Polynomial Regression Models

Recall, that our definition of linear regression models extends to models, in which several powers of the same predictor variable are included

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

intercept: β 0
main effects: β 1 and 2
quadratic effects: β 11 and 22
interaction effects: β 12

and even to higher order polynomial models in two (or more) variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon$$

how each effect influences the model? (in shape and position)

Polynomials are widely used in regression if the relationship between the predictor(s) and response is not linear but curve-linear.

Polynomial Models in One Variable

DEFINITION: The regression model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

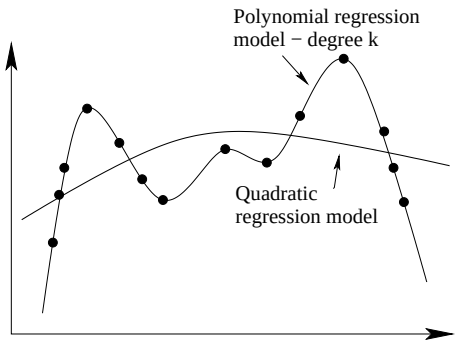
is called a SECOND-ORDER MODEL or a QUADRATIC MODEL in one variable. The expected value of y is a parabola in x. Here, β_0 can still be interpreted as the y-intercept of the parabola. β_1 is called the linear effect parameter and β_2 is called the quadratic effect parameter. The regression model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k + \epsilon$$

is called a k-TH ORDER POLYNOMIAL MODEL in one variable.

Polynomial models may be analyzed with the techniques we have previously developed for multiple regression models, if we use $x_j = x^j$ for the j^{th} predictor variable. There are several important decisions that have to be made when a polynomial regression model is fit. They are discussed in more detail below.

1. Order of the Model Since our goal is to model the data well with the simplest possible regression model, polynomial models of lower degree are usually preferable. Keep in mind, that most datasets of $k+1$ observations can be (perfectly!) modeled by a polynomial of degree k. That clearly cannot be the goal. In practice, we usually start with models of degree one, and - if transformations on the predictor or the response are insufficient - also consider models of degree two. Higher degree models should be avoided unless the context the data is coming from explicitly calls for one of these models.



2. Model-Building Strategy To decide the appropriate degree of a polynomial regression model, two different strategies are possible. One can start with a linear model and include higher order terms one by one until the highest order term becomes non-significant (look at p-values for t-test for slopes). This method is generally called FORWARD VARIABLE SELECTION. Or one could start with a high order model and exclude the non-significant highest order terms one by one until the remaining highest order term becomes significant. This method is generally referred to as BACKWARD VARIABLE SELECTION. In general, the two methods do not have to lead to the same model. For polynomial models, these methods are likely overpowered, since we can restrict our attention to first and second order polynomial models.

3. Extrapolation You have to take extreme care when making predictions outside the range of observed variables in polynomial models. Recall the windmill data, where we could have modeled the non-linear increase in DC output as a polynomial function of wind velocity. Obviously, the DC output should not decrease as wind velocity increases, so predictions made on the part of the parabola with negative slope would be incorrect.

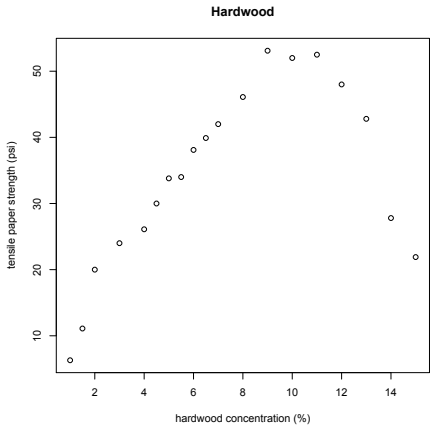
4. Ill-Conditioning Usually, we hope that the predictors in multiple regression models are (almost) independent. If they are highly correlated, we have problems with multicollinearity. The predictors in a polynomial regression are *not* independent. One predictor is x, while the next predictor is x^2, for instance. Even though the correlation between x and x^2 is not perfect, it can be high enough to make the matrix $\mathbf{X}'\mathbf{X}$ ill-conditioned. That means that the matrix is "hard" to invert, or that considerable numerical error will be involved in computing the inverse. It is possible to select the predictor functions more carefully as curve-linear functions of x to avoid this problem.

