Statistical Learning Theory and Model Validation

Matthew Drury

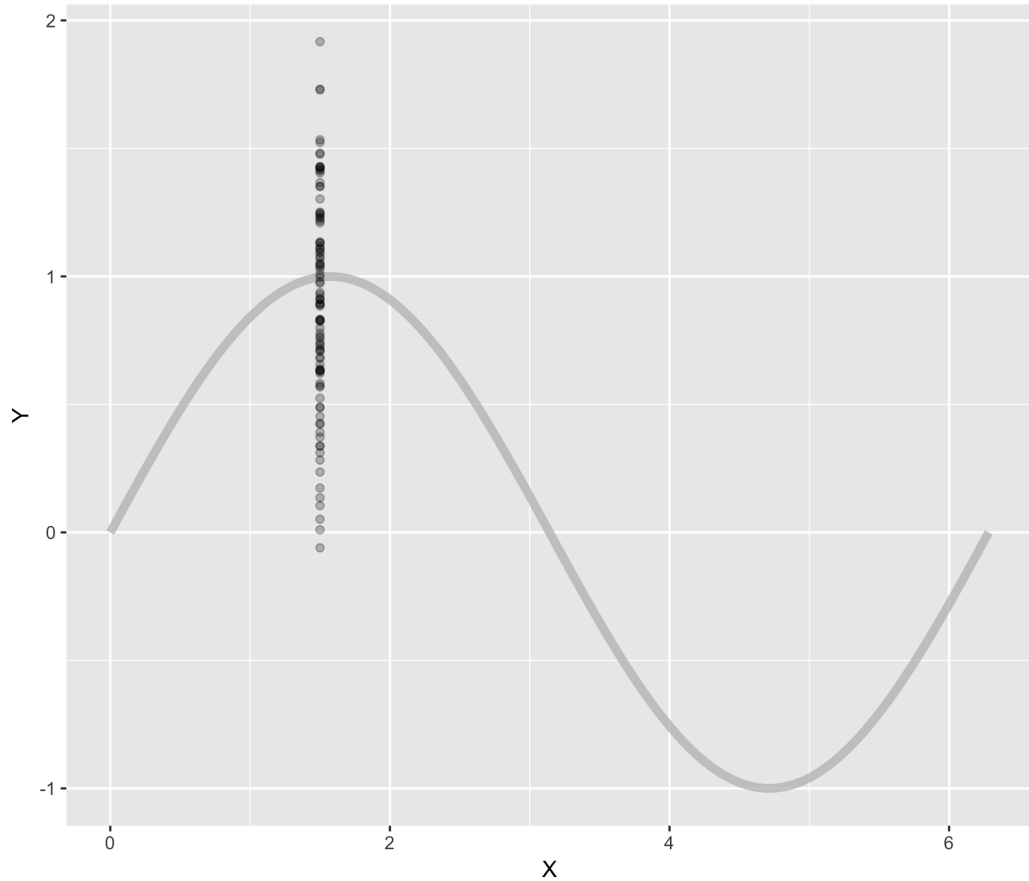
May 10, 2017

Introduction

In this talk we will discuss the error incurred when generalizing a learning model to unseen data.

