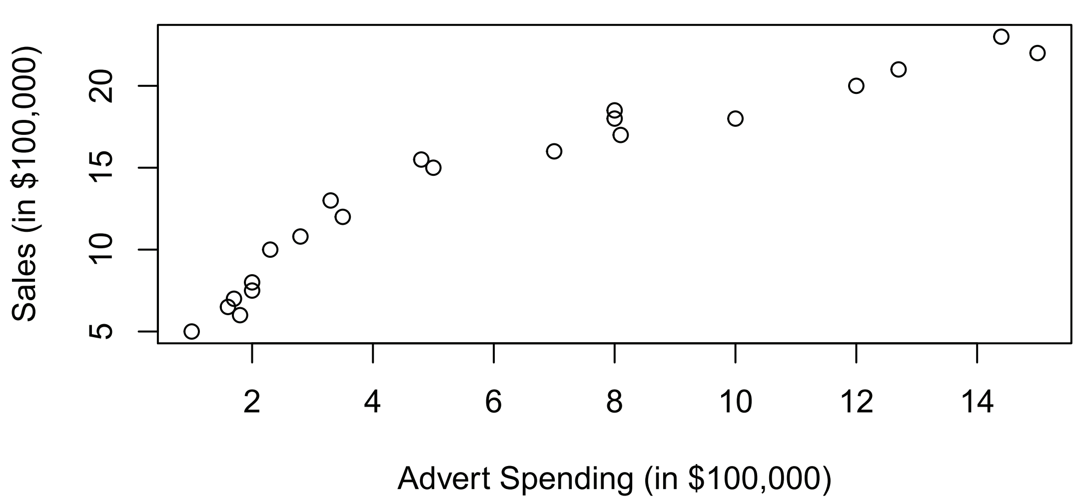
Polynomial Regression

It is well known that the sales response to advertising usually follows a curve reflecting the diminishing returns to advertising expenditure. As a company increases its advertising expenditure, sales increase, but the rate of increase drops continually after a certain point. If we consider company sales profits as a function of advertising expenditure, we find that the response function can be very well approximated by a second-order (quadratic) model. For a particular company, the data on monthly sales y and monthly advertising expenditure x, both in hundred thousand dollars, can be found in the data below.

```
sales <- c(5.0,6.0,6.5,7.0,7.5,8.0,10.0,10.8,12.0,13.0,15.5,15.0,16.0,17.0,
           18.0,18.0,18.5,21.0,20.0,22.0,23.0)
advert <- c(1.0,1.8,1.6,1.7,2.0,2.0,2.3,2.8,3.5,3.3,4.8,5.0,7.0,8.1,8.0,10.0,
            8.0,12.7,12.0,15.0,14.4)
marketing <- data.frame(sales,advert)
head(marketing)
```

```
##   sales advert
## 1   5.0    1.0
## 2   6.0    1.8
## 3   6.5    1.6
## 4   7.0    1.7
## 5   7.5    2.0
## 6   8.0    2.0
```

```
plot(marketing$advert,marketing$sales,,
     xlab = "Advert Spending (in $100,000)", ylab = "Sales (in $100,000)")
```



We want to fit the model,

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma^2)$ for $i = 1, 2, \cdots 21$. Thus, our $\boldsymbol{X}$ matrix is,

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ \cdots & \cdots & \cdots \\ 1 & x_{21} & x_{21}^2 \end{bmatrix}$$

We can then proceed to fit the model as we have in the past for multiple linear regression.

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y} \qquad E[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta} \qquad Var[\hat{\boldsymbol{\beta}}] = \boldsymbol{\sigma}^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}$$

```r
mark_mod <- lm(sales ~ advert, data = marketing)
summary(mark_mod)
```

```
##
## Call:
## lm(formula = sales ~ advert, data = marketing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7845 -1.4762 -0.5103  1.2361  3.1869
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5927     0.7031   9.377 1.47e-08 ***
## advert        1.1918     0.0937  12.718 9.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.907 on 19 degrees of freedom
## Multiple R-squared:  0.8949, Adjusted R-squared:  0.8894
## F-statistic: 161.8 on 1 and 19 DF,  p-value: 9.646e-11
```

```r
mark_mod_poly2 <- lm(sales ~ advert + I(advert^2), data = marketing)
summary(mark_mod_poly2)
```

```
##
## Call:
## lm(formula = sales ~ advert + I(advert^2), data = marketing)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.9175 -0.8333 -0.1948  0.9292  2.1385
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.51505    0.73847   4.760 0.000157 ***
## advert       2.51478    0.25796   9.749 1.32e-08 ***
## I(advert^2) -0.08745    0.01658  -5.275 5.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.228 on 18 degrees of freedom
## Multiple R-squared:  0.9587, Adjusted R-squared:  0.9541
## F-statistic:   209 on 2 and 18 DF,  p-value: 3.486e-13
```

Here we see that with the first order term in the model, the quadratic term is also significant.

```r
X <- cbind( rep(1,21), marketing$advert, marketing$advert^2 )
t(X) %*% X
```

```
##         [,1]     [,2]      [,3]
## [1,]   21.00   127.00   1182.26
## [2,]  127.00  1182.26  13416.17
## [3,] 1182.26 13416.17 166843.66
```

```r
solve(t(X) %*% X) %*% t(X) %*% marketing$sales
```

```
##             [,1]
## [1,]  3.51504670
## [2,]  2.51478201
## [3,] -0.08745394
```

Here we verify the parameter estimates were found as we would expect.

We could also add higher order terms, our $\boldsymbol{X}$ matrix simply becomes larger again.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$$

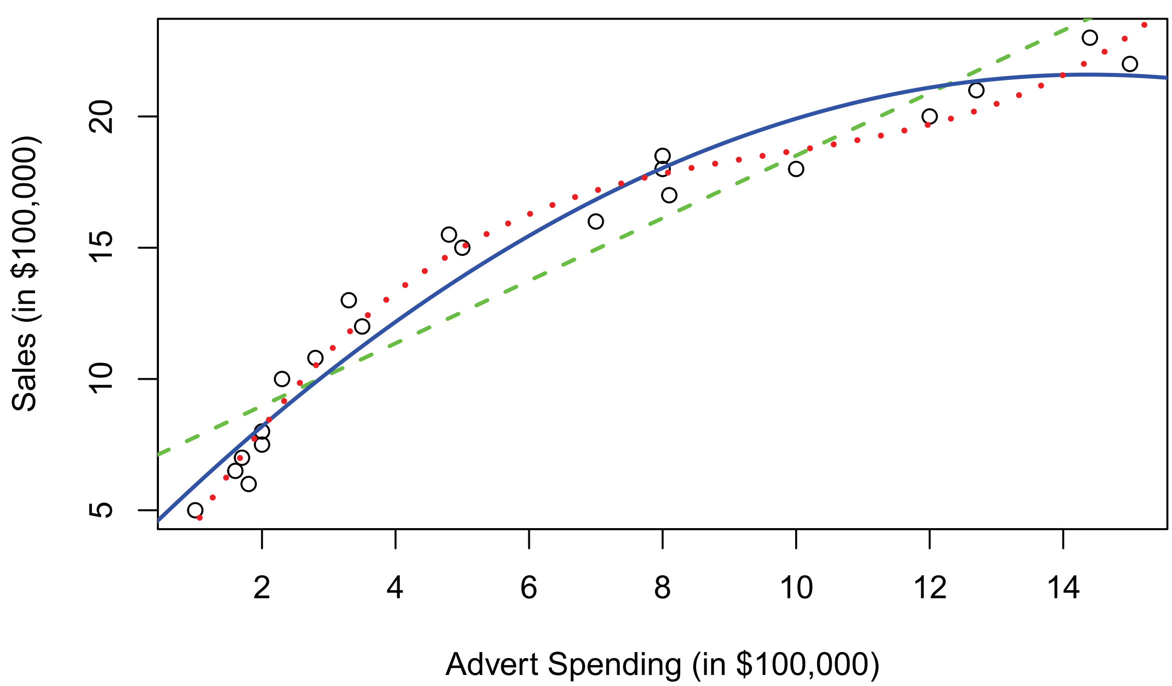
```r
mark_mod_poly3 <- lm(sales ~ advert + I(advert^2) + I(advert^3), data = marketing)
summary(mark_mod_poly3)
```

```
##
## Call:
## lm(formula = sales ~ advert + I(advert^2) + I(advert^3), data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44322 -0.61310 -0.01527  0.68131  1.22517
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.055888   0.909533  -0.061 0.951720
```

```
## advert       4.959310   0.548371   9.044 6.61e-08 ***
## I(advert^2) -0.469669   0.082033  -5.725 2.48e-05 ***
## I(advert^3)  0.016131   0.003429   4.704 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8329 on 17 degrees of freedom
## Multiple R-squared:  0.9821, Adjusted R-squared:  0.9789
## F-statistic: 310.2 on 3 and 17 DF,  p-value: 4.892e-15
```

