Week of March 29 Tutorials

Neil Montgomery

March 29, 2016

Topic this week

We'll do some exercises related to higher order terms and dummy variables. In some sense these exercises are also just more exercises relating to multiple regression.

The following textbook exercises specifically mention higher order terms and dummy variables. Some of them deal with fake datasets with ridiculously small sample size, so don't take those ones too seriously. 12.6, 12.7, 12.8(a), 12.9, 12.10, 12.45, 12.46, 12.51, 12.59.

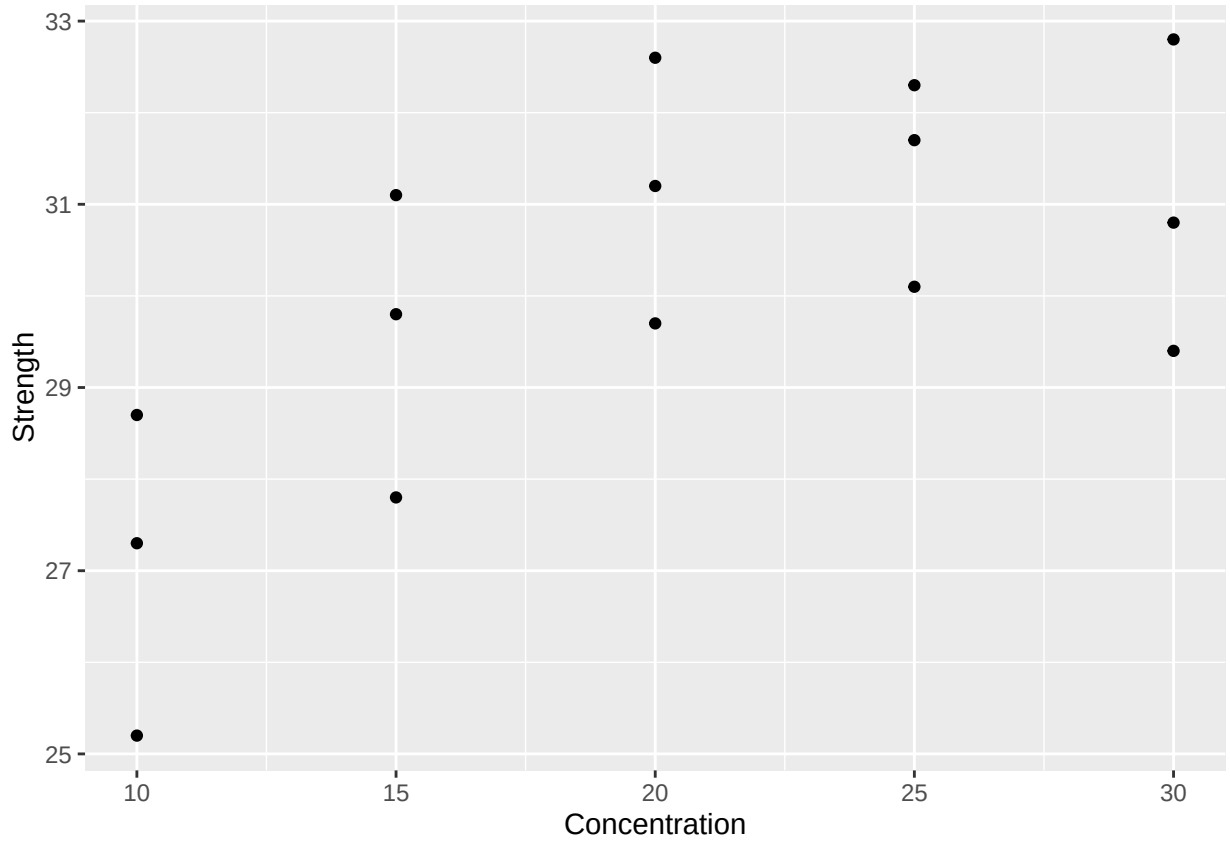
This tutorial will look at a one of those questions and one with simulated data.

Your "lab" this week consists mainly of solving those questions using R yourself. The second assignment will come out shortly and will involve fitting multiple regression models so it will be a good idea to get some practice in.