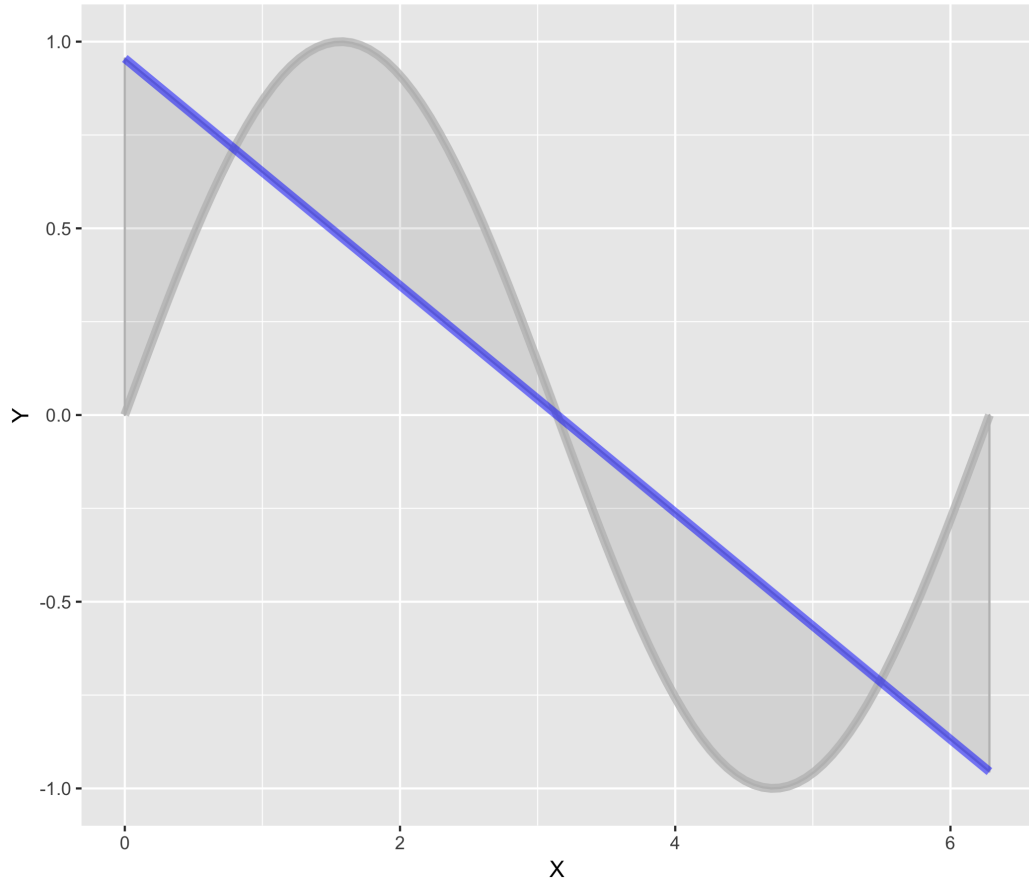
We will see that this error comes from three distinct sources:

First is the error that results from the inherent randomness in the quantity being modeled.



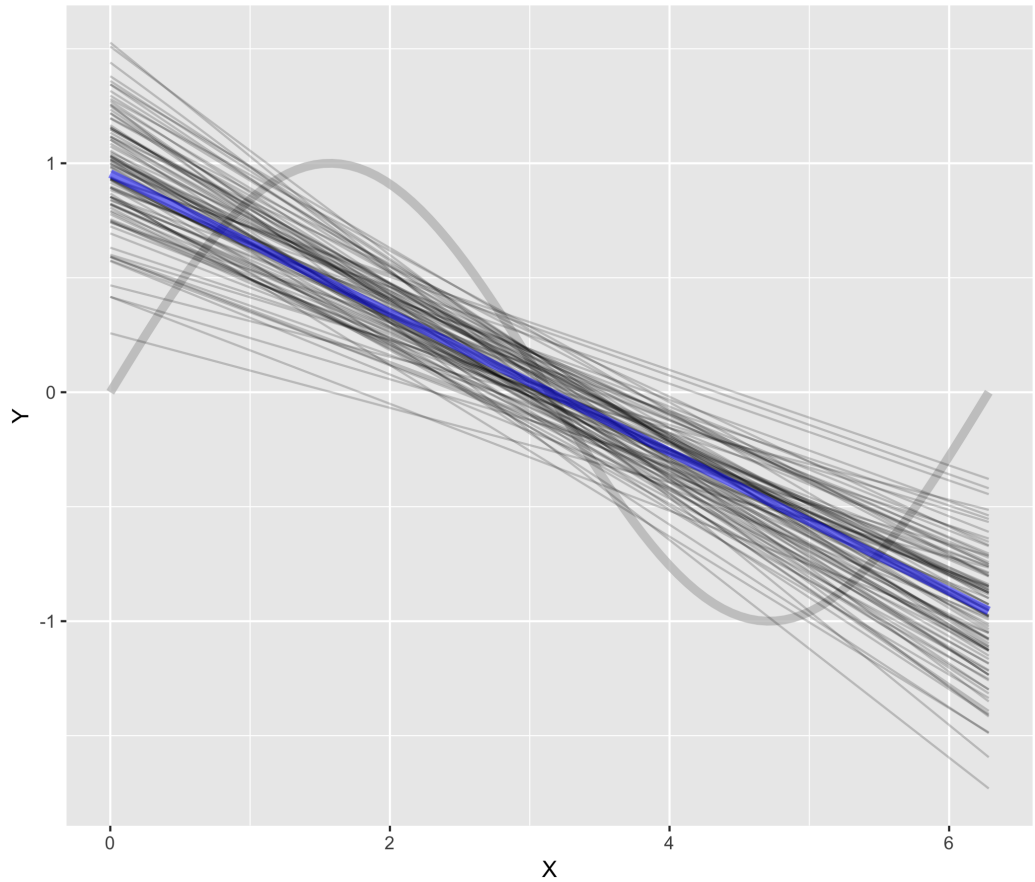
This is called the **irreducible error**

Second is the error incurred from misspecification of the model, resulting in an inability of the model to adapt to the target.