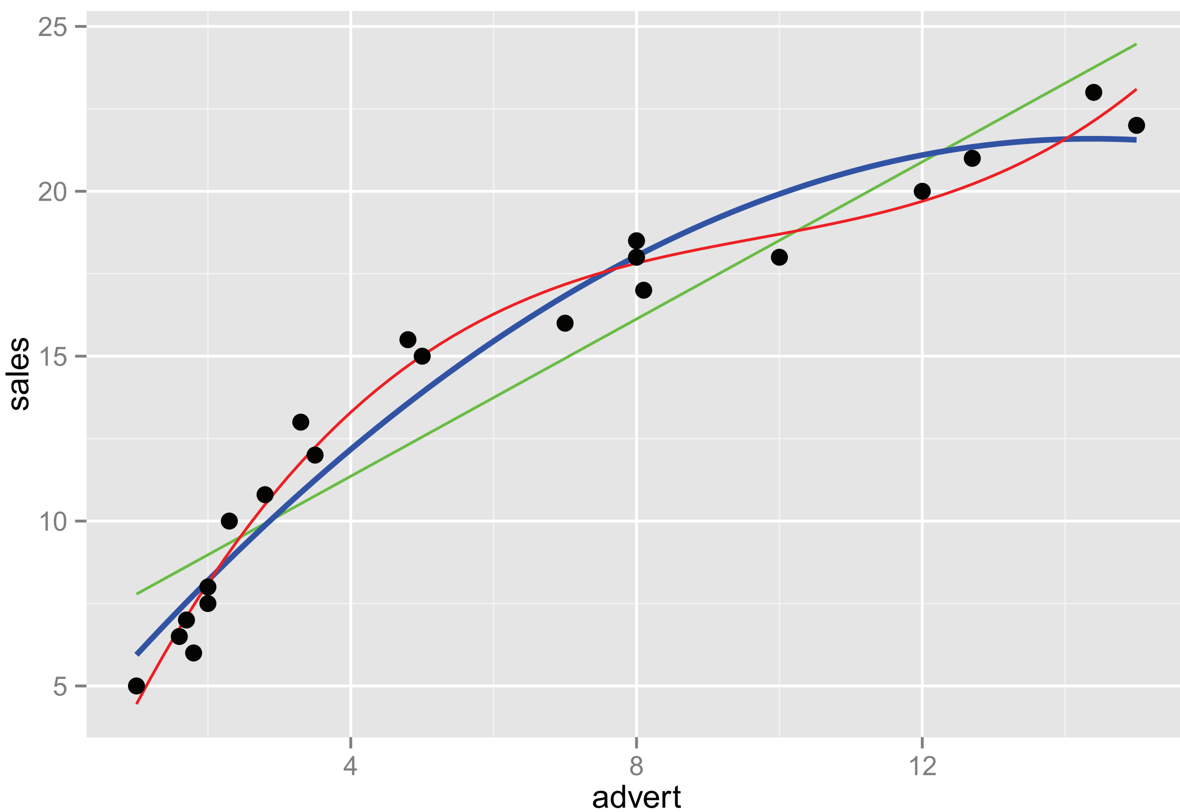
Now we see that with the first and second order terms in the model, the third order term is also significant. But does this make sense practically? The following plot should gives hints as to why it doesn't.

```
plot(marketing$advert,marketing$sales,
     xlab = "Advert Spending (in $100,000)", ylab = "Sales (in $100,000)")
abline(mark_mod, lty = 2, col = "green", lwd = 2)
xplot <- seq(0, 16, by = 0.01)
lines(xplot, predict(mark_mod_poly2, newdata = data.frame(advert = xplot)),
      col="blue", lwd = 2)
lines(xplot, predict(mark_mod_poly3, newdata = data.frame(advert = xplot)),
      col="red", lty = 3, lwd = 3)
```



The previous plot was made using base graphics in `R`. The next plot was made using the package `ggplot2`, a popular plotting method in `R`.

```
library(ggplot2)
ggplot(data = marketing, aes(x = advert, y = sales)) +
  stat_smooth(method = "lm", se=FALSE, color="green", formula = y ~ x) +
  stat_smooth(method = "lm", se=FALSE, color="blue", size = 1, formula = y ~ x + I(x^2)) +
  stat_smooth(method = "lm", se=FALSE, color="red", formula = y ~ x + I(x^2)+ I(x^3)) +
  geom_point(colour = "black", size = 3)
```



