Nonlinearity: Polynomial Effects

2018-12-07

Preparation

In this set of notes, you will learn one method of dealing with nonlinearity. Specifically, we will look at the including polynomial effects into a model. The data we will use in this set of notes, *mnSchools.csv*, contains institutional data for several Minnesota colleges and universities collected in 2011. The variables are:

- `id`: Institution ID number
- `name`: Institution name
- `gradRate`: Six-year graduation rate. This measure represents the proportion of first-time, full-time, bachelor's or equivalent degree-seeking students who started in Fall 2005 and graduated within 6 years.
- `public`: Dummy variable indicating educational sector (0 = private institution; 1 = public institution)
- `sat`: Estimated median SAT score for incoming freshmen at the institution
- `tuition`: Cost of attendance for full-time, first-time degree/certificate-seeking in-state undergraduate students living on campus for academic year 2013-14.

These source of these data is: http://www.collegeresults.org. We will examine use these data to examine if (and how) academic "quality" of the student-body (measured by SAT score) is related to institutiional graduation rate.

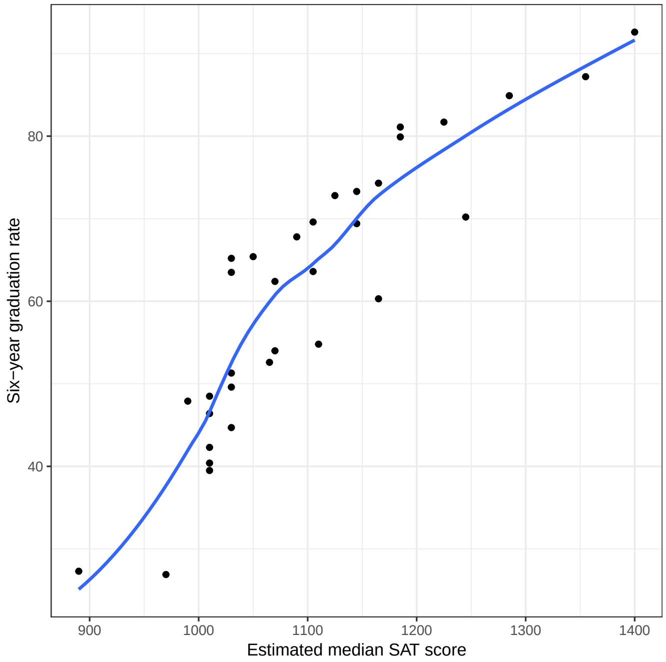
```r
# Load libraries
library(broom)
library(dplyr)
library(ggplot2)
library(readr)
library(sm)
library(tidyr)

# Read in data
mn = read_csv(file = "~/Documents/github/epsy-8251/data/mn-schools.csv")
head(mn)
```

```
# A tibble: 6 x 6
     id name                             gradRate public   sat tuition
  <int> <chr>                               <dbl>  <int> <int>   <int>
1     1 Augsburg College                     65.2      0  1030   39294
2     3 Bethany Lutheran College             52.6      0  1065   30480
3     4 Bethel University, Saint Paul, MN    73.3      0  1145   39400
4     5 Carleton College                     92.6      0  1400   54265
5     6 College of Saint Benedict            81.1      0  1185   43198
6     7 Concordia College at Moorhead        69.4      0  1145   36590
```

Examine Relationship between Graduation Rate and SAT Scores

As always, we begin the analysis by graphing the data.

```
ggplot(data = mn, aes(x = sat, y = gradRate)) +
    geom_point() +
    geom_smooth(se = FALSE) +
    theme_bw() +
  xlab("Estimated median SAT score") +
  ylab("Six-year graduation rate")
```

