Process Example: Regression with Interaction

The response is the yield (Y) of a certain process, the predictors are temperature and concentration. We consider models with and without interaction between temp and conc. In this example, we treat both temp and conc as continuous, but in practice, with just two levels for each predictor, I would probably run this as a 2way ANOVA (treating temp and conc as categorical).

In this example we look at several models:

1. Model1: Simple linear regression using just Temp.
2. Model2: Simple linear regression using just Conc.
3. Model3: Multiple regression with both Temp and Conc (but no interaction).
4. Model4: Multiple regression including Temp, Conc and Temp:Conc interaction. "By hand" for illustration.
5. Model5: Multiple regression including Temp, Conc and Temp:Conc interaction. Same as Model4.

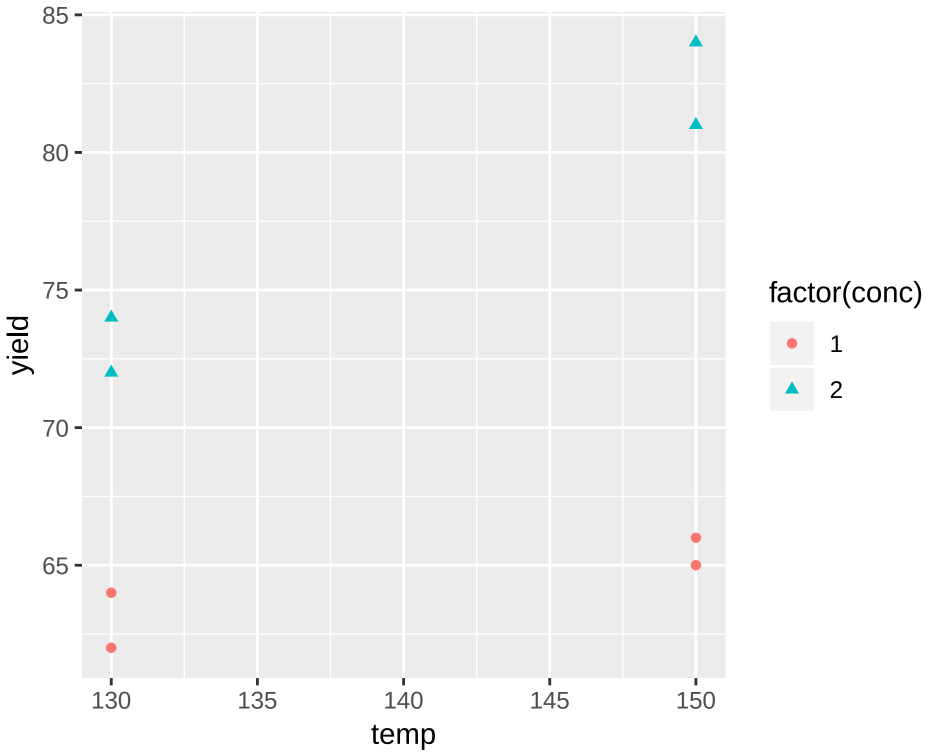
```
library(ggplot2)
library(dplyr)
library(effects)
library(gridExtra)
Process <- read.csv("~/Dropbox/STAT512/Lectures/MultReg2/MR2_Process.csv")
Process
```

```
##   yield conc temp
## 1    62    1  130
## 2    64    1  130
## 3    74    2  130
## 4    72    2  130
## 5    65    1  150
## 6    66    1  150
## 7    81    2  150
## 8    84    2  150
```