Note we can fit a polynomial of an arbitrary order,

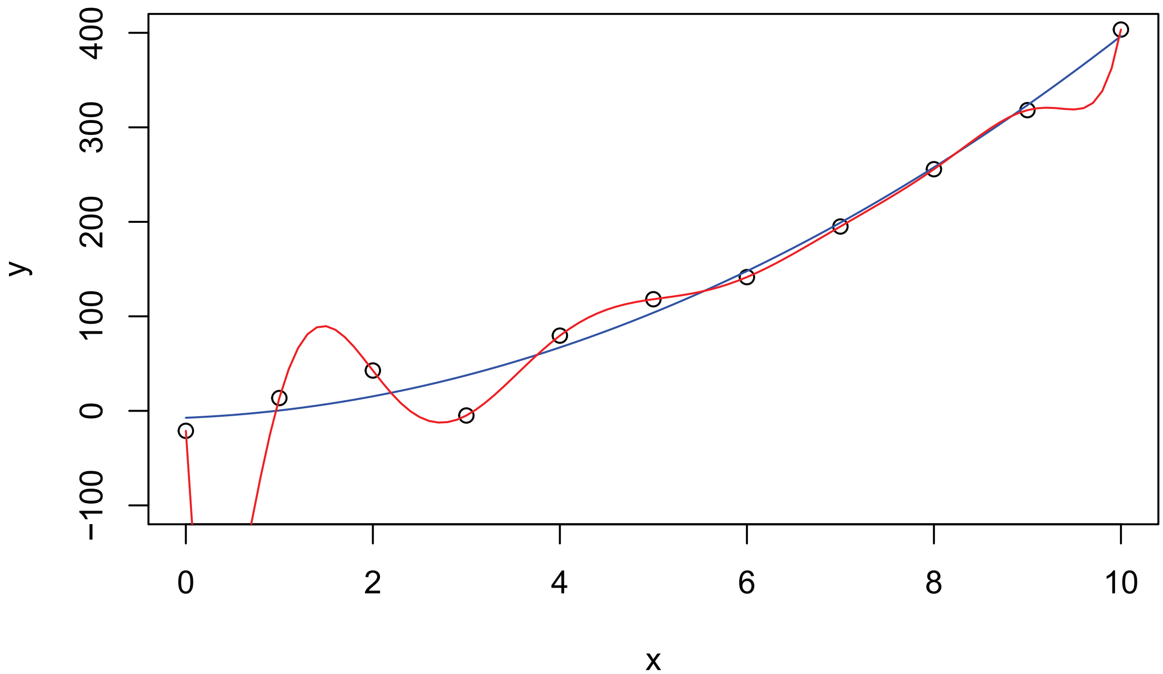
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_{p-1} x_i^{p-1} + \epsilon_i$$

However, we should be careful about overfitting, since with a polynomial of degree one less than the number of observations, it is sometimes possible to fit a model perfectly.

```
set.seed(1234)
x <- seq(0,10)
y <- 3 + x + 4*x^2 + rnorm(11,0,20)
plot(x,y,ylim=c(-100,400))
fit <- lm(y ~ x + I(x^2))
#summary(fit)
fit_perf <- lm(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6)
               + I(x^7) + I(x^8) + I(x^9) + I(x^10))
summary(fit_perf)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) +
##     I(x^7) + I(x^8) + I(x^9) + I(x^10))
##
## Residuals:
## ALL 11 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.114e+01         NA      NA       NA
## x           -1.918e+03         NA      NA       NA
## I(x^2)       4.969e+03         NA      NA       NA
## I(x^3)      -4.932e+03         NA      NA       NA
## I(x^4)       2.581e+03         NA      NA       NA
## I(x^5)      -8.035e+02         NA      NA       NA
## I(x^6)       1.570e+02         NA      NA       NA
## I(x^7)      -1.947e+01         NA      NA       NA
## I(x^8)       1.490e+00         NA      NA       NA
## I(x^9)      -6.424e-02         NA      NA       NA
## I(x^10)      1.195e-03         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 10 and 0 DF,  p-value: NA
```

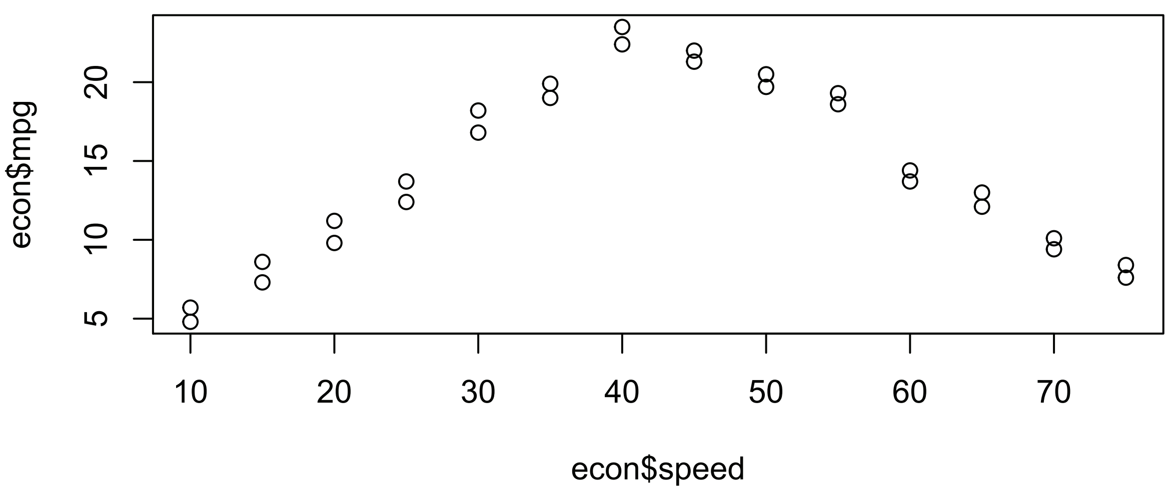
```
xplot <- seq(0, 10, by = 0.1)
lines(xplot, predict(fit, newdata = data.frame(x = xplot)),
      col="blue", lwd = 1, lty = 1)
lines(xplot, predict(fit_perf, newdata = data.frame(x = xplot)),
      col="red", lwd = 1, lty = 1)
```



400
300
200
100
0
−100
y
0
2
4
6
8
10
x

We now wish to develop a model to predict miles per gallon based on highway speed for a particular brand of SUV. An experiment is designed in which a test car is driven at speeds ranging from 10 miles per hour to 75 miles per hour. We will fit a polynomial model and use it to predict the average mileage obtained when the car is driven at 55 miles per hour.

```
mpg <- c(4.8,5.7,8.6,7.3,9.8,11.2,13.7,12.4,18.2,16.8,19.9,19.0,22.4,23.5,21.3,
         22.0,20.5,19.7,18.6,19.3,14.4,13.7,12.1,13.0,10.1,9.4,8.4,7.6)
speed <- c(10,10,15,15,20,20,25,25,30,30,35,35,40,40,45,45,50,50,55,55,60,
           60,65,65,70,70,75,75)
econ <- data.frame(speed, mpg)
plot(econ$speed, econ$mpg)
```



In this example, we will be frequently looking at the fitted vs residuals plot, so we will write a function to make our life easier.

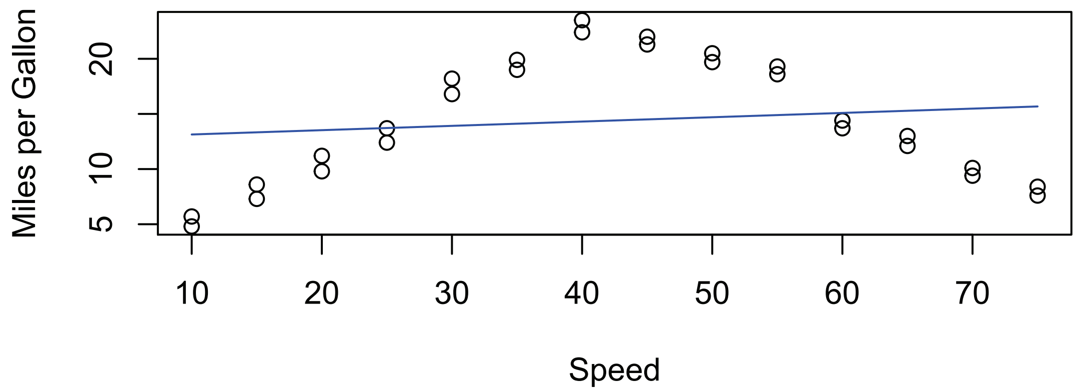
```
plot_fit_res <- function(model){
  plot(fitted(model),resid(model), xlab = "Fitted", ylab = "Residuals")
  abline(h = 0)
}
```

We will do the same for added a curve to our data points.