12.8(a)

The input is an additive concentration and the output is compressive strength of an alloy.

```
library(rio)
library(dplyr)
library(ggplot2)
alloy <- import("Ex12.08.txt")
alloy %>%
  ggplot(aes(x=Concentration, y=Strength)) + geom_point()
```

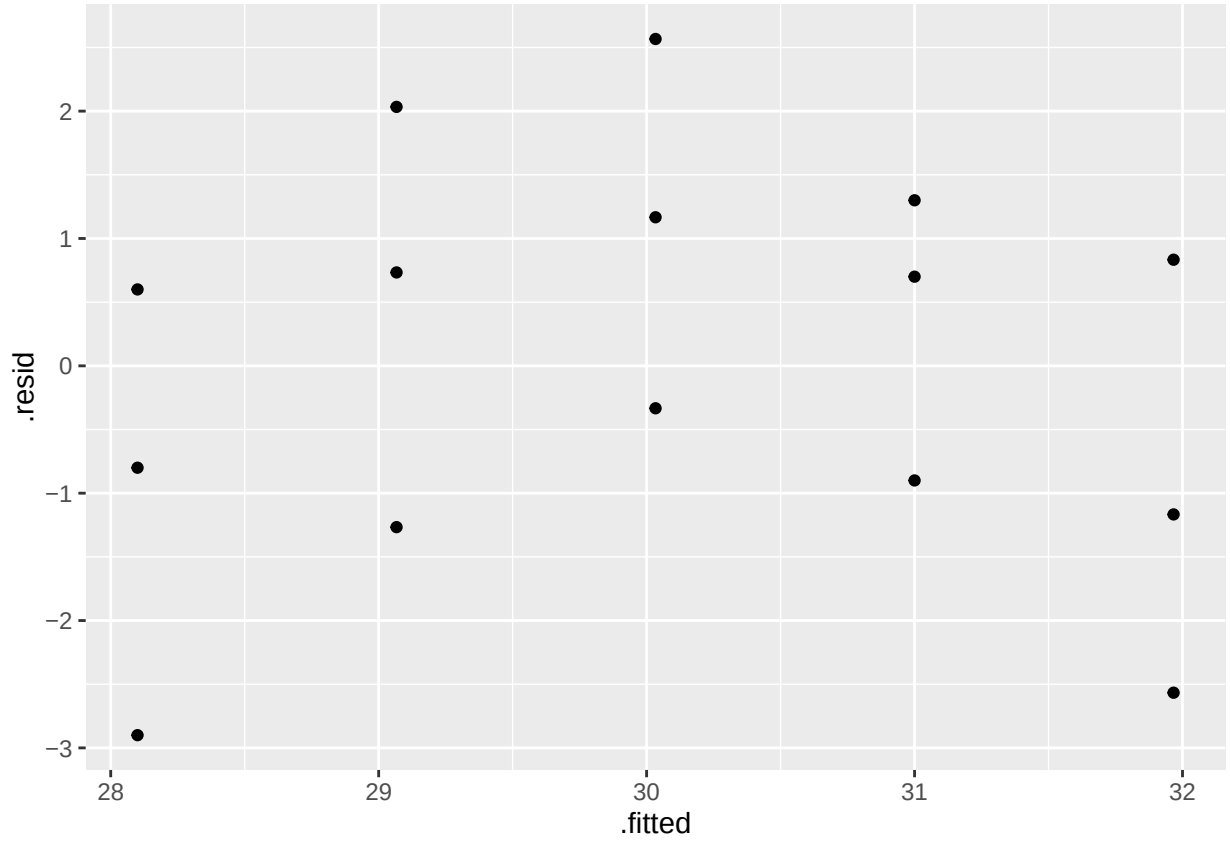


The plot shows a pretty clear curve. If we try the simple linear model things might look OK but the plot of residuals versus fitted values shows the problem more clearly:

```
alloy %>%
  lm(Strength ~ Concentration, data=.) -> alloy_lm
summary(alloy_lm)
```

```
##
## Call:
## lm(formula = Strength ~ Concentration, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.900 -1.033  0.600  1.000  2.567
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.16667    1.27786  20.477 2.82e-11 ***
## Concentration  0.19333    0.06024   3.209  0.00684 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 13 degrees of freedom
## Multiple R-squared:  0.4421, Adjusted R-squared:  0.3992
## F-statistic:  10.3 on 1 and 13 DF,  p-value: 0.006842
```

```
alloy_lm %>%
  ggplot(aes(x=.fitted, y=.resid)) + geom_point()
```