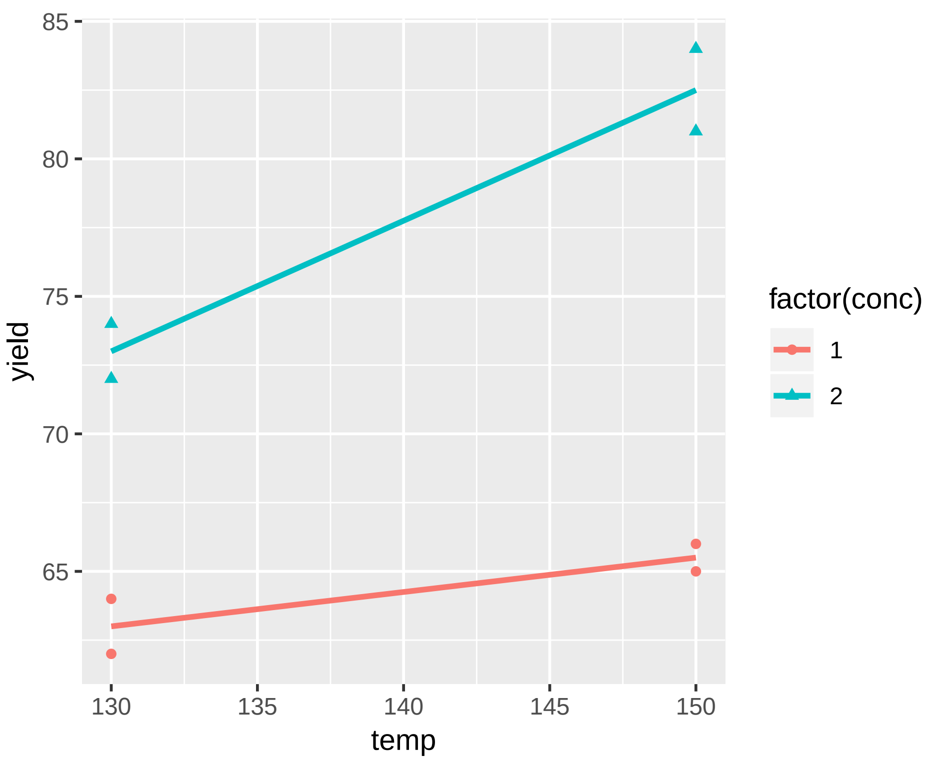
Scatterplot with color coded observations

Here we use the qplot() function from ggplot2. ggplot2 allows us to build plots in "layers".

```
p <- qplot(temp, yield, shape = factor(conc), color = factor(conc), data = Process)
p
```




```r
p + geom_smooth(method = lm, se = FALSE)
```



Simple and Multiple Regressions

We start by looking at the simple means for each treatment combination and the pairwise correlations. Then fit some models.

```r
aggregate(yield ~ temp + conc, data = Process, FUN = mean)
```

```
##   temp conc yield
## 1  130    1  63.0
## 2  150    1  65.5
## 3  130    2  73.0
## 4  150    2  82.5
```

```r
cor(Process)
```

```
##           yield      conc      temp
## yield 1.0000000 0.8806429 0.3913968
## conc  0.8806429 1.0000000 0.0000000
## temp  0.3913968 0.0000000 1.0000000
```

```r
Model1 <- lm(yield ~ temp, data = Process)
summary(Model1)
```

```
##
## Call:
## lm(formula = yield ~ temp, data = Process)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9.00  -6.50   0.00   6.25  10.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   29.000     40.416   0.718    0.500
## temp           0.300      0.288   1.042    0.338
##
## Residual standard error: 8.145 on 6 degrees of freedom
## Multiple R-squared:  0.1532, Adjusted R-squared:  0.01206
## F-statistic: 1.085 on 1 and 6 DF,  p-value: 0.3376
```

```r
Model2 <- lm(yield ~ conc, data = Process)
summary(Model2)
```

```
##
## Call:
## lm(formula = yield ~ conc, data = Process)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.750 -2.625  0.250  2.125  6.250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.750      4.688  10.825 3.68e-05 ***
## conc          13.500      2.965   4.553  0.00388 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.193 on 6 degrees of freedom
## Multiple R-squared:  0.7755, Adjusted R-squared:  0.7381
## F-statistic: 20.73 on 1 and 6 DF,  p-value: 0.003879
```

```r
Model3 <- lm(yield ~ temp + conc, data = Process)
summary(Model3)
```

```
##
## Call:
## lm(formula = yield ~ temp + conc, data = Process)
##
## Residuals:
##     1     2     3     4     5     6     7     8
##  0.75  2.75 -0.75 -2.75 -2.25 -1.25  0.25  3.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.75000   13.13488   0.666  0.53480
## temp         0.30000    0.09152   3.278  0.02200 *
## conc        13.50000    1.83030   7.376  0.00072 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.588 on 5 degrees of freedom
## Multiple R-squared:  0.9287, Adjusted R-squared:  0.9002
## F-statistic: 32.57 on 2 and 5 DF,  p-value: 0.001356
```

Adding Interaction term "By Hand" (for illustration)