Multicollinearity becomes more of a concern if the range of predictor values is very narrow, so that x and x^2 are almost linearly related. Some of the ill-conditioning can be remedied by centering the variables (subtracting the mean $\bar{x}$ from x) before taking their powers.

5. Hierarchy Regression models that contain all powers of x (from 1 to k) are said to be hierarchical. It is possible to exclude lower order terms from the model while keeping some of the higher order terms. Statisticians are split on whether this is a good idea. As always, any knowledge about the context the data is taken from should be utilized to decide which polynomial regression model to fit.

Example: The Hardwood Data

The strength of kraft paper is related to the percentage of hardwood in the batch of pulp that the paper is produced from. Not enough hardwood makes the paper weak, but too much hardwood makes the paper brittle. The variables in this dataset are the hardwood concentration (x, in %) and the tensile strength of the paper (y, in psi) for 19 samples of paper. The scatterplot of tensile strength against hardwood percentage (below) shows a clear non-linear relationship:



Without centering the predictor variable, the correlation between x and x^2 is unacceptably high (0.97). Based on the scatterplot, a quadratic model seems like a possibly good fit:

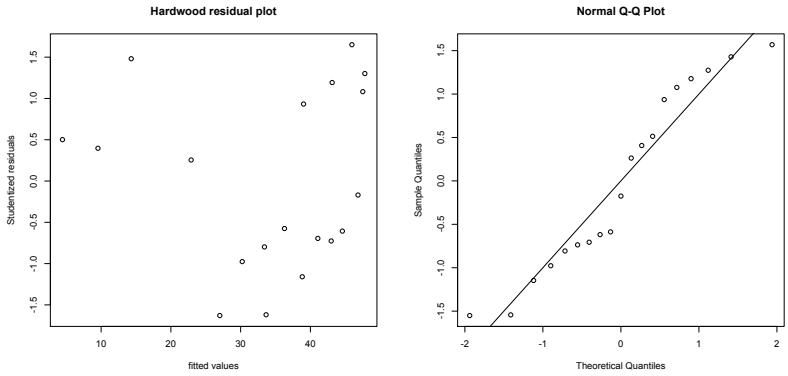
$$y = \beta_0 + \beta_1(x - \bar{x}) + \beta_2(x - \bar{x})^2 + \epsilon$$

The summary statistics for this model fitted with R are shown below:

```
Call:
lm(formula = y ~ poly(x - mean(x), degree = 2, raw = TRUE), data = hardwood)

Residuals:
    Min      1Q  Median      3Q     Max
-5.8503 -3.2482 -0.7267  4.1350  6.5506

Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                           45.29497    1.48287   30.55 1.29e-15 ***
poly(x - mean(x), degree = 2, raw = TRUE)1  2.54634    0.25384   10.03 2.63e-08 ***
poly(x - mean(x), degree = 2, raw = TRUE)2 -0.63455    0.06179  -10.27 1.89e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.42 on 16 degrees of freedom
Multiple R-squared: 0.9085, Adjusted R-squared: 0.8971
F-statistic: 79.43 on 2 and 16 DF,  p-value: 4.912e-09
```

Both the linear and quadratic coefficients (β_1 and β_2) are significant, the model R^2 is 0.9085. Compare this to the R^2-value of 0.3054 for the corresponding linear regression model (output not shown).

A plot of the Studentized residuals against the fitted values $\hat{y}_i$ reveals no outliers and no (strong) patterns. A qq-plot of the standardized residuals shows that the distribution of the residuals is not perfectly Normal.