This is called the **model bias**.

Finally is the error incurred because our training data is not fully informative of random process at work, causing the model to deviate from the ideal fit.



This is called the **model variance**.

A Framework for Statistical Learning

In this section we will outline a very general framework we can use to study statistical modeling.

Suppose that we have a jointly distributed random variable X, Y.
The variable X is called the **predictor**.
The variable Y is called the **response**.

In general, we think of the distribution of X as unknowable, and our goal is to learn something about the conditional distribution $Y|X$.

The simplest thing we could want to know is the expectation $E(Y|X)$, which is a pure function of X, and is often referred to as the **regression function**.

An attractive method for estimating approximations to the regression function is to minimize some **loss functional** chosen so that the minimizer is "close" to the regression function.

The most popular choice is the **squared error loss**.

$$\mathbf{ESE}(f) = E_{X,Y}\left[(y - f(x))^2\right]$$

which is ubiquitous for good reason: the pointwise minimizer of

$$\mathbf{ESE}(f; x) = E_{Y}\left[(y - f(x))^2 \mid x\right]$$

is the regression function.

$$\mathbf{ESE}(f; x) = E_Y\left[(y - f(x))^2 \mid x\right]$$

$$\begin{aligned}\textbf{ESE}(f;x) &= E_Y\left[(y-f(x))^2 \mid x\right] \\ &= E_Y\left[(y-E[y \mid x]+E[y \mid x]-f(x))^2 \mid x\right]\end{aligned}$$

The most common trick in mathematics: add zero.

$$
\begin{aligned}
\mathbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right] \\
&\quad + 2E_Y\left[(y - E[y \mid x])(E[y \mid x] - f(x)) \mid x\right]
\end{aligned}
$$

Square.

$$
\begin{aligned}
\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right] \\
&\quad + 2E_Y\left[(y - E[y \mid x])\,(E[y \mid x] - f(x)) \mid x\right]
\end{aligned}
$$

This factor has no dependence on y, so it is a constant from the view of the outside expectation.

$$
\begin{aligned}
\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right] \\
&\quad + 2E_Y\left[(y - E[y \mid x])\,(E[y \mid x] - f(x)) \mid x\right]
\end{aligned}
$$

This factor is zero in expectation, so the cross term is zero.

$$
\begin{aligned}
\mathbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right]
\end{aligned}
$$

Goodbye!

$$
\begin{aligned}
\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right] \\
&\geq E_y\left[(y - E[y \mid x])^2 \mid x\right]
\end{aligned}
$$

Discarding a positive term.

$$
\begin{aligned}
\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x] + E[y \mid x] - f(x))^2 \mid x\right] \\
&= E_Y\left[(y - E[y \mid x])^2 \mid x\right] + E_Y\left[(E[y \mid x] - f(x))^2 \mid x\right] \\
&\geq E_y\left[(y - E[y \mid x])^2 \mid x\right]
\end{aligned}
$$

The regression function $E[Y|X]$ is the minimizer.

Because of it's importance, let's reserve the symbol $\mathcal{F}$ for the regression function:

$$\mathcal{F}(x) = E_Y[Y|X]$$

It is also common to refer to $\mathcal{F}$ as the **ground truth**, just the **truth**, or the **signal**.

The Bias-Variance Decomposition

Although we never have full knowledge about X, Y, we often do have sample data drawn from this distribution:

$$\mathcal{D} = \{(x_i, y_i) \mid x_i, y_i \sim X, Y\}$$

Approximating $\mathcal{F}$ often takes the form of a **learning algorithm**:

$$\mathcal{A} : \mathcal{D} \mapsto f$$

which, given a sample dataset $\mathcal{D}$, produces a function f that approximates $\mathcal{F}$.

A learning algorithm induces an extremely enlightening decomposition of the expected squared error. This is called the **bias-variance** decomposition.