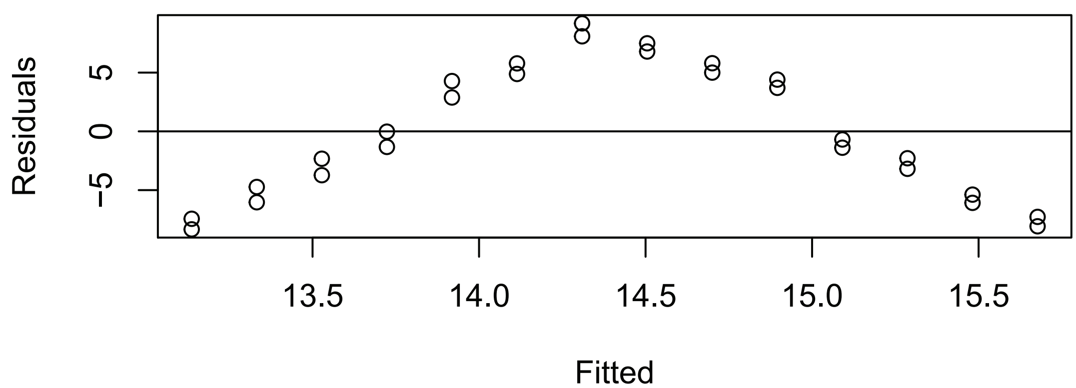
```
plot_econ_curve <- function(model){
  plot(econ$speed, econ$mpg, xlab = "Speed", ylab = "Miles per Gallon")
  xplot <- seq(10, 75, by = 0.1)
  lines(xplot, predict(model, newdata = data.frame(speed = xplot)),
        col="blue", lwd = 1, lty = 1)
}
```

So now we first fit a simple linear regression to this data.

```r
fit1 <- lm(mpg ~ speed, data = econ)
plot_econ_curve(fit1)
```




```r
plot_fit_res(fit1)
```



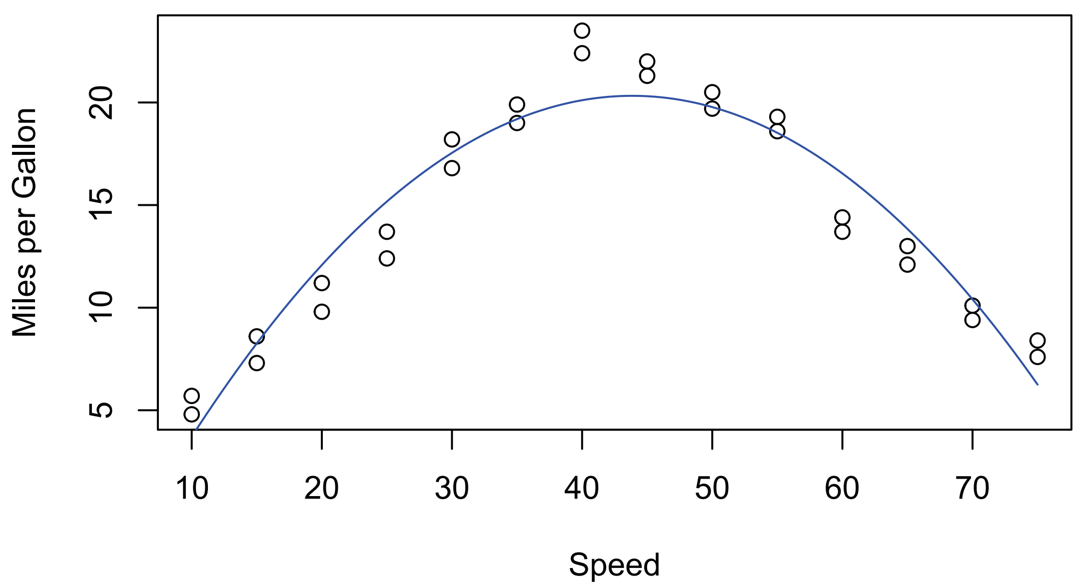
Pretty clearly we can do better. We will now add more polynomial terms until we fit a suitable fit.

```r
fit2 <- lm(mpg ~ speed + I(speed^2), data = econ)
summary(fit2)
```

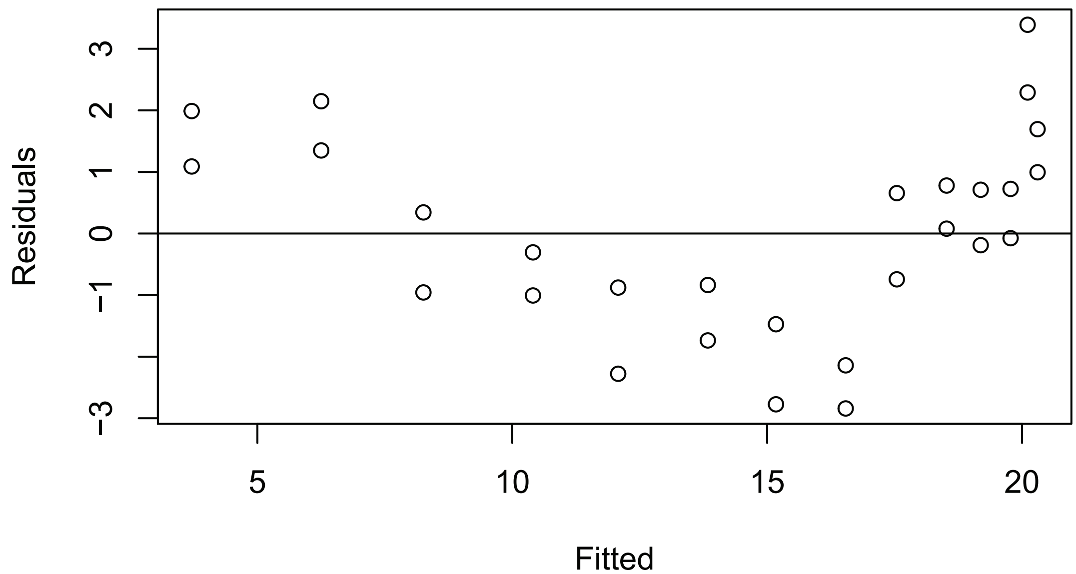
```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2), data = econ)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8411 -0.9694  0.0017  1.0181  3.3900
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.5555495  1.4241091  -5.305 1.69e-05 ***
## speed        1.2716937  0.0757321  16.792 3.99e-15 ***
## I(speed^2)  -0.0145014  0.0008719 -16.633 4.97e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.663 on 25 degrees of freedom
## Multiple R-squared:  0.9188, Adjusted R-squared:  0.9123
## F-statistic: 141.5 on 2 and 25 DF,  p-value: 2.338e-14
```

```r
plot_econ_curve(fit2)
```




```r
plot_fit_res(fit2)
```