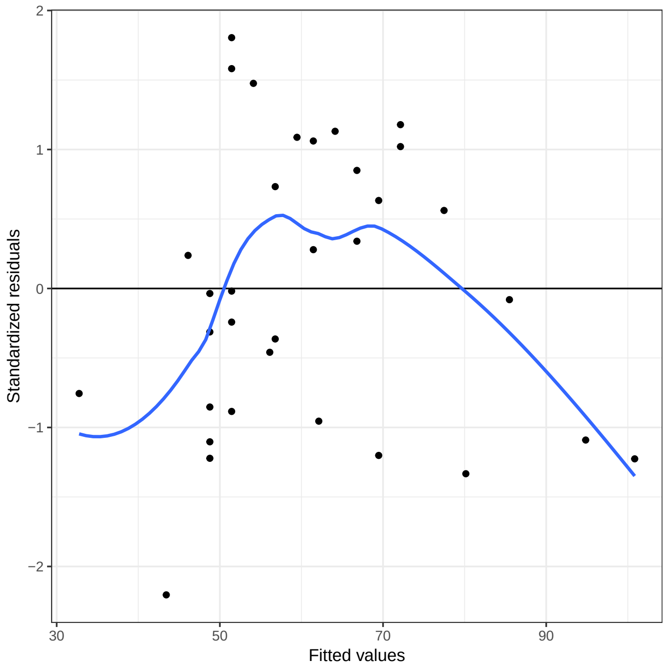


The loess smoother suggests that the relationship between SAT scores and graduation rate is non-linear. Nonlinearity implies that a the effect of SAT on graduation rates is not constant across the range of X; for colleges with lower values of SAT (say SAT < 1100) the effect of SAT has a rather high, positive effect (steep slope), while for colleges with higher values of SAT (≥ 1100) the effect of SAT is positive and moderate (the slope is less steep). Another way of saying this is that for schools with lower SAT scores, a one-unit difference in SAT is associated with a larger change in graduation rates than the same one-unit change for schools with higher SAT values.

Residual Plot: Another Way to Spot Nonlinearity

Sometimes, the nonlinear relationship is difficult to detect from the scatterplot of Y versus X. Often it helps to fit the linear model and then examine the assumption of linearity in the residuals. It is sometimes easier to detect nonlinearity in the plot of the residuals versus the fitted values.

```
# Fit linear model
lm.1 = lm(gradRate ~ 1 + sat, data = mn)

# Obtain residuals
out = augment(lm.1)

# Examine residuals for linearity
ggplot(data = out, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth(se = FALSE) +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")
```



This plot suggests that the assumption of linearity may be violated. There is systematic over-estimation for low fitted values, systematic under-estimation for moderate fitted values, and systematic over-estimation for high fitted values.

Polynomial Models

One way of modeling non-linearity is by including polynomial effects. In regression, a polynomial effects are predictors that have a power greater than one. For example, x^2 (quadratic term), or x^3 (cubic term). Note that

$$x^2 = x \times x.$$

So the quadratic term, x^2 is a product of x times itself. Recall that products are how we express interactions. Thus the quadratic term of x^2 is really the interaction of x with itself. To model this, we simply (1) create the product term, and (2) include the product term and all constituent main-effects in the regression model.

```
# Create quadratic term in the data
mn = mn %>%
  mutate(
    sat_quadratic = sat * sat
  )

head(mn)
```

```
# A tibble: 6 x 7
     id name                   gradRate public   sat tuition sat_quadratic
  <int> <chr>                     <dbl>  <int> <int>   <int>         <int>
1     1 Augsburg College           65.2      0  1030   39294       1060900
2     3 Bethany Lutheran Coll~     52.6      0  1065   30480       1134225
3     4 Bethel University, Sa~     73.3      0  1145   39400       1311025
4     5 Carleton College           92.6      0  1400   54265       1960000
5     6 College of Saint Bene~     81.1      0  1185   43198       1404225
6     7 Concordia College at ~     69.4      0  1145   36590       1311025
```

```
# Fit model
lm.2 = lm(gradRate ~ 1 + sat + sat_quadratic, data = mn)

# Model-level output
glance(lm.2)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
*     <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.835         0.824  7.02      76.0 1.81e-12     3  -110.  227.  233.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
# Coefficient-level output
tidy(lm.2)
```

```
# A tibble: 3 x 5
  term              estimate  std.error statistic  p.value
  <chr>                <dbl>      <dbl>     <dbl>    <dbl>
1 (Intercept)   -366.       98.6            -3.71 0.000831
2 sat              0.627     0.173           3.63 0.00104
3 sat_quadratic   -0.000215  0.0000751      -2.86 0.00756
```

Since this is an interaction model, we start by examining the interaction term; the quadratic coefficient. This term is statistically reliable ($p = .008$), suggesting that the quadratic term explains variation above and beyond the linear term. This suggests that we should keep the quadratic term in the model.