Recall our decomposition of the expected squared error from the previous section:

$$\begin{aligned}\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\ &= E_Y\left[(y - \mathcal{F}(x))^2 \mid x\right] + E_Y\left[(\mathcal{F}(x) - f(x))^2 \mid x\right]\end{aligned}$$

$$\begin{aligned}\textbf{ESE}(f;x) &= E_Y\left[(y-f(x))^2 \mid x\right] \\ &= E_Y\left[(y-\mathcal{F}(x))^2 \mid x\right] + E_Y\left[(\mathcal{F}(x)-f(x))^2 \mid x\right]\end{aligned}$$

This term cannot be reduced by a learning algorithm, it measures the variance of Y about its mean. This is called the **irreducible error**

$$\begin{aligned}\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\ &= E_Y\left[(y - \mathcal{F}(x))^2 \mid x\right] + E_Y\left[(\mathcal{F}(x) - f(x))^2 \mid x\right]\end{aligned}$$

This term does not depend on Y, and so the expectation can be discarded.

$$\begin{aligned}\textbf{ESE}(f; x) &= E_Y\left[(y - f(x))^2 \mid x\right] \\ &= E_Y\left[(y - \mathcal{F}(x))^2 \mid x\right] + (\mathcal{F}(x) - f(x))^2\end{aligned}$$

We can reduce this term by choosing f well, and it is the goal of the learning algorithm to make this term as small as possible. We call it the **reducible error**.

$$\mathbf{IESE}(x) = E_Y\left[(y - \mathcal{F}(x))^2 \mid x\right]$$

$$\mathbf{RESE}(f; x) = (\mathcal{F}(x) - f(x))^2$$

The second term in the previous equation can be further decomposed, but to do so we will have to introduce a new concept.

Recall that f depends on the data set $\mathcal{D}$ through our learning algorithm:

$$\mathcal{A} : \mathcal{D} \mapsto f$$

We can make this dependence explicit by writing $f(x; \mathcal{D})$.

The datasets $\mathcal{D}$ (of a fixed size) can be thought of as being drawn from their own distribution, the **sampling distribution** of X.

We would like to study how the expected error of our predictions depends on the randomness in $\mathcal{D}$:

$$\textbf{ESE}(f; x) = E_{Y,\mathcal{D}}\left[(y - f(x; \mathcal{D}))^2 \mid x\right]$$

Note that the previous decomposition into irreducible and reducible error still holds for this expectation, as our calculations made no assumptions about f.

To break down the reducible error, we introduce the expectation of f with respect to the data $\mathcal{D}$:

$$Ef(x) = E_{\mathcal{D}}\left[f(x, \mathcal{D}) \mid x\right]$$

$$\textbf{RESE}(f; x)$$
$$= E_D \left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x \right]$$

$$\textbf{RESE}(f; x)$$
$$= E_D \left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x \right]$$
$$= E_{\mathcal{D}} \left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x \right]$$

Add zero.

$$
\begin{aligned}
&\textbf{RESE}(f; x) \\
&\quad = E_D\left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))^2 \mid x\right] + E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\qquad + 2E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))(Ef(x) - f(x, \mathcal{D})) \mid x\right]
\end{aligned}
$$

Square.

$$
\begin{aligned}
&\textbf{RESE}(f; x) \\
&\quad = E_D\left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))^2 \mid x\right] + E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\qquad + 2E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))\,(Ef(x) - f(x, \mathcal{D})) \mid x\right]
\end{aligned}
$$

This factor has no dependence on $\mathcal{D}$, so it is a constant from the view of the enclosing expectation.

$$
\begin{aligned}
&\textbf{RESE}(f; x) \\
&\quad = E_D\left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x\right] \\
&\quad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))^2 \mid x\right] + E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\qquad + 2E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))\,(Ef(x) - f(x, \mathcal{D})) \mid x\right]
\end{aligned}
$$

This factor is zero in expectation, so the cross term is zero.

$$\begin{aligned}
&\textbf{RESE}(f; x) \\
&\qquad = E_D\left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x\right] \\
&\qquad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x\right] \\
&\qquad = E_{\mathcal{D}}\left[(\mathcal{F}(x) - Ef(x))^2 \mid x\right] + E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right]
\end{aligned}$$

This term has no dependence on $\mathcal{D}$, so we can remove the expectation.

$$
\begin{aligned}
&\textbf{RESE}(f; x) \\
&\qquad = E_D \left[(\mathcal{F}(x) - f(x, \mathcal{D}))^2 \mid x \right] \\
&\qquad = E_{\mathcal{D}} \left[(\mathcal{F}(x) - Ef(x) + Ef(x) - f(\mathcal{D}))^2 \mid x \right] \\
&\qquad = (\mathcal{F}(x) - Ef(x))^2 + E_{\mathcal{D}} \left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x \right]
\end{aligned}
$$

This is the **bias-variance decomposition.**

$$\textbf{RESE}(f; x)$$

$$= \left(\mathcal{F}(x) - Ef(x)\right)^2 + E_{\mathcal{D}}\left[\left(Ef(x) - f(x, \mathcal{D})\right)^2 \mid x\right]$$

This is the **model (squared) bias**, which measures the deviation of the algorithm's average result approximation from the ground truth.

$$\textbf{RESE}(f; x)$$
$$= (\mathcal{F}(x) - Ef(x))^2 + E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right]$$

This is the **model variance**, which measures the variance of the algorithm's results around its average result.

$$\textbf{BIAS}(x)^2 = (\mathcal{F}(x) - Ef(x))^2$$

The model bias tends to be decrease as the learning algorithm becomes more complex, and increase as it becomes more rigid.

$$\mathbf{VAR}(x) = E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right]$$

The model variance tends to increase as the learning algorithm becomes more complex, and decrease as it becomes more rigid.