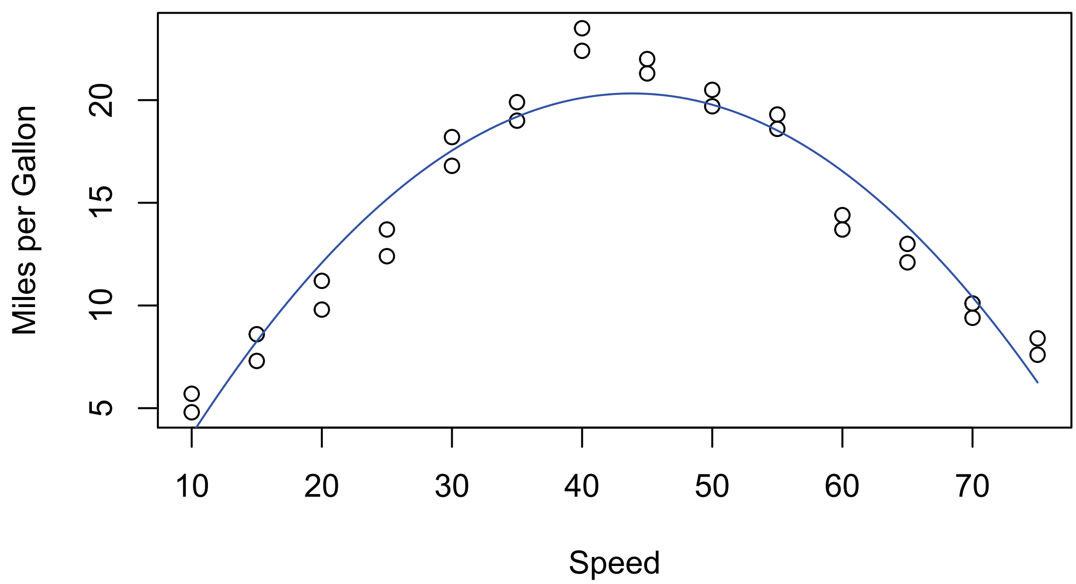


While this model clearly fits much better, and the second order term is significant, we still see a pattern in the fitted versus residuals plot which suggests higher order terms will help. Also, we would expect the curve to flatten as speed increases or decreases, not go sharply downward as we see here.

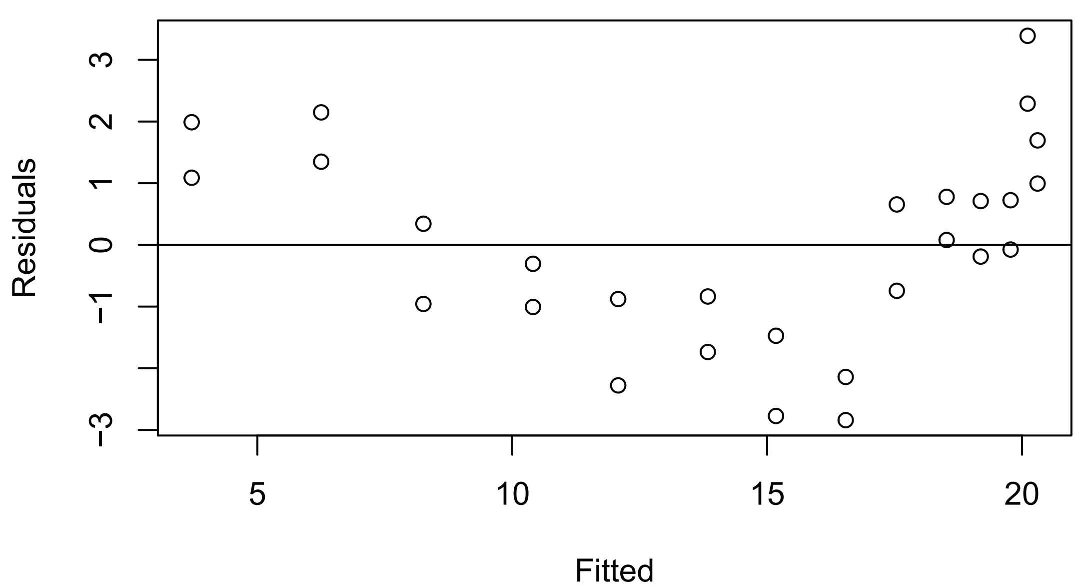
```
fit3 <- lm(mpg ~ speed + I(speed^2) + I(speed^3), data = econ)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2), data = econ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8411 -0.9694  0.0017  1.0181  3.3900
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.5555495  1.4241091  -5.305 1.69e-05 ***
## speed        1.2716937  0.0757321  16.792 3.99e-15 ***
## I(speed^2)  -0.0145014  0.0008719 -16.633 4.97e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.663 on 25 degrees of freedom
## Multiple R-squared:  0.9188, Adjusted R-squared:  0.9123
## F-statistic: 141.5 on 2 and 25 DF,  p-value: 2.338e-14
```

```r
plot_econ_curve(fit2)
```




```r
plot_fit_res(fit2)
```

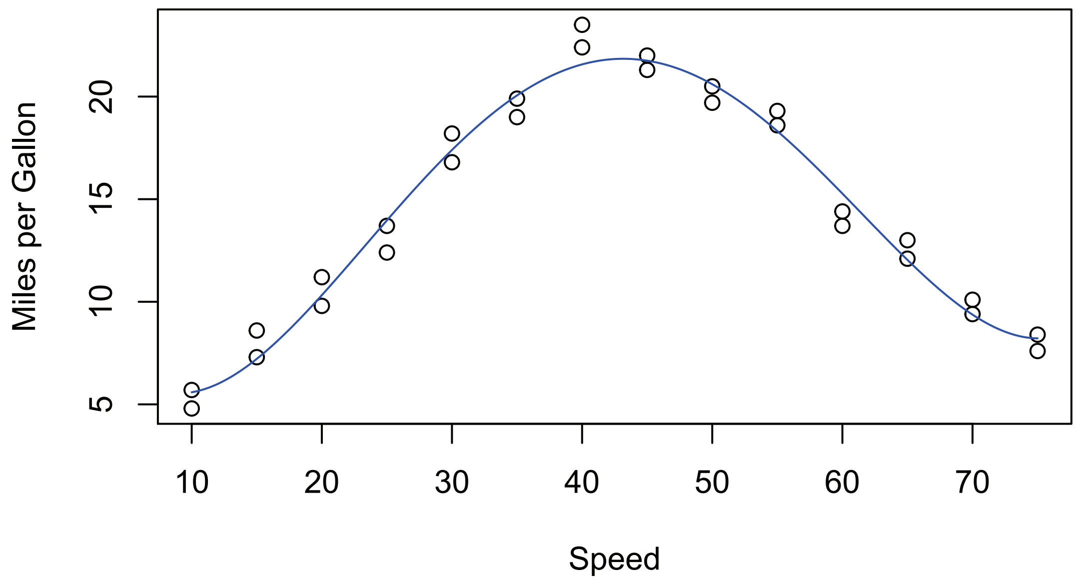


Adding the third order term doesn't seem to help. This makes sense, since what we would like is for the curve to flatten at the extremes. For this we will need an even polynomial term.