We use the mutate() function from dplyr to calculate a new variable tc (=temp*conc) and add this term to the model.

```r
Process <- mutate(Process, tc = temp*conc)
str(Process)
```

```
## 'data.frame':    8 obs. of  4 variables:
##  $ yield: int  62 64 74 72 65 66 81 84
##  $ conc : int  1 1 2 2 1 1 2 2
##  $ temp : int  130 130 130 130 150 150 150 150
##  $ tc   : int  130 130 260 260 150 150 300 300
```

```r
Model4 <- lm(yield ~ temp + conc + tc, data = Process)
summary(Model4)
```

```
##
## Call:
## lm(formula = yield ~ temp + conc + tc, data = Process)
##
## Residuals:
##    1    2    3    4    5    6    7    8
## -1.0  1.0  1.0 -1.0 -0.5  0.5 -1.5  1.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  82.2500    23.5385   3.494   0.0250 *
## temp          -0.2250     0.1677  -1.342   0.2508
## conc         -35.5000    14.8871  -2.385   0.0756 .
## tc             0.3500     0.1061   3.300   0.0299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.5 on 4 degrees of freedom
## Multiple R-squared:  0.9809, Adjusted R-squared:  0.9665
## F-statistic:  68.3 on 3 and 4 DF,  p-value: 0.0006831
```

Adding Interaction term (Standard Approach)

In practice, we do not need to create the interaction term in advance. Note that the predicted values exactly equal the means, because we have a saturated model.

```r
Model5 <- lm(yield ~ temp*conc, data = Process)
#Equivalent to
#lm(yield ~ temp + conc + temp:conc, data = Process)
summary(Model5)
```

```
##
## Call:
## lm(formula = yield ~ temp * conc, data = Process)
##
## Residuals:
##    1    2    3    4    5    6    7    8
## -1.0  1.0  1.0 -1.0 -0.5  0.5 -1.5  1.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  82.2500    23.5385   3.494   0.0250 *
## temp          -0.2250     0.1677  -1.342   0.2508
## conc         -35.5000    14.8871  -2.385   0.0756 .
## temp:conc      0.3500     0.1061   3.300   0.0299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.5 on 4 degrees of freedom
## Multiple R-squared:  0.9809, Adjusted R-squared:  0.9665
## F-statistic:  68.3 on 3 and 4 DF,  p-value: 0.0006831
```

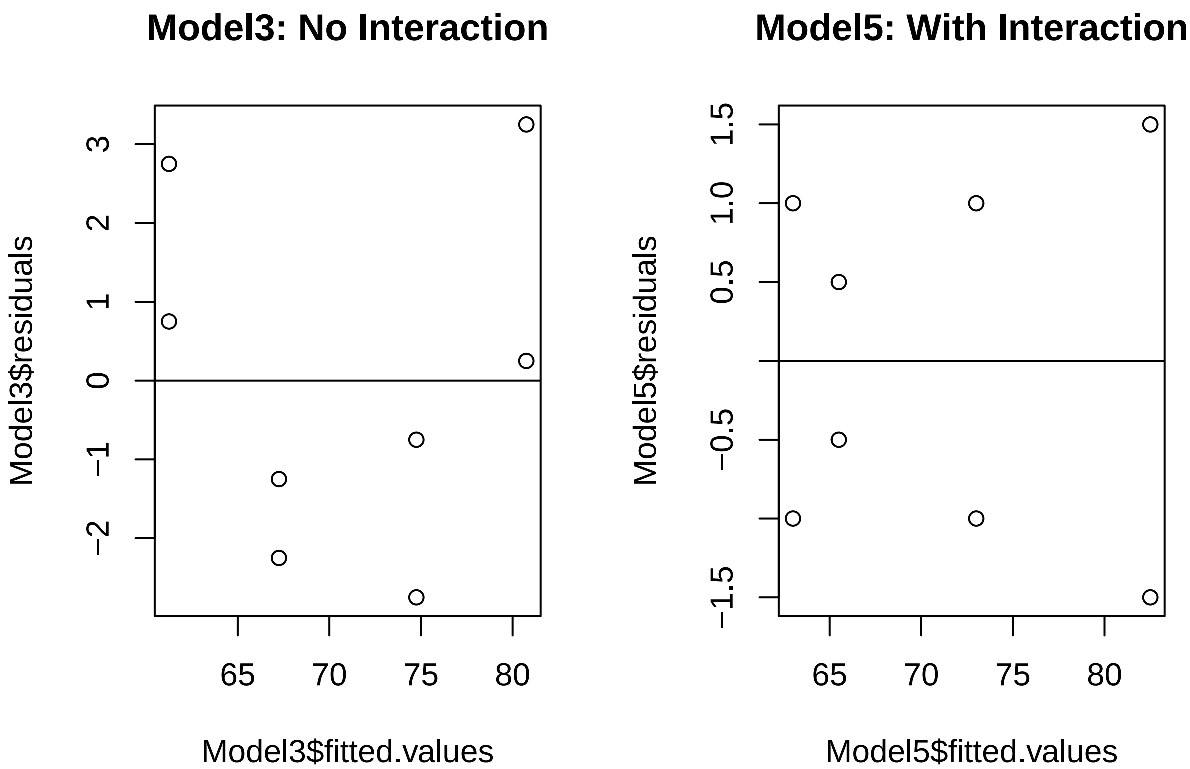
```r
Temp <- data.frame(Yhat = predict(Model5), Process)
Temp
```

```
##   Yhat yield conc temp  tc
## 1 63.0    62    1  130 130
## 2 63.0    64    1  130 130
## 3 73.0    74    2  130 260
## 4 73.0    72    2  130 260
## 5 65.5    65    1  150 150
## 6 65.5    66    1  150 150
## 7 82.5    81    2  150 300
## 8 82.5    84    2  150 300
```