It is possible to make the phrases “learning algorithm becomes more/less complex” precise, but it is not easy, and it will have to wait for another talk.

Here is our final decomposition of the expected squared error into various sources of error:

$$\begin{aligned}
&\textbf{ESE}(f; x) \\
&\quad = E_Y\left[(y - E_Y[y \mid x])^2 \mid x\right] \\
&\qquad + (E_Y[y \mid x] - E_{\mathcal{D}}\left[f(x, \mathcal{D}) \mid x\right])^2 \\
&\qquad + E_{\mathcal{D}}\left[(E_{\mathcal{D}}\left[f(x, \mathcal{D}) \mid x\right] - f(x, \mathcal{D}))^2 \mid x\right]
\end{aligned}$$

Irreducible Error - Model Bias - Model Variance

So far, we have concentrated on the **pointwise** error rates **ESE**(x). Often time, we concern ourselves with the **overal** error rates, which are found by taking the expectation with respect to X as well:

$$\begin{aligned}\textbf{ESE}(f) &= E_{X,Y}\left[(y - f(x))^2\right] \\ &= E_X E_{Y|X}\left[(y - f(x))^2 \mid x\right]\end{aligned}$$

Discussions of this nature can often become confusing if you do not keep in mind whether the **overall** of **pointwise** error is being considered. We will try to be explicit.

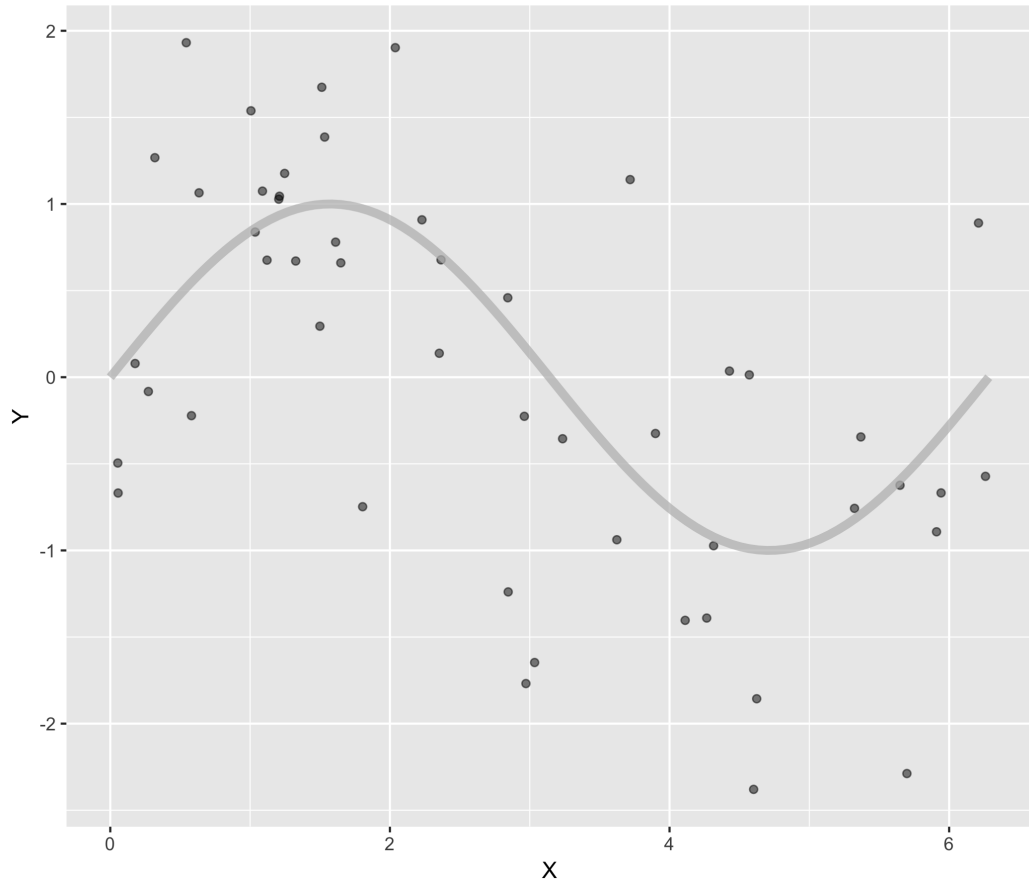
A Toy Model

So far we have worked out a sound theoretical foundation to understand the errors incurred when building learning algorithms. In this section we will analyze in detail how these concepts look with a toy model.