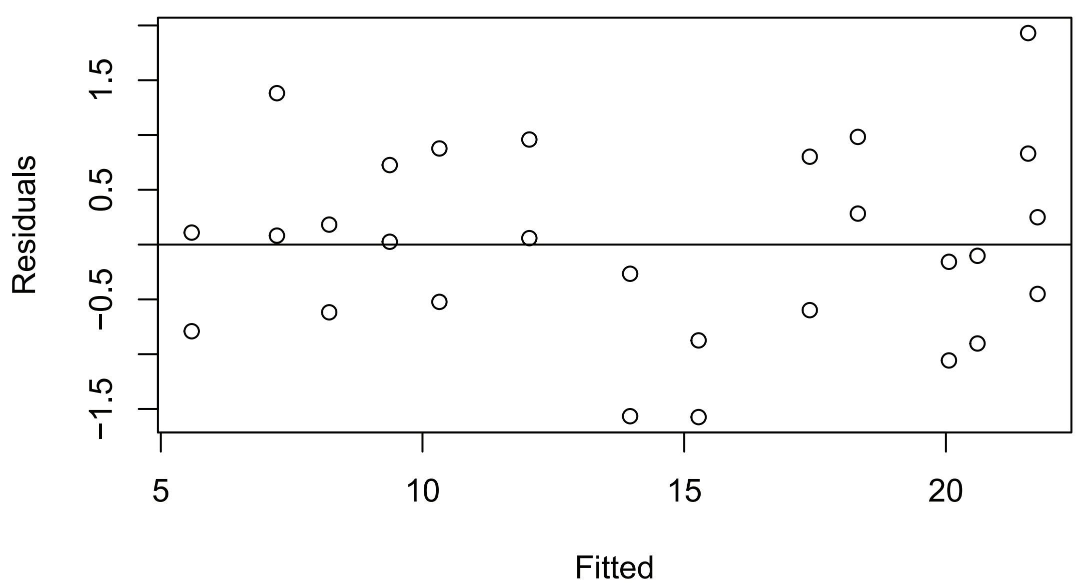
```r
fit4 <- lm(mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4), data = econ)
summary(fit4)
```

```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4),
##     data = econ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.57410 -0.60308  0.04236  0.74481  1.93038
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.146e+01  2.965e+00   3.866 0.000785 ***
## speed       -1.468e+00  3.913e-01  -3.751 0.001042 **
## I(speed^2)   1.081e-01  1.673e-02   6.463 1.35e-06 ***
## I(speed^3)  -2.130e-03  2.844e-04  -7.488 1.31e-07 ***
## I(speed^4)   1.255e-05  1.665e-06   7.539 1.17e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9307 on 23 degrees of freedom
## Multiple R-squared:  0.9766, Adjusted R-squared:  0.9726
## F-statistic: 240.2 on 4 and 23 DF,  p-value: < 2.2e-16
```

```r
plot_econ_curve(fit4)
```




```
plot_fit_res(fit4)
```

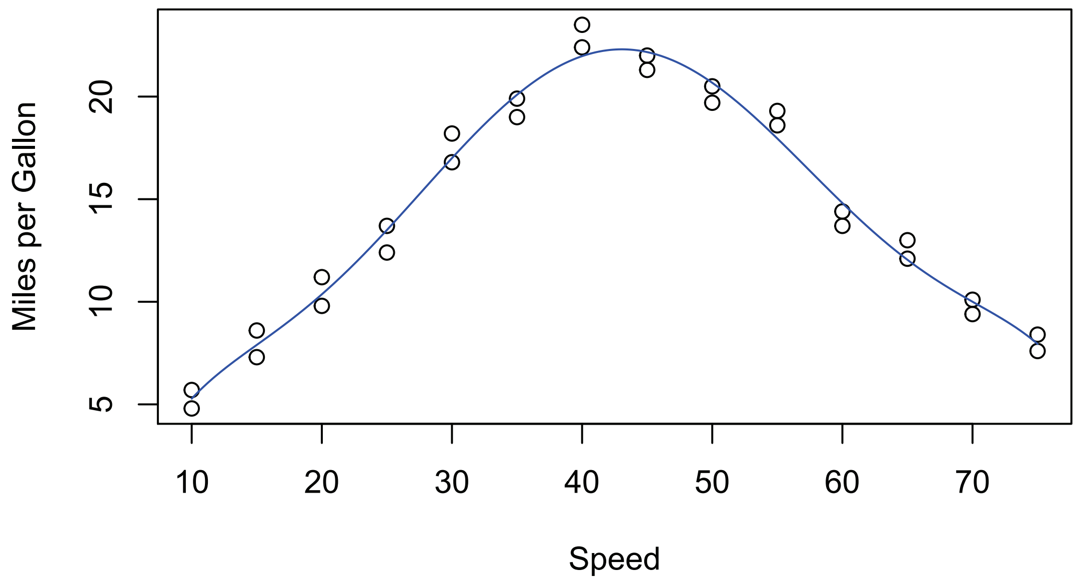


Now we are making progress. The fourth order term is significant with the other terms in the model. Also we are starting to see what we expected for low and high speeds. However, there still seems to be a bit of a pattern in the residuals, so we will again try more higher order terms. (We will add the fifth and sixth together, since adding the fifth will be similar to adding the third.)

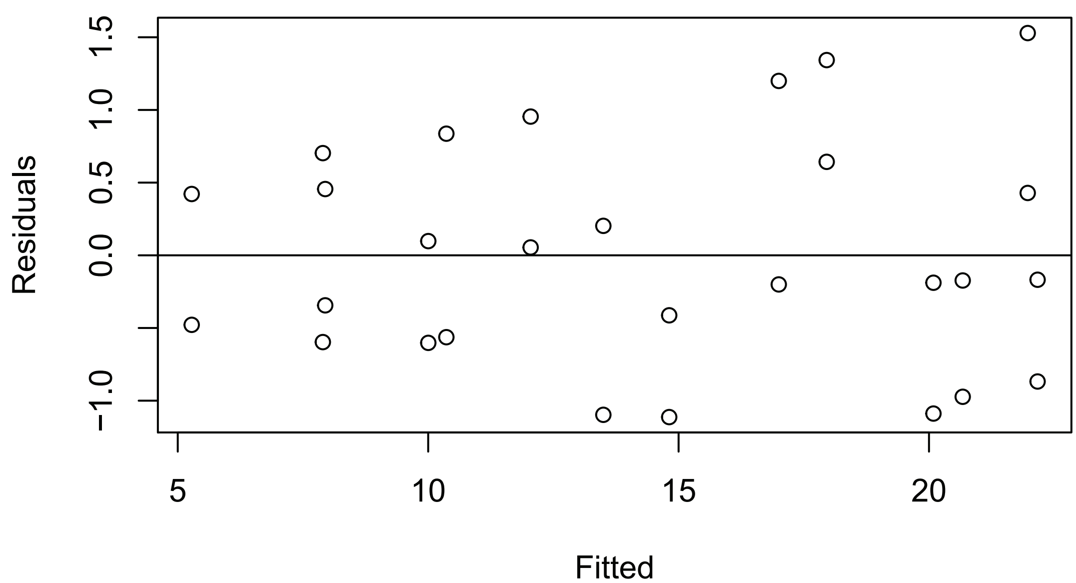
```
fit6 <- lm(mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) + I(speed^5)
           + I(speed^6), data = econ)
summary(fit6)
```

```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) +
##     I(speed^5) + I(speed^6), data = econ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1129 -0.5717 -0.1707  0.5026  1.5288
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.421e+01  1.204e+01  -1.180   0.2514
## speed        4.203e+00  2.553e+00   1.646   0.1146
## I(speed^2)  -3.521e-01  2.012e-01  -1.750   0.0947 .
## I(speed^3)   1.579e-02  7.691e-03   2.053   0.0527 .
## I(speed^4)  -3.473e-04  1.529e-04  -2.271   0.0338 *
## I(speed^5)   3.585e-06  1.518e-06   2.362   0.0279 *
## I(speed^6)  -1.402e-08  5.941e-09  -2.360   0.0280 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8657 on 21 degrees of freedom
## Multiple R-squared:  0.9815, Adjusted R-squared:  0.9762
## F-statistic:   186 on 6 and 21 DF,  p-value: < 2.2e-16
```

```
plot_econ_curve(fit6)
```




```r
plot_fit_res(fit6)
```



