Tool Example: Regression with Polynomial and Interaction Terms

This is an "overkill" example where we fit a complicated model to a very small data set (n = 18 obs). The response is tool life. The predictors are cutting angle and speed.

The goal of this example is to illustrate some graphing techniques (contour and perspective plots), but even these plots are not commonly required.

```r
library(ggplot2)
Tool <- read.csv("C:/hess/STAT512/RNotes/MultReg2/MR2_Tool.csv", header=TRUE)
str(Tool)
```

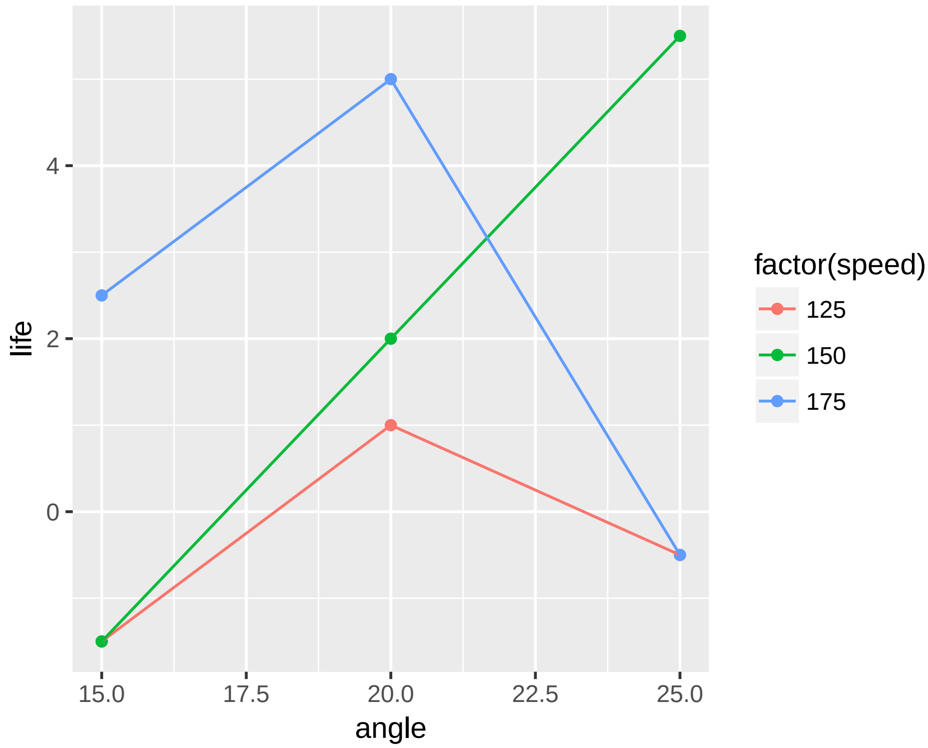
```
## 'data.frame':    18 obs. of  3 variables:
##  $ angle: int  15 15 15 15 15 15 20 20 20 20 ...
##  $ speed: int  125 125 150 150 175 175 125 125 150 150 ...
##  $ life : int  -2 -1 -3 0 2 3 0 2 1 3 ...
```

Summary Statistics, Plot and Model Fitting

```r
ToolMeans <- aggregate(life ~ angle + speed, data = Tool, FUN = mean)
ToolMeans
```

```
##   angle speed life
## 1    15   125 -1.5
## 2    20   125  1.0
## 3    25   125 -0.5
## 4    15   150 -1.5
## 5    20   150  2.0
## 6    25   150  5.5
## 7    15   175  2.5
## 8    20   175  5.0
## 9    25   175 -0.5
```

```r
p <- qplot(angle, life, colour = factor(speed), data = ToolMeans)
p + geom_line()
```




```
Model <- lm(life ~ speed + I(speed^2) + angle + I(angle^2) + I(angle*speed)
  + I(angle^2*speed) +  I(angle*speed^2) + I(angle^2*speed^2), data = Tool)
summary(Model)
```

```
##
## Call:
## lm(formula = life ~ speed + I(speed^2) + angle + I(angle^2) +
##     I(angle * speed) + I(angle^2 * speed) + I(angle * speed^2) +
##     I(angle^2 * speed^2), data = Tool)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -1.5   -0.5    0.0    0.5    1.5
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.068e+03  7.022e+02  -1.521   0.1626
## speed                 1.448e+01  9.503e+00   1.524   0.1619
## I(speed^2)           -4.960e-02  3.164e-02  -1.568   0.1514
## angle                 1.363e+02  7.261e+01   1.877   0.0932 .
## I(angle^2)           -4.080e+00  1.810e+00  -2.254   0.0507 .
## I(angle * speed)     -1.864e+00  9.827e-01  -1.897   0.0903 .
## I(angle^2 * speed)    5.600e-02  2.450e-02   2.285   0.0481 *
## I(angle * speed^2)    6.400e-03  3.272e-03   1.956   0.0822 .
## I(angle^2 * speed^2) -1.920e-04  8.158e-05  -2.353   0.0431 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.202 on 9 degrees of freedom
```

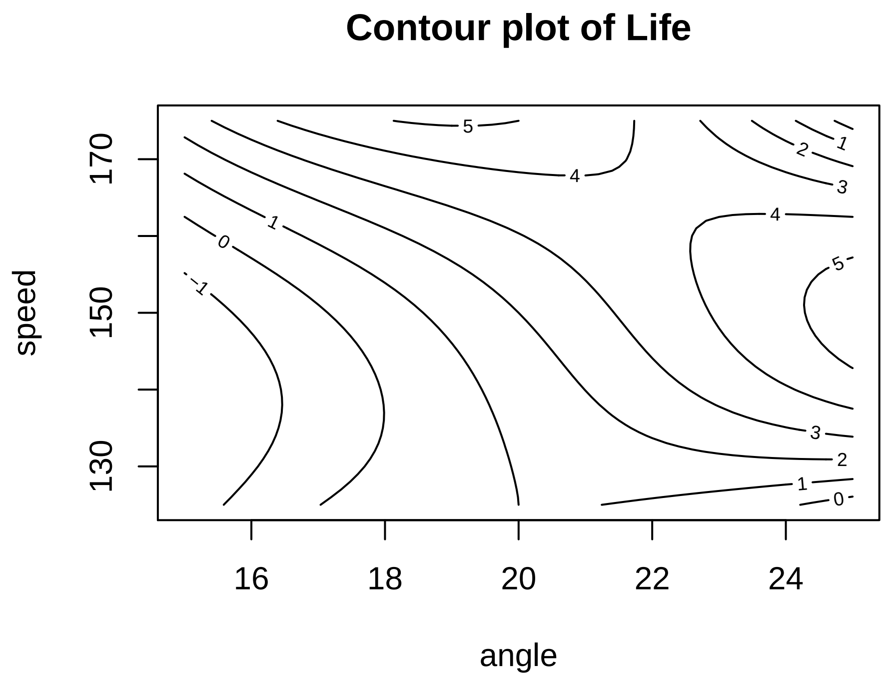
```
## Multiple R-squared:  0.8952, Adjusted R-squared:  0.802
## F-statistic: 9.606 on 8 and 9 DF,  p-value: 0.001337
```

```
Tool$yhat <- predict(Model)
Tool
```

```
##    angle speed life yhat
## 1     15   125   -2 -1.5
## 2     15   125   -1 -1.5
## 3     15   150   -3 -1.5
## 4     15   150    0 -1.5
## 5     15   175    2  2.5
## 6     15   175    3  2.5
## 7     20   125    0  1.0
## 8     20   125    2  1.0
## 9     20   150    1  2.0
## 10    20   150    3  2.0
## 11    20   175    4  5.0
## 12    20   175    6  5.0
## 13    25   125   -1 -0.5
## 14    25   125    0 -0.5
## 15    25   150    5  5.5
## 16    25   150    6  5.5
## 17    25   175    0 -0.5
## 18    25   175   -1 -0.5
```

Visualizing the fitted model

```
# Start 3D plotting by defining the plotting the grid:
a <- seq(15, 25, by = 0.2)
s <- seq(125, 175)
# Create a function to compute the height of the surface:
pfun1 <- function(a,s) -1.068e+03 + 1.448e+01*s -4.960e-02*s^2 +
  1.363e+02*a -4.080e+00*a^2  -1.864e+00*a*s   +
  5.600e-02*a^2*s +  6.400e-03*a*s^2 -1.920e-04*a^2*s^2
# Vectorize the function for use in the "outer" command
pfun1v <- Vectorize(pfun1, c("a", "s"))
contour(a, s, outer(a, s, pfun1v), xlab = "angle", ylab = "speed", main = "Contour plot of Life")
```




```r
#  Do a "perspective" plot, axis rotated for visibility
persp(a, s, outer(a, s, pfun1v), xlab = "angle", ylab = "speed", zlab = "Life", theta=30, phi=30)
```