Our data generating process will be very simple so that we can fully analyse the situation:

$$X \sim U(0, 2\pi)$$
$$Y \sim \sin(X) + N(0, \epsilon)$$

Where U is the uniform distribution on an interval, and N is the normal distribution with a given mean and variance.



Clearly, the regression function $E(Y \mid X)$ is given by:

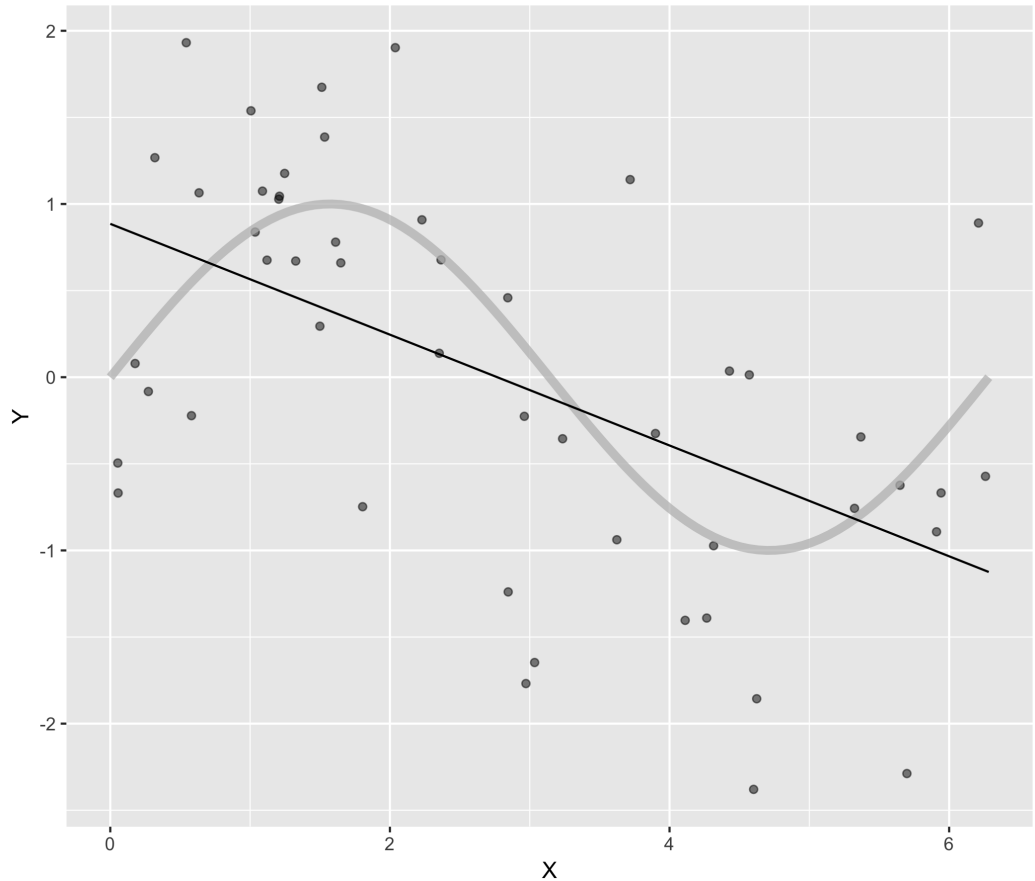
$$\mathcal{F}(X) = E[\sin(x) + N(0, \epsilon) \mid x] = \sin(x)$$

The irreducible error component, which does not depend on our choice of learning algorithm, is easy to compute straight from the definition:

$$\begin{aligned}
\textbf{IESE}(x) &= E_Y\left[(y - \mathcal{F}(x))^2 \mid x\right] \\
&= E_Y\left[(\sin(x) + N(0,\epsilon) - \sin(x))^2\right] \\
&= E_Y\left[N(0,\epsilon)^2\right] \\
&= \epsilon
\end{aligned}$$

We take as our learning algorithm **linear regression**:

$$\text{LinReg} : \mathcal{D} \mapsto \text{LinReg}(\mathcal{D}_X, \mathcal{D}_Y)$$