Again the sixth order term is significant with the other terms in the model and here we see less pattern in the residuals plot. Let's now test for which of the previous two models we prefer. Namely we will test.

$$H_0 : \beta_5 = \beta_6 = 0.$$

```r
anova(fit4,fit6)
```

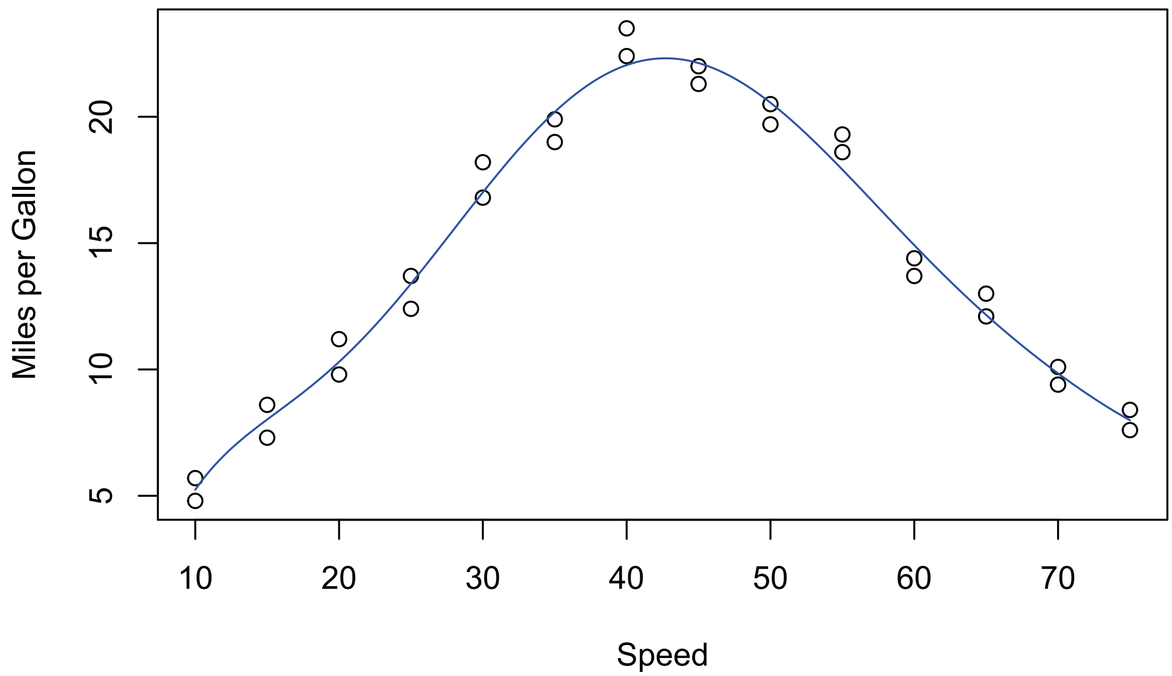
```
## Analysis of Variance Table
##
## Model 1: mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4)
## Model 2: mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) + I(speed^5) +
##     I(speed^6)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     23 19.922
## 2     21 15.739  2    4.1828 2.7905 0.0842 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, this test does not reject the null hypothesis at a level of significance of $\alpha = 0.05$, however the p-value is still rather small, and the fitted versus residuals plot is much better for the model with the sixth order term. This makes the sixth order model a good choice. We could repeat this process one more time.

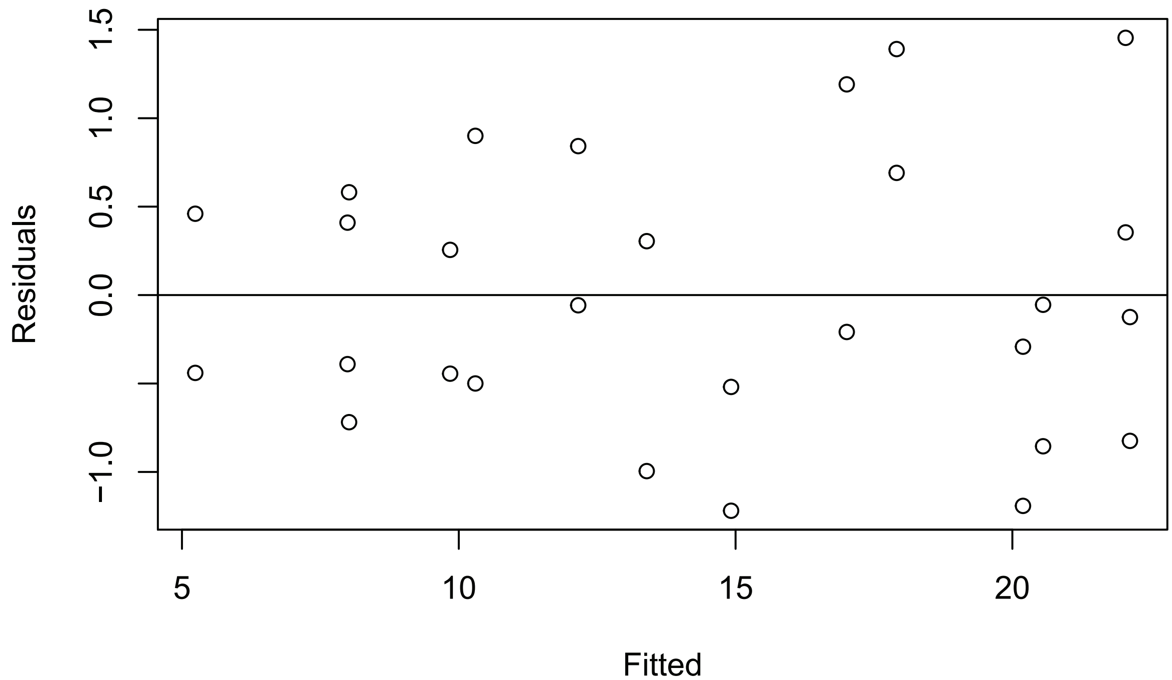
```r
fit8 <- lm(mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) + I(speed^5)
           + I(speed^6) + I(speed^7) + I(speed^8), data = econ)
summary(fit8)
```