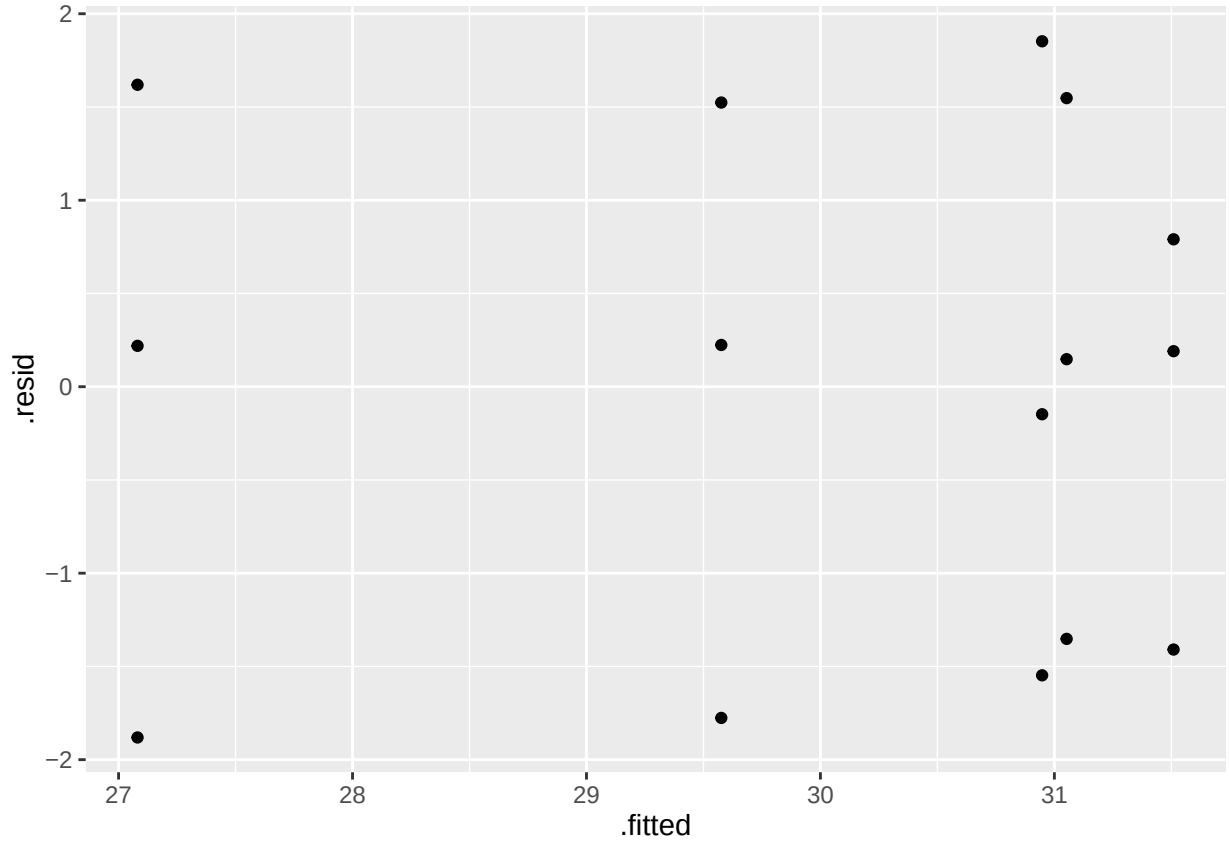
So let's "update" the model by adding the square of `Concentration`. A good way of doing that in `R` is to use the `update` function. Everything is better now.

```
alloy_lm2 <- update(alloy_lm, . ~ . + I(Concentration^2), data=alloy)
summary(alloy_lm2)
```

```
##
## Call:
## lm(formula = Strength ~ Concentration + I(Concentration^2), data = alloy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8809 -1.3809  0.1905  1.1571  1.8524
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        19.033333   3.277549   5.807 8.38e-05 ***
## Concentration       1.008571   0.356431   2.830   0.0152 *
## I(Concentration^2) -0.020381   0.008815  -2.312   0.0393 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.428 on 12 degrees of freedom
## Multiple R-squared:  0.614,  Adjusted R-squared:  0.5497
## F-statistic: 9.545 on 2 and 12 DF,  p-value: 0.003307
```

```
alloy_lm2 %>%
  ggplot(aes(x=.fitted, y=.resid)) + geom_point()
```