Let's study the bias of our toy model. Recall the definition:

$$\textbf{BIAS}(x)^2 = (\mathcal{F}(x) - Ef(x))^2$$

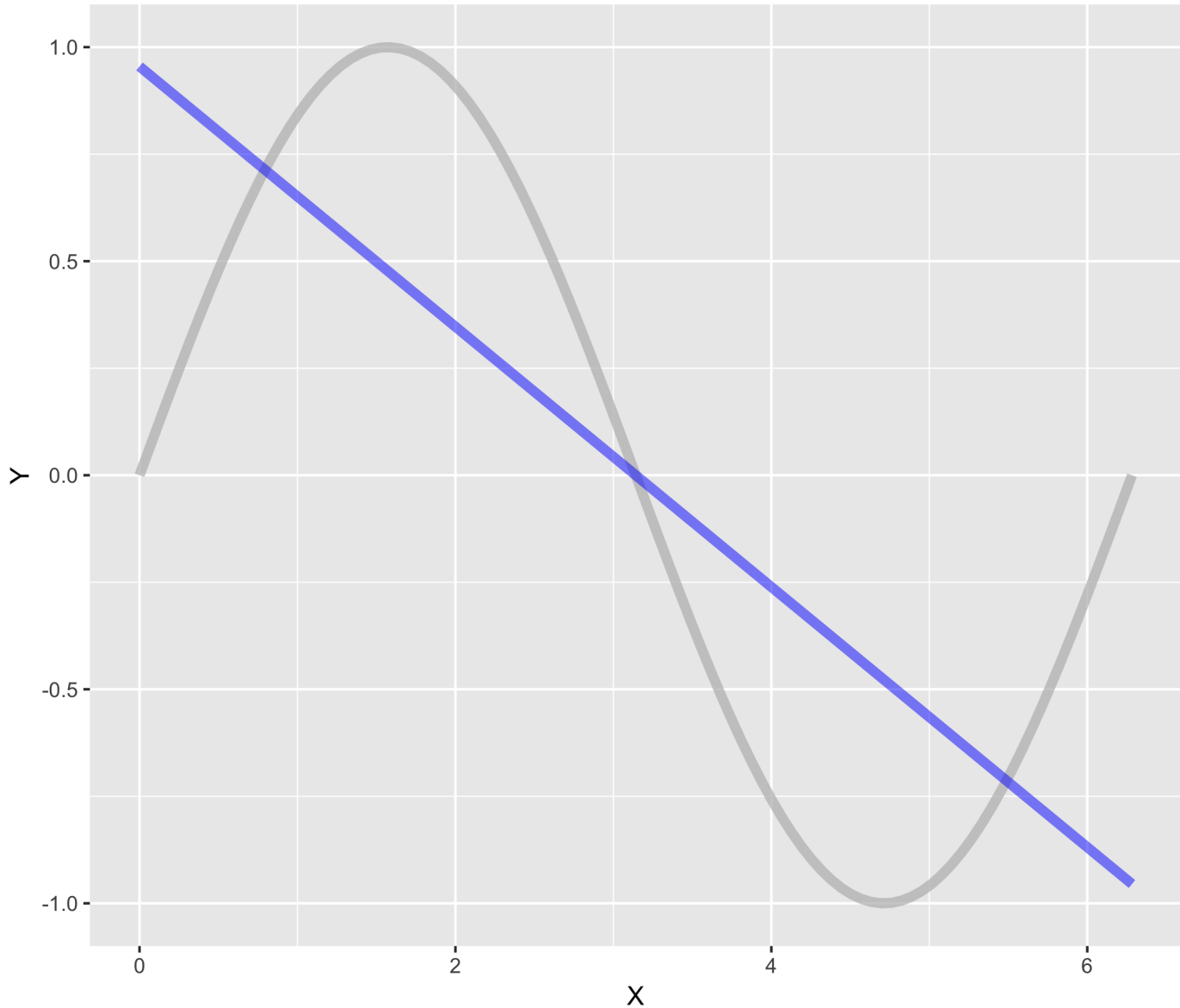
Where:

$$Ef(x) = E_D\left[f(x;\mathcal{D}) \mid x\right]$$

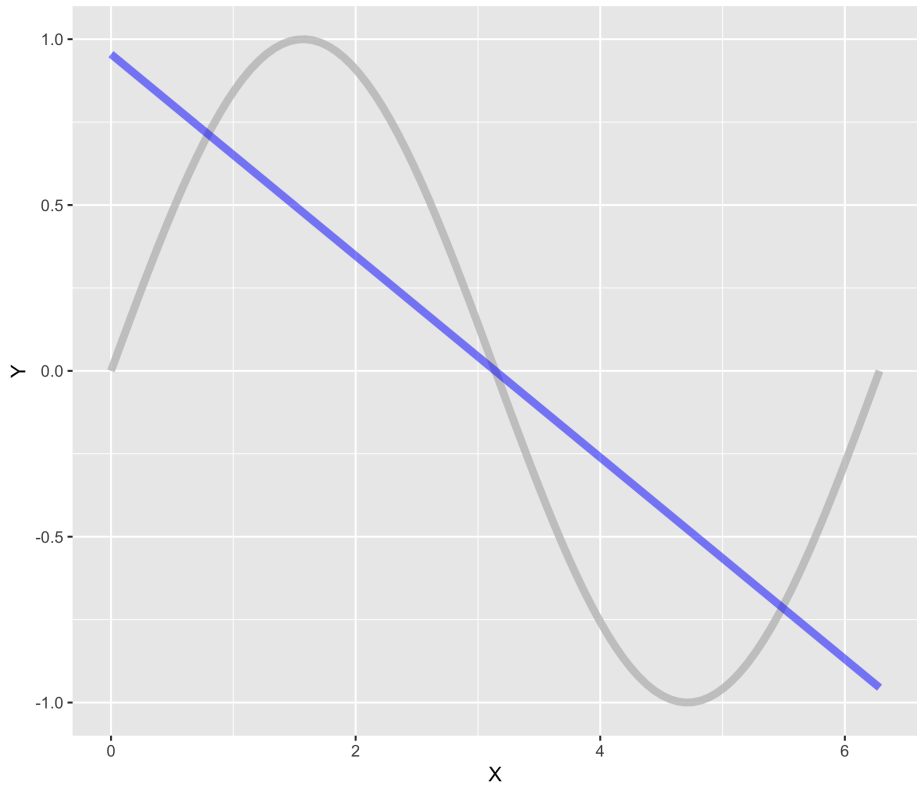
is the expected output of our modeling algorithm.

For our toy situation we can calculate Ef numerically (I used scipy):

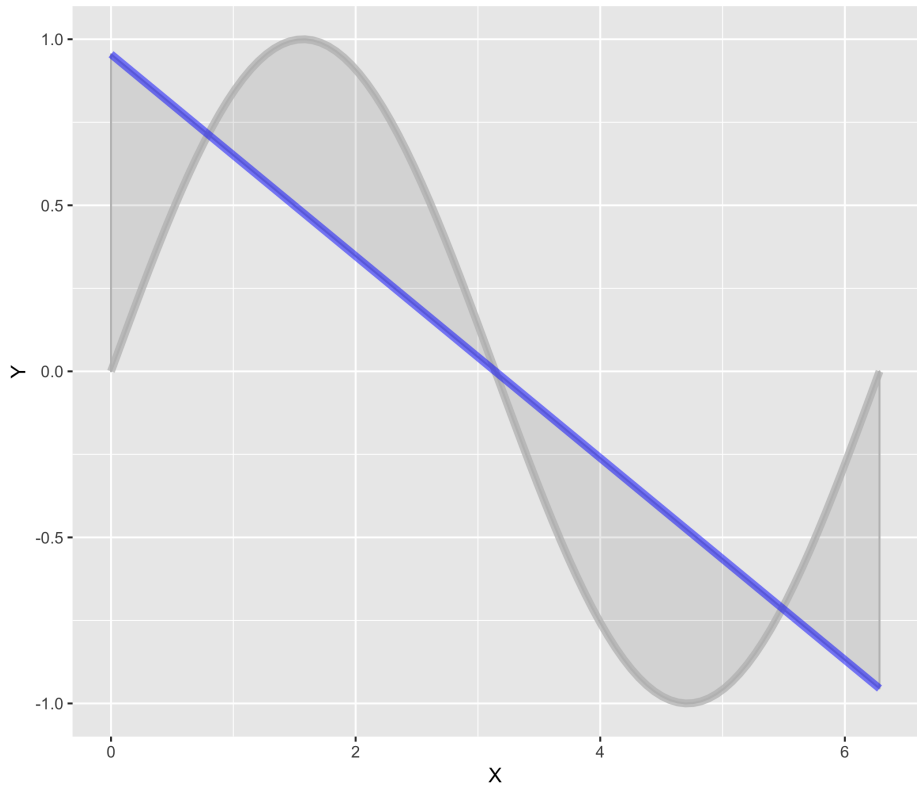
$$Ef(x) \approx -0.304x + 0.955$$