Interpretation of a Significant Polynomial Term

How do we interpret the quadratic term? First, we will write out the fitted model.

$$\widehat{\text{Graduation Rate}} = -366.3 + 0.63(\text{SAT}) - 0.0002(\text{SAT}^2)$$

From algebra, you may remember that the coefficient in front of the quadratic term (-0.0002) informs us of whether the quadratic is an upward-facing U-shape, or a downward-facing U-shape. Since our term is negative, the U-shape is downward-facing. It also indicates whether the U-shape is skinny or wide. The intercept and linear terms help us locate the U-shape in the coordinate plane (moving it right, left, up, or down from the origin). You could work these out algebraically, but typically, we will just plot the predicted values and interpret from the plot.

Refit the model using the I() function

Before we create the plot, we use a different method of fitting polynomial terms in a regression. Rather than create a new variable in the data set, we insert the polynomial directly into the model using the `I()` function.

```
# Fit model using I() function
lm.2 = lm(gradRate ~ 1 + sat + I(sat ^ 2), data = mn)

# Model-level output
glance(lm.2)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
*     <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.835         0.824  7.02      76.0 1.81e-12     3  -110.  227.  233.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
# Coefficient-level output
tidy(lm.2)
```

```
# A tibble: 3 x 5
  term           estimate  std.error statistic  p.value
  <chr>             <dbl>      <dbl>     <dbl>    <dbl>
1 (Intercept) -366.       98.6           -3.71 0.000831
2 sat            0.627     0.173          3.63 0.00104
3 I(sat^2)      -0.000215  0.0000751     -2.86 0.00756
```

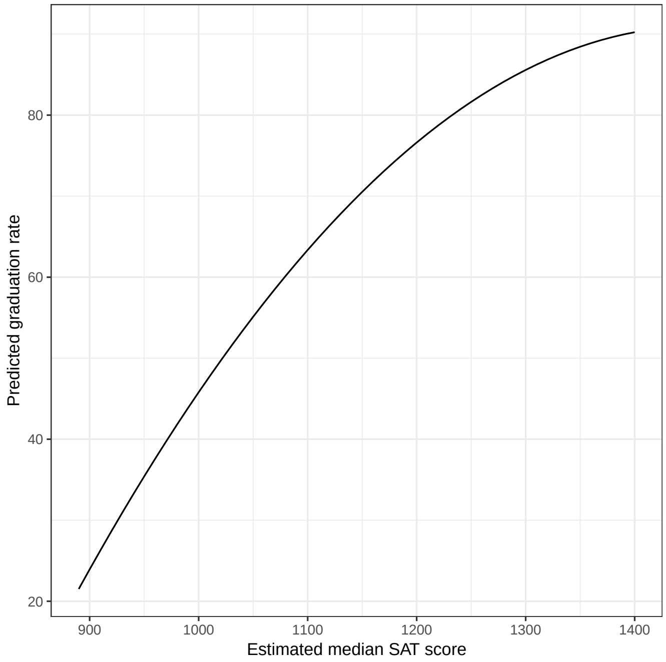
The `I()` function forces R to create a separate term for the quadratic (which is essentially what we do by adding the product term to the model). Without it, R will do the computation `sat + sat^2` and use those values as a single variable…not what we want. The other advantage for plotting is that we have only used a single predictor, `sat` in specifying the model. Thus we only need to include `sat` in our plot data rather than both `sat` and `sat_quadratic`. (Note: You cannot use the colon notation (`:`) to fit a polynomial term in the model.)

```
# Set up data and predict
plotData = crossing(
    sat = seq(from = 890, to = 1400, by = 10)
    ) %>%
  mutate(
    yhat = predict(lm.2, newdata = .)
  )


# Examine data
head(plotData)
```

```
# A tibble: 6 x 2
    sat  yhat
  <dbl> <dbl>
1   890  21.5
2   900  23.9
3   910  26.3
```

```
4   920  28.6
5   930  30.9
6   940  33.2
```

```r
# Plot
ggplot(data = plotData, aes(x = sat, y = yhat)) +
    geom_line() +
  theme_bw() +
  xlab("Estimated median SAT score") +
  ylab("Predicted graduation rate")
```