In this case it is pretty obvious that the quadratic term is a meaningful addition to the model. But we could also test "by-hand" the hypothesis

$$H_0 : \beta_2 = 0 \qquad \text{vs.} \qquad H_a : \beta_2 \neq 0$$

using the extra sums-of-squares method. For the linear model

$$y = \beta_0 + \beta_1(x - \bar{x}) + \epsilon$$

the sum of squares of regression is $SS_R(\beta_1|\beta_0) = 1043.4$. Use the output from the previous page to compute the sum of squares of regression for the quadratic model and to conduct the F-test for the hypotheses mentioned above.

```
> anova(fit.3, fit)
Analysis of Variance Table

Model 1: y ~ I(x - mean(x))
Model 2: y ~ poly(x - mean(x), degree = 2, raw = TRUE)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     17 2373.46
2     16  312.64  1    2060.8 105.47 1.894e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Piecewise Polynomial Fitting - Splines

If low-order polynomials do not provide a good fit for curve-linear data (poor residual plots, for instance) then the cause can be that the data behave differently for different parts of the predictor range. Sometimes, transformations on x or y can help. If transformations don't help, then the predictor range can be divided into segments and a different function is fit in each segment.

DEFINITION: SPLINES are piecewise polynomial functions that satisfy certain "smoothness" criteria. These criteria are often continuity and possibly continuity of the derivative(s). The points where different polynomial pieces are joined together are called the KNOTS of the spline.

Example: A cubic spline ($k = 3$) with continuous first and second derivatives and h knots $t_1 < t_2 < \cdots < t_h$ can be written as

$$S(x) = \sum_{j=0}^{3} \beta_{0j} x^j + \sum_{i=1}^{h} \sum_{j=0}^{3} \beta_{ij} (x - t_i)_+^j$$

where

$$(x - t_i)_+ = \begin{cases} (x - t_i) & \text{if } x - t_i > 0 \\ 0 & \text{if } x - t_i \leq 0 \end{cases}$$

If the positions of the knots in a spline are known, then fitting a spline function to data reduces to a nonlinear regression problem. If the positions are not known, the problem becomes more complicated. It is not easy to decide how many knots to use and how they should be placed. In general, each piece of the spline should be kept as simple as possible to avoid over-fitting the data.

A special case of spline functions are the piecewise-linear functions. As with splines, we can assume piecewise linear functions to be continuous, or we can allow discontinuities at the knots.

Example: Consider a (not necessarily continuous) piecewise linear function with a single knot at t:

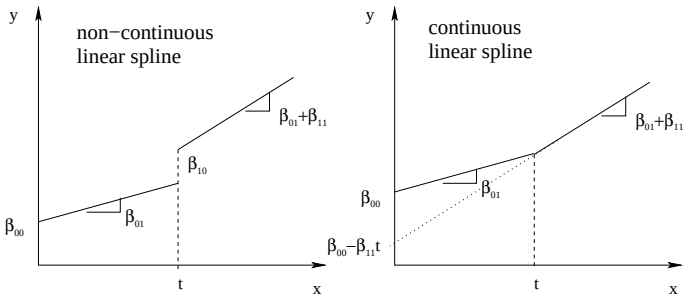
$$S(x) = \beta_{00} + \beta_{01} x + \beta_{10} (x - t)_+^0 + \beta_{11} (x - t)_+^1$$

If $x \leq t$ (before the knot), the function is the line $y = \beta_{00} + \beta_{01} x$. If $x > t$ (after the knot) the function is

$$y = \beta_{00} + \beta_{01} x + \beta_{10} + \beta_{11} (x - t) = (\beta_{00} + \beta_{10} - \beta_{11} t) + (\beta_{01} + \beta_{11}) x$$