```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) +
##     I(speed^5) + I(speed^6) + I(speed^7) + I(speed^8), data = econ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.21938 -0.50464 -0.09105  0.49029  1.45440
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.202e+01  7.045e+01  -0.313    0.758
## speed        6.021e+00  2.014e+01   0.299    0.768
## I(speed^2)  -5.037e-01  2.313e+00  -0.218    0.830
## I(speed^3)   2.121e-02  1.408e-01   0.151    0.882
## I(speed^4)  -4.008e-04  5.017e-03  -0.080    0.937
## I(speed^5)   1.789e-06  1.080e-04   0.017    0.987
## I(speed^6)   4.486e-08  1.381e-06   0.032    0.974
## I(speed^7)  -6.456e-10  9.649e-09  -0.067    0.947
## I(speed^8)   2.530e-12  2.835e-11   0.089    0.930
##
## Residual standard error: 0.9034 on 19 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9741
## F-statistic: 128.1 on 8 and 19 DF,  p-value: 7.074e-15
```

```r
plot_econ_curve(fit8)
```




```r
plot_fit_res(fit8)
```




```r
anova(fit6,fit8)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) + I(speed^5) +
##     I(speed^6)
## Model 2: mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) + I(speed^5) +
##     I(speed^6) + I(speed^7) + I(speed^8)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     21 15.739
## 2     19 15.506  2    0.2324 0.1424 0.8682
```

Here we would clearly stick with `fit6`. The eighth order term is not significant with the other terms in the model and the F-test does not reject.

While we have been using polynomial regression to fit better models, we have also be creating an issue, namely, we are using highly collinear terms. One solution we discussed was to look at centered data, which is particularly useful when the x_i's all lie far from 0.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

$$y_i = \beta_0^* + \beta_1^*(x_i - \bar{x}) + \beta_2^*(x_i - \bar{x})^2 + \epsilon_i$$

Note that $\beta_2 = \beta_2^*$ as we will see in this example data. Also note the differences in the standard errors for the lower order terms.

```
x <- c(280,284,292,295,298,304,308,315)
y <- c(770,800,840,810,735,640,590,560)
x_cent <- x - mean(x)

fit <- lm(y ~ x + I(x^2))
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##      1      2      3      4      5      6      7      8
## -24.09  -1.93  52.91  38.97 -14.25 -48.53 -45.33  42.24
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.503e+04  1.241e+04  -2.016   0.0998 .
## x            1.812e+02  8.364e+01   2.167   0.0825 .
## I(x^2)      -3.179e-01  1.407e-01  -2.259   0.0734 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.54 on 5 degrees of freedom
## Multiple R-squared:  0.8596, Adjusted R-squared:  0.8035
## F-statistic: 15.31 on 2 and 5 DF,  p-value: 0.007383
```

```
fit <- lm(y ~ x_cent + I(x_cent^2))
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x_cent + I(x_cent^2))
##
## Residuals:
##      1      2      3      4      5      6      7      8
## -24.09  -1.93  52.91  38.97 -14.25 -48.53 -45.33  42.24
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 757.1458    24.1013  31.415 6.14e-07 ***
## x_cent       -7.5775     1.5175  -4.993  0.00413 **
## I(x_cent^2)  -0.3179     0.1407  -2.259  0.07344 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.54 on 5 degrees of freedom
## Multiple R-squared:  0.8596, Adjusted R-squared:  0.8035
## F-statistic: 15.31 on 2 and 5 DF,  p-value: 0.007383
```

```
x^2
```

```
## [1] 78400 80656 85264 87025 88804 92416 94864 99225
```

```
cor(x,x^2)
```

```
## [1] 0.9998355
```

```
cor(x_cent,x_cent^2)
```

```
## [1] 0.02492943
```

We also very briefly mentioned orthogonal polynomials, and showed an example with the SUV data. Further details can be found in the text.

```
fit_ortho <- lm(mpg ~ poly(speed,8), data = econ)
summary(fit_ortho)
```

```
##
## Call:
## lm(formula = mpg ~ poly(speed, 8), data = econ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.21938 -0.50464 -0.09105  0.49029  1.45440
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      14.40714    0.17073  84.388  < 2e-16 ***
## poly(speed, 8)1   4.16770    0.90339   4.613  0.00019 ***
## poly(speed, 8)2 -27.66686    0.90339 -30.625  < 2e-16 ***
## poly(speed, 8)3   0.13447    0.90339   0.149  0.88324
## poly(speed, 8)4   7.01671    0.90339   7.767 2.59e-07 ***
## poly(speed, 8)5   0.09289    0.90339   0.103  0.91918
## poly(speed, 8)6  -2.04308    0.90339  -2.262  0.03565 *
## poly(speed, 8)7   0.47529    0.90339   0.526  0.60490
## poly(speed, 8)8   0.08061    0.90339   0.089  0.92983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9034 on 19 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9741
## F-statistic: 128.1 on 8 and 19 DF,  p-value: 7.074e-15
```

```r
summary(fit8)
```

```
##
## Call:
## lm(formula = mpg ~ speed + I(speed^2) + I(speed^3) + I(speed^4) +
##     I(speed^5) + I(speed^6) + I(speed^7) + I(speed^8), data = econ)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.21938 -0.50464 -0.09105  0.49029  1.45440
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.202e+01  7.045e+01  -0.313    0.758
## speed        6.021e+00  2.014e+01   0.299    0.768
## I(speed^2)  -5.037e-01  2.313e+00  -0.218    0.830
## I(speed^3)   2.121e-02  1.408e-01   0.151    0.882
## I(speed^4)  -4.008e-04  5.017e-03  -0.080    0.937
## I(speed^5)   1.789e-06  1.080e-04   0.017    0.987
## I(speed^6)   4.486e-08  1.381e-06   0.032    0.974
## I(speed^7)  -6.456e-10  9.649e-09  -0.067    0.947
## I(speed^8)   2.530e-12  2.835e-11   0.089    0.930
##
## Residual standard error: 0.9034 on 19 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9741
## F-statistic: 128.1 on 8 and 19 DF,  p-value: 7.074e-15
```

```r
#poly(econ$speed,8)
round(cor(poly(econ$speed, 8)),2)
```

```
##   1 2 3 4 5 6 7 8
## 1 1 0 0 0 0 0 0 0
## 2 0 1 0 0 0 0 0 0
## 3 0 0 1 0 0 0 0 0
## 4 0 0 0 1 0 0 0 0
## 5 0 0 0 0 1 0 0 0
## 6 0 0 0 0 0 1 0 0
## 7 0 0 0 0 0 0 1 0
## 8 0 0 0 0 0 0 0 1
```

```r
x <- econ$speed
X <- cbind(x,x^2,x^3,x^4,x^5,x^6,x^7,x^8)
round(cor(X),2)
```

```
##      x
## x 1.00 0.98 0.94 0.89 0.85 0.81 0.78 0.75
##   0.98 1.00 0.99 0.96 0.93 0.90 0.88 0.85
##   0.94 0.99 1.00 0.99 0.98 0.96 0.94 0.92
##   0.89 0.96 0.99 1.00 1.00 0.98 0.97 0.96
##   0.85 0.93 0.98 1.00 1.00 1.00 0.99 0.98
##   0.81 0.90 0.96 0.98 1.00 1.00 1.00 0.99
##   0.78 0.88 0.94 0.97 0.99 1.00 1.00 1.00
##   0.75 0.85 0.92 0.96 0.98 0.99 1.00 1.00
```