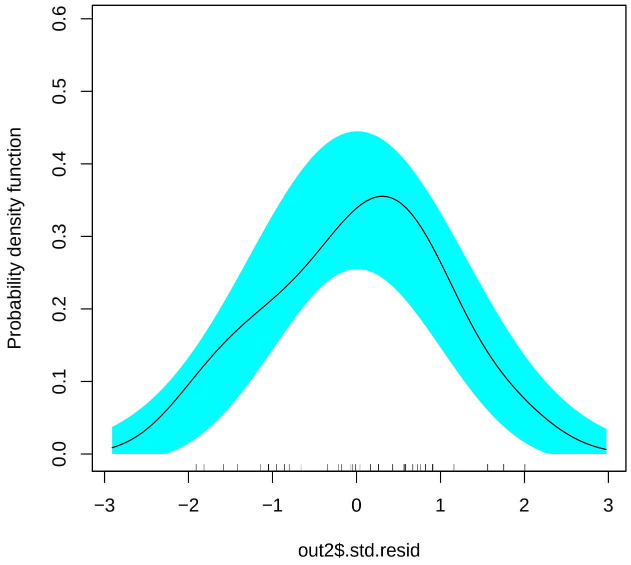


The quadratic relationship is expressed in the predicted values. Aside from plotting them versus SAT scores, there is nothing further we need to do to get the quadratic relationship to appear. The plot, more importantly, helps us interpret the relationship between SAT scores and graduation rates. The effect of SAT on graduation rate depends on SAT score (definition of an interaction). For schools with low SAT scores, the effect of SAT score on graduation rate is positive and fairly high. For schools with high SAT scores, the effect of SAT score on graduation rate remains positive, but it has a smaller effect on graduation rates.

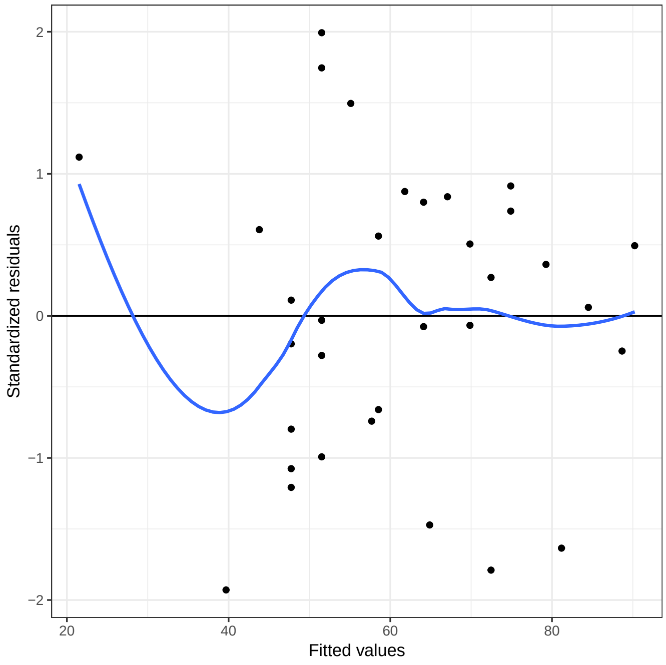
Re-Examining the Residuals for the Quadratic Model

Since we fitted a different model, we should examine the residuals to see whether the assumptions for the model seem satisfied.

```r
out2 = augment(lm.2)

# Check normality
sm.density(out2$.std.resid, model = "normal")
```



```r
# Check other assumptions
ggplot(data = out2, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth(se = FALSE) +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")
```



Based on the plots, the quadratic model seems to meet the assumptions for regression. At the very least, it clearly does so better than the linear model.

Adding Covariates

We can also include covariates in a polynomial model (to control for other predictors), the same way we do in a linear model, by including them as additive terms in the `lm()` model. Below we include the `public` dummy-coded predictor to control for the effects of sector.

```r
# Fit model
lm.3 = lm(gradRate ~ 1 + sat + I(sat ^ 2) + public, data = mn)

# Model-level output
glance(lm.3)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
*     <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.897         0.886  5.64      84.1 2.05e-14     4  -102.  214.  221.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```r
# Coefficient-level output
tidy(lm.3)
```