Notice, that β_{10} is the height of the vertical "jump" at the knot when $x = t$. If we require the piecewise linear function to be continuous, then that means that β_{10} should be equal to zero.

$$S(x) = \beta_{00} + \beta_{01} x + \beta_{11} (x - t)_+^1$$



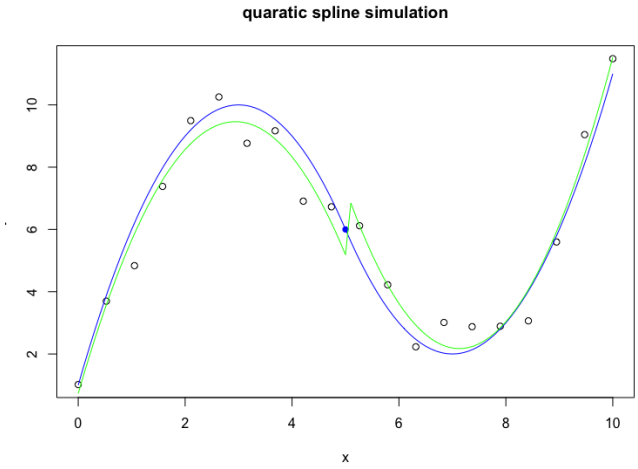
Example: How does one fit spline models in R? Consider data generated from a quadratic spline model with one knot at x=5. The spline we're generating data from (see Spline simulation code on Canvas) happens to be continuous and smooth. There are n=20 data points simulated to come from the following two polynomials of degree 2:

$$\text{if x<5, then } y(x) = 10 - (x-3)^2 + \epsilon$$

$$\text{if x} \geq 5\text{, then } y(x) = 2 + (x-7)^2 + \epsilon$$

These two parabola curves from which data are simulated are shown as the blue lines in the plot below. Here, errors ϵ are assumed to be IID Normal(0,1) random variables.

How many β-parameters does the quadratic spline model with one knot t have? What are the columns of the model matrix X?



Just like in any other multiple regression model, the slope parameters of the model can be estimated in R and predicted values can be produced based off that model. The green curve in the plot shows the predicted spline model. Note, that the predicted model is not continuous at $t = 5$. What would you have to do to force it to be continuous?

Nonparametric Regression

So far, we have discussed linear regression and polynomial regression (and some more exotic variants of regression). What all these models have in common is that they specify a functional relationship (line, plane, parabolic surface etc.) between the predictors and the response. And so far, we (the users) have always been the ones with the ultimate decision of which model to use. There is an alternative. We could not specify a model and instead allow the data to "pick" its own model.

In nonparametric regression, the regression function does not take any predetermined shape but is derived entirely from the data. The conventional parametrized regression model is

$$y_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + \epsilon_i$$

where we specify the general class of the function f (linear, quadratic, etc.) and use the data to estimate the function parameters $\boldsymbol{\beta}$. The nonparametric regression model is similar

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

but now our (more ambitious) goal is to estimate the function f itself. Of course, without any restrictions, there are uncountably infinitely many choices. So we'll restrict the problem a bit.

Recall, that in ordinary linear least squares regression, the predicted values $\hat{\mathbf{y}}$ can be written as linear combinations of the observed values $\mathbf{y}$. The coefficients in the linear combinations are determined by the entries of the hat-matrix.

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$$

