOVA Table

To decide we look at the output of `anova` for the full model.

```
> anova(metal2.lm)
          Df Sum Sq Mean Sq  F value     Pr(>F)
hard       1 3679.3  3679.3 420.1866 7.307e-09 ***
set        2  878.8   439.4  50.1834 1.316e-05 ***
hard:set   2   16.6     8.3   0.9504    0.4222
Residuals  9   78.8     8.8
```

- No evidence that the interaction is need – the parallel lines model is fine.

Diagnostic Plots



Conclusions?

Interactions Between Numerical Regressors

Interaction terms between numerical regressors can also be useful in a regression model.

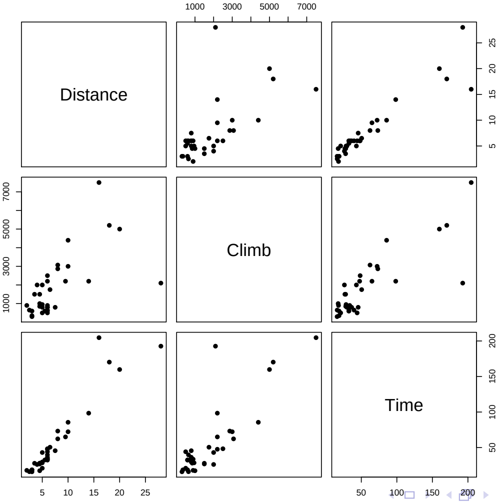
- A bit harder to understand but work much the same way: an interaction allows the impact of one regressor to change with the level of another.
- A two-factor interaction allows the surface to twist in the direction of one regressor as the value of another regressor changes.

Scottish Hill Races

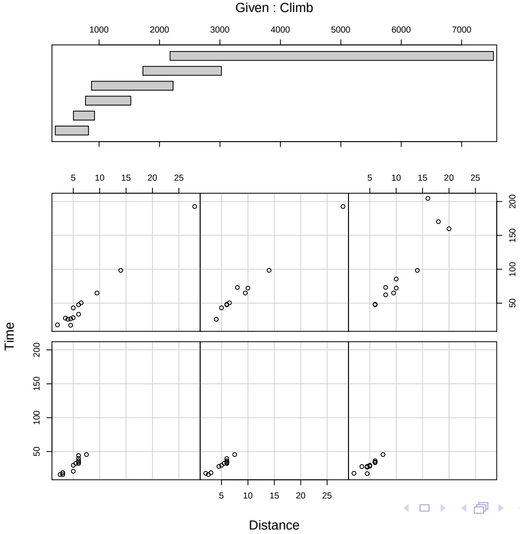
This data consists of the record times for 35 “hill” races in Scotland.

Distance	the length of the race in miles.
Climb	the change in elevation in feet.
Time	the record winning time in minutes.

Pairs Plot


Distance
Climb
Time
1000
3000
5000
7000
25
20
15
10
5
7000
5000
3000
1000
200
150
100
50
5
10
15
20
25
50
100
150
200

Conditional Plot



Interaction Model

The conditional plot suggests that an interaction between `Distance` and `Climb` may be useful.

```
> hills1.lm<-lm(Time˜Climb*Distance,data=hills.df)
> anova(hills1.lm)
Response: Time
               Df Sum Sq Mean Sq  F value    Pr(>F)
Climb           1  59667   59667 1104.372 < 2.2e-16 ***
Distance        1  23997   23997  444.163 < 2.2e-16 ***
Climb:Distance  1    803     803   14.865 0.0005449 ***
Residuals      31   1675      54
```

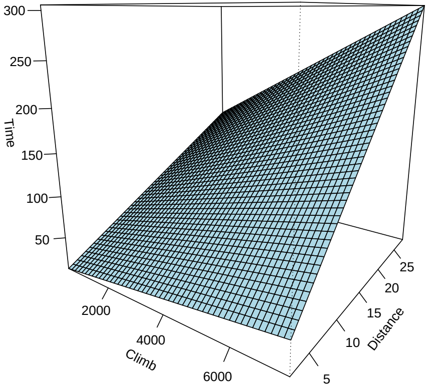
The Model Matrix

```
> model.matrix(hills1.lm)
   (Intercept) Climb Distance Climb:Distance
1            1   650      2.5           1625
2            1  2500      6.0          15000
3            1   900      6.0           5400
4            1   800      7.5           6000
5            1  3070      8.0          24560
6            1  2866      8.0          22928
7            1  7500     16.0         120000
8            1   800      6.0           4800
9            1   800      5.0           4000
================================================
31           1  4400     10.0          44000
32           1   600      6.0           3600
33           1  5200     18.0          93600
34           1   850      4.5           3825
35           1  5000     20.0         100000
```

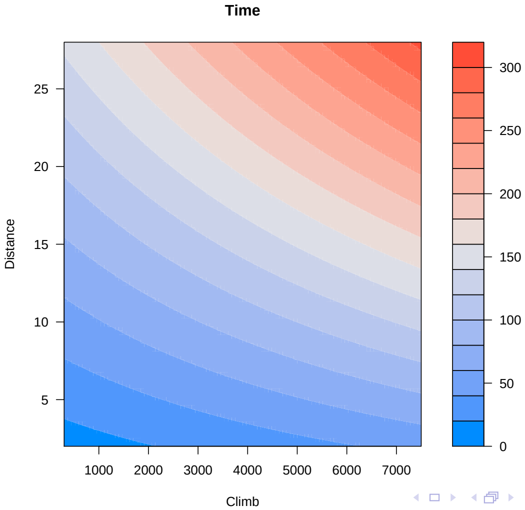
The Fitted Model

```
> summary(hills1.lm)
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.3532285  3.9121887  -0.090 0.928638
Climb           0.0035217  0.0023686   1.487 0.147156
Distance        4.9290166  0.4750168  10.377 1.32e-11 ***
Climb:Distance  0.0006731  0.0001746   3.856 0.000545 ***
---
Residual standard error: 7.35 on 31 degrees of freedom
Multiple R-squared:  0.9806,Adjusted R-squared:  0.9787
F-statistic: 521.1 on 3 and 31 DF,  p-value: < 2.2e-16
```