Diagnostic Plots

Resids vs Fitted values for models with and without interaction. Note that I could also have used the plot() command directly (ex: plot(Model3)).

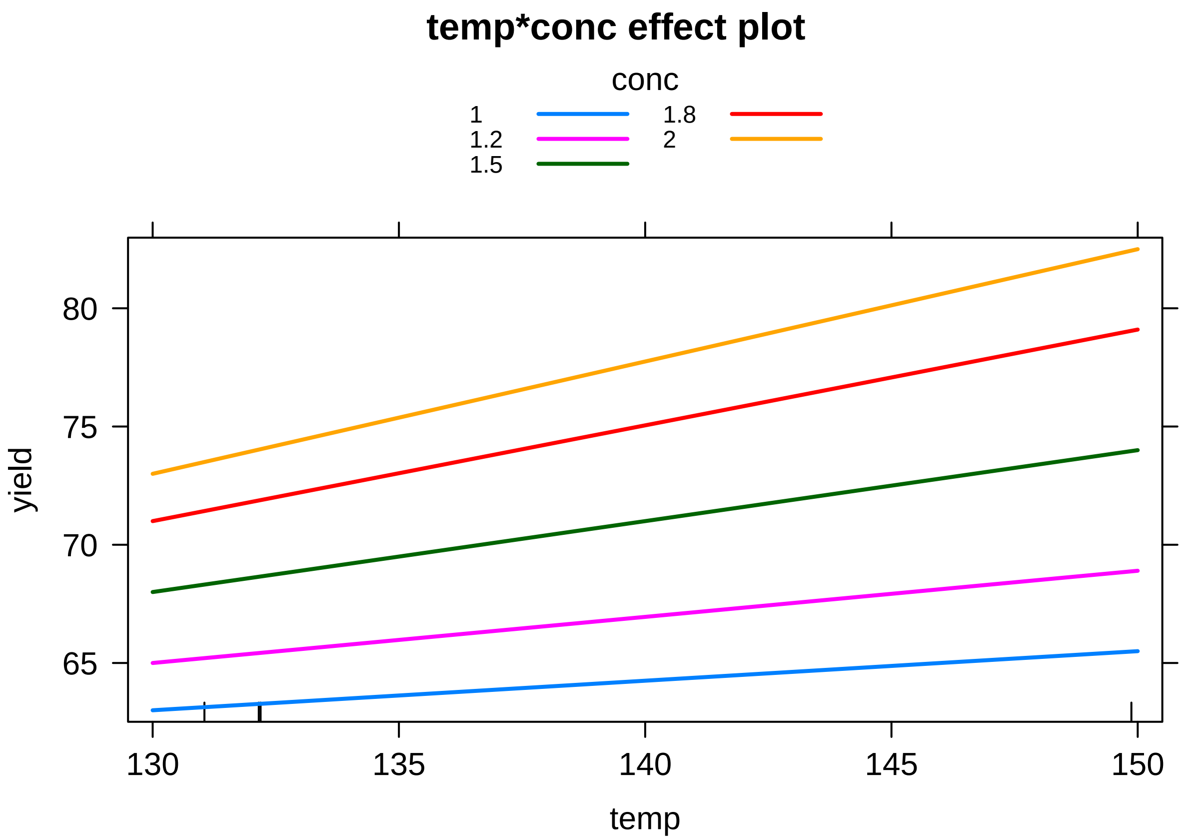
```
par(mfrow = c(1, 2))
plot(Model3$residuals ~ Model3$fitted.values)
abline(h = 0)
title("Model3: No Interaction")
plot(Model5$residuals ~ Model5$fitted.values)
abline(h = 0)
title("Model5: With Interaction")
```



Visualizing the fitted models

Note in Approach#2, that I use yet another approach for adding a column to a data.frame.

```
#Approach#1: Using the effects package
plot(effect(term = "temp:conc", mod = Model5, default.levels = 5),
     multiline = TRUE)
```