Most nonparametric regression models also model the predicted values as linear combinations of the observations, but with different weights.

Kernel Regression

As in the OLS case, the predicted observations are computed as weighted sums of the actual observations. The weights are set with the help of a kernel function. Let $\tilde{y}_i$ be the kernel smoother estimate for observation y_i. Then

$$\tilde{y}_i = \sum_{j=1}^{n} w_{ij} y_j$$

where the weights w_{ij} sum to one (over j). Alternatively, we could write

$$\tilde{\mathbf{y}} = \mathbf{S}\mathbf{y}, \quad \text{where} \quad \mathbf{S} = (w_{ij})$$

$\mathbf{S}$ is called the smoothing matrix. The weights are typically chosen such that $w_{ij} = 0$ for all points y_j outside of a specified "neighborhood" of the point y_i (in x-direction). The width of this neighborhood is sometimes also called the BANDWIDTH (b) of the

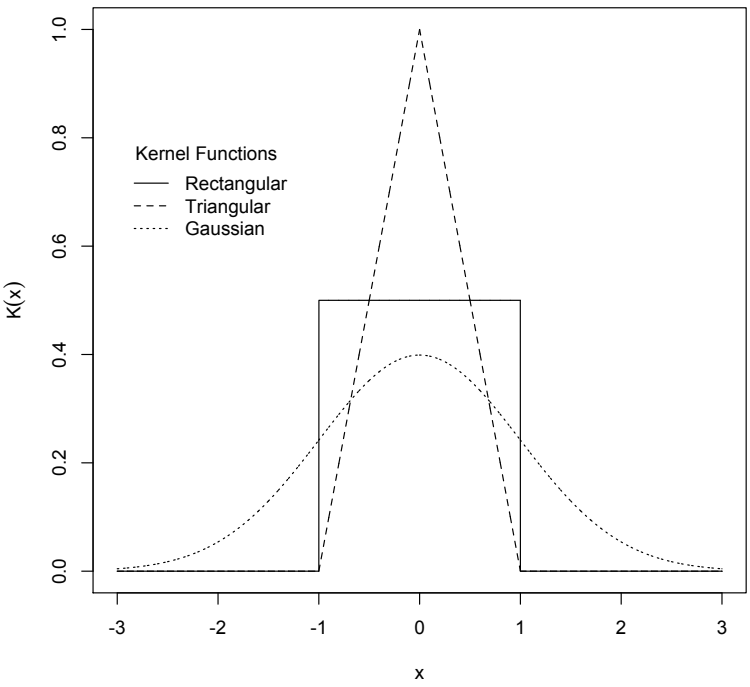
kernel. The larger the bandwidth is chosen, the smoother the estimated function will become.

There are many possible kernel functions that could be used to determine the weights. They need to satisfy the following properties:

- $K(t) \geq 0$ for all t
- $\int_{-\infty}^{\infty} K(t)dt = 1$
- $K(-t) = K(t)$ (symmetry).

Note, that these are the properties of symmetric probability density functions. Which means that for instance the Normal distribution (Gaussian kernel), the triangular distribution (triangular kernel), and the uniform distribution (uniform or box kernel) make good kernel functions.

Gaussian kernel function	$K(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$
Triangular kernel function	$K(t) = \begin{cases} 1 - \lvert t \rvert & t \leq 1 \\ 0 & t > 1 \end{cases}$
Uniform kernel	$K(t) = \begin{cases} 0.5 & \lvert t \rvert \leq 1 \\ 0 & \lvert t \rvert > 1 \end{cases}$



The weights for the kernel smoother are then chosen as

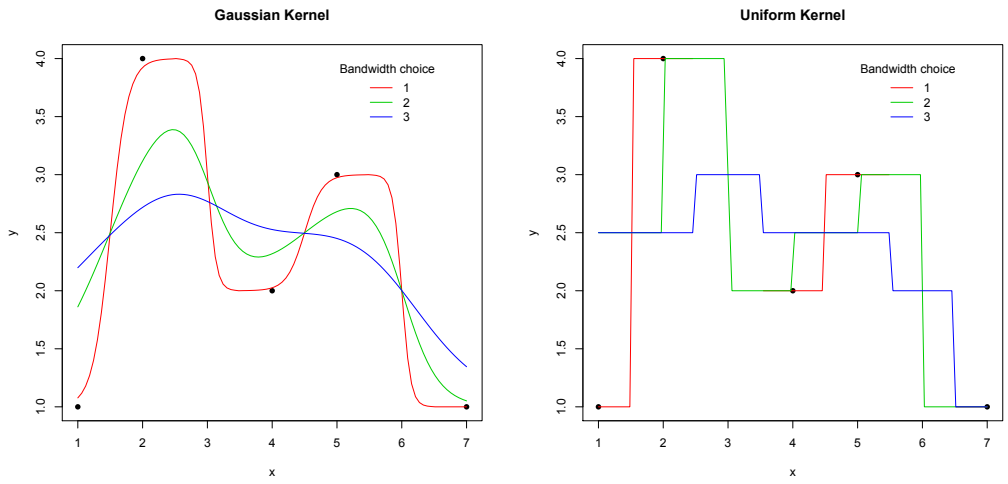
$$w_{ij} = \frac{K\left(\frac{x_i - x_j}{b}\right)}{\sum_{k=1}^{n} K\left(\frac{x_i - x_k}{b}\right)}$$

Note, that the weight w_{ij} determines how much observation y_j will matter in predicting $\hat{y}_i$. The kernel functions use the distance between x_i and x_j divided by the bandwidth as their arguments. Points x_j that are further away than the bandwidth b from x_i will be ignored in some kernels.

Example: Suppose you want to fit a nonparametric regression curve to the following data:

x	1	2	4	5	7
y	1	4	2	3	1

The R-function `ksmooth()` computes a kernel smoother (options are Gaussian and uniform kernels) with chosen bandwidth. The results for three different bandwidths (1,2, and 3) are shown below.



Note: The regression "function" in this case is a sequence of points for which we have computed predictions. There is no algebraic function that is estimated.

For large sample sizes, the bandwidth of the kernel to be used becomes much more important than the choice of the kernel function.