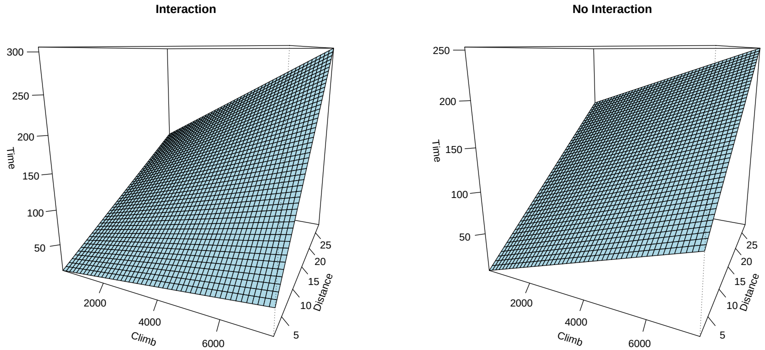
Wireframe Plot



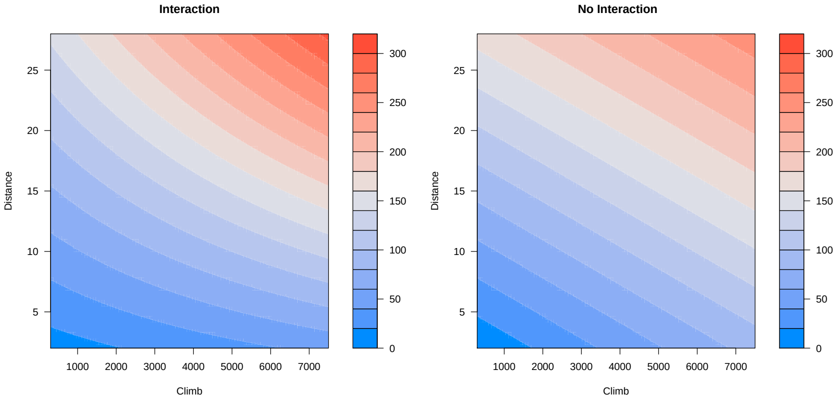
Filled Contour Plot



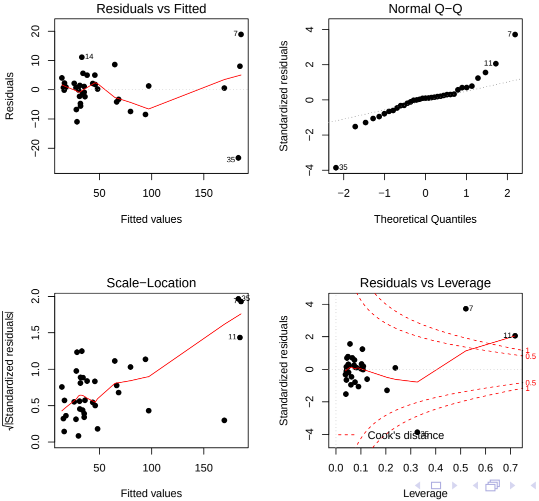
Comparison of Surfaces



Comparison of Surfaces



Diagnostic Plots



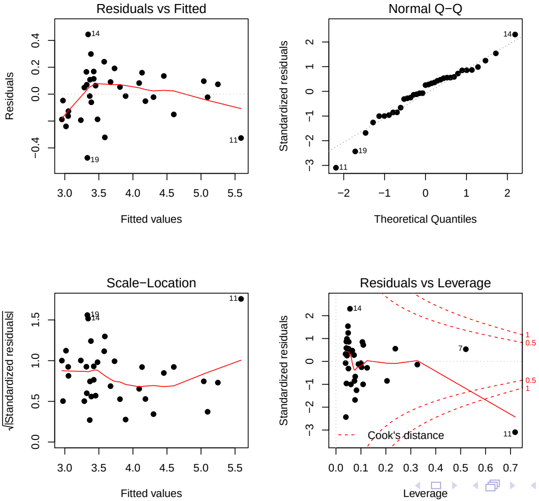
Log the Response?

Might try using the log of the response – expect variability to increase as the expected time increases.

```
> hills3.lm<-lm(log(Time)~Climb*Distance,data=hills.df)
> anova(hills3.lm)
Analysis of Variance Table

Response: log(Time)
               Df  Sum Sq Mean Sq F value    Pr(>F)
Climb           1 11.5876 11.5876 293.087 < 2.2e-16 ***
Distance        1  4.0030  4.0030 101.250 2.762e-11 ***
Climb:Distance  1  0.4301  0.4301  10.879  0.002447 **
Residuals      31  1.2256  0.0395
```

Diagnostic Plots



Fitted Model

```
> summary(hills3.lm)
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.517e+00  1.058e-01  23.780  < 2e-16 ***
Climb           3.686e-04  6.407e-05   5.753 2.49e-06 ***
Distance        1.147e-01  1.285e-02   8.929 4.46e-10 ***
Climb:Distance -1.558e-05  4.722e-06  -3.298  0.00245 **
---
Residual standard error: 0.1988 on 31 degrees of freedom
Multiple R-squared:  0.9289,Adjusted R-squared:  0.9221
F-statistic: 135.1 on 3 and 31 DF,  p-value: < 2.2e-16
```