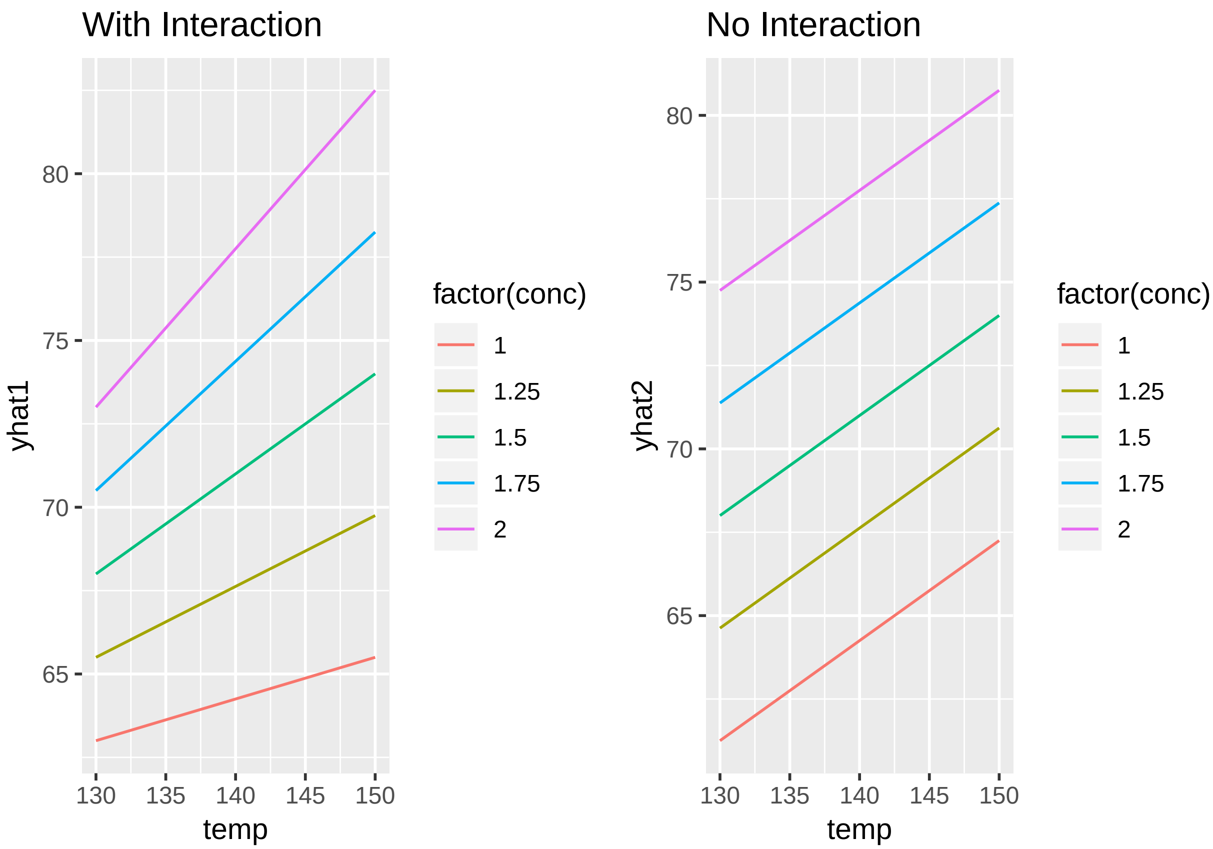

```r
#Approach#2: Using predict() and ggplot2
grid <- with(Process, expand.grid(
  temp = seq(min(temp), max(temp), by = 2),
  conc = seq(min(conc), max(conc), by = 0.25)
  ))
grid$yhat1 <- predict(Model5, newdata = grid)
grid$yhat2 <- predict(Model3, newdata = grid)
colnames(grid)
```

```
## [1] "temp"  "conc"  "yhat1" "yhat2"
```

```r
g1 <- qplot(temp, yhat1, color = factor(conc), geom = "blank", data = grid) +
  geom_line(data = grid) + ggtitle("With Interaction")

g2 <- qplot(temp, yhat2, color = factor(conc), geom = "blank", data = grid) +
  geom_line(data = grid) + ggtitle("No Interaction")

grid.arrange(g1, g2, ncol = 2)
```


With Interaction
No Interaction
yhat1
yhat2
temp
factor(conc)
1
1.25
1.5
1.75
2
65
70
75
80
130
135
140
145
150