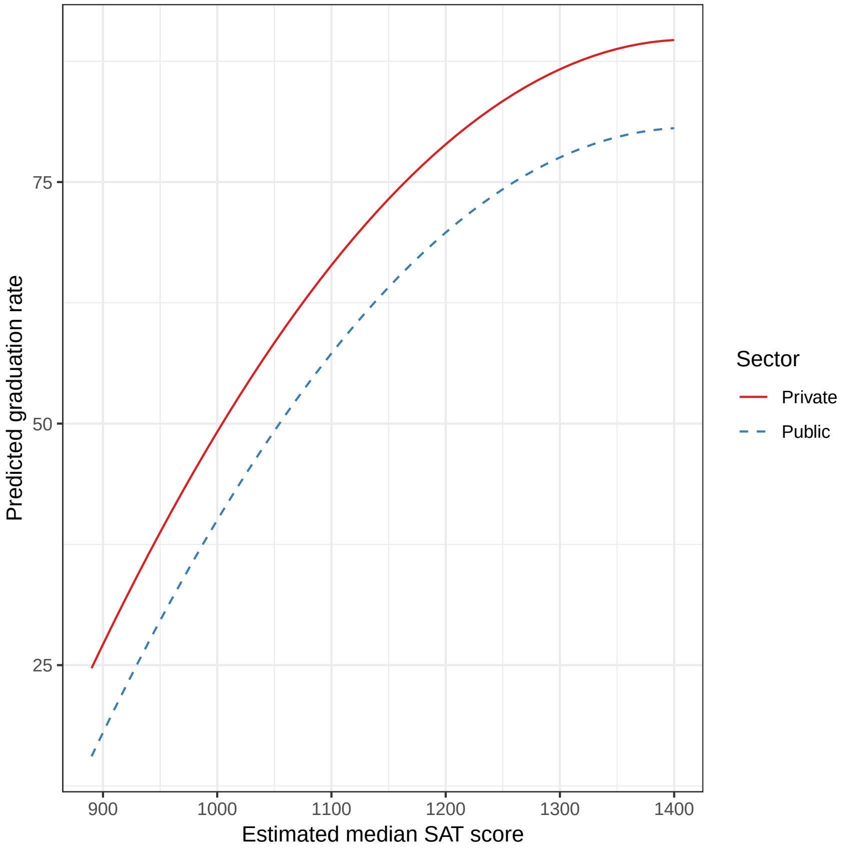
```
# A tibble: 4 x 5
  term         estimate  std.error statistic   p.value
  <chr>           <dbl>      <dbl>     <dbl>     <dbl>
1 (Intercept) -384.      79.4          -4.84 0.0000398
2 sat            0.670    0.139         4.81 0.0000425
3 I(sat^2)      -0.000237 0.0000606    -3.91 0.000507
4 public        -9.12     2.19         -4.17 0.000251
```

Here there is still a quadratic effect of SAT on graduation rates, even after controlling for differences in sector ($p = 0.0005$). This means that after controlling for sector differences, the effect of SAT on graduation rate depends on SAT (the effect of SAT is differen for different levels of SAT). There are also statistically reliable differences in sector after controlling for the linear and quadratic effects of SAT ($p = 0.003$). The sector effect can be interpreted as, after controlling for the effect of SAT, public schools have a graduation rate that is 9.1% lower than private schools, on average. Plot the fitted model to aid interpretation.

```r
# Set up data; get predicted values, coerce public into a factor for better plotting
plotData = crossing(
    sat = seq(from = 890, to = 1400, by = 10),
    public = c(0, 1)
    ) %>%
  mutate(
    yhat = predict(lm.3, newdata = .),
    public = factor(public, levels = c(0, 1), labels = c("Private", "Public"))
  )


# Examine data
head(plotData)
```

```
# A tibble: 6 x 3
    sat public   yhat
  <dbl> <fct>   <dbl>
1   890 Private  24.7
2   890 Public   15.6
3   900 Private  27.1
4   900 Public   18.0
5   910 Private  29.6
6   910 Public   20.4
```

```
# Plot
ggplot(data = plotData, aes(x = sat, y = yhat, group = public, color = public, linetype = public)) +
    geom_line() +
  theme_bw() +
  xlab("Estimated median SAT score") +
  ylab("Predicted graduation rate") +
  scale_color_brewer(name = "Sector", palette = "Set1") +
  scale_linetype_manual(name = "Sector", values = c(1, 2))
```



The plot shows the quadratic effect of SAT scores on graduation rate; the effect of SAT on graduation rates is positive, but this effect declines for increasingly higher SAT scores, after controlling for sector differences. Private schools have higher graduation rates, on average, than public schools for all levels of SAT score.