Consider the following:

1. The change in R^2 from one model to the next.
2. The meaning and interpretation of the invidual p-values in this case.
3. Try to reconstruct the ANOVA table for the second case given SST=63.4133333.

Interaction simulated example

The textbook lacks any good interaction questions, so I'll make one up with simulated data. Here is *The Truth*:

```
# The Truth
set.seed(1)
x1 <- runif(100, 0, 1)
x2 <- runif(100, 2, 3)
y <- 0.5 + 1.2*x1 - 0.9*x2 + 0.4*x1*x2 + rnorm(100, 0, 0.3)
intdata <- data.frame(y, x1, x2)
```

Here is the regression output for 4 models: the two one-term models, the model with both terms, and the model with both terms plus interaction. I didn't reproduce them here, but all the residual plots were beautiful and perfect.

Consider the following: 1. The R^2 values for all the models. 2. The individual and overall p-values for the models. 3. How to interpret the individual p-values for the model with the interaction.

```
summary(lm(y ~ x1, data = intdata))
```

```
##
## Call:
## lm(formula = y ~ x1, data = intdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69298 -0.22267 -0.02695  0.23394  0.92906
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.76312    0.07537  -23.39   <2e-16 ***
## x1           2.18100    0.12944   16.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3446 on 98 degrees of freedom
## Multiple R-squared:  0.7434, Adjusted R-squared:  0.7408
## F-statistic: 283.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(y ~ x2, data = intdata))
```

```
##
## Call:
## lm(formula = y ~ x2, data = intdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41807 -0.47406 -0.04565  0.50265  1.39243
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9390     0.6165   1.523   0.1310
## x2           -0.6247     0.2435  -2.565   0.0118 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6586 on 98 degrees of freedom
## Multiple R-squared:  0.06293,    Adjusted R-squared:  0.05337
## F-statistic: 6.582 on 1 and 98 DF,  p-value: 0.01182
```

```
summary(lm(y ~ x1 + x2, data = intdata))
```

```
##
```

```
## Call:
## lm(formula = y ~ x1 + x2, data = intdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54419 -0.19993 -0.04858  0.16695  0.70828
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1038     0.2811  -0.369    0.713
## x1            2.1924     0.1108  19.787  < 2e-16 ***
## x2           -0.6615     0.1091  -6.064 2.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.295 on 97 degrees of freedom
## Multiple R-squared:  0.8139, Adjusted R-squared:  0.8101
## F-statistic: 212.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
# Notice how this model automatically includes all lower order terms
summary(lm(y ~ x1*x2, data = intdata))
```

```
##
## Call:
## lm(formula = y ~ x1 * x2, data = intdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57335 -0.18431 -0.03182  0.17034  0.71227
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.20625    0.67365   1.791   0.0765 .
## x1          -0.04823    1.05655  -0.046   0.9637
## x2          -1.17945    0.26553  -4.442 2.39e-05 ***
## x1:x2        0.88458    0.41489   2.132   0.0356 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2897 on 96 degrees of freedom
## Multiple R-squared:  0.8224, Adjusted R-squared:  0.8168
## F-statistic: 148.1 on 3 and 96 DF,  p-value: < 2.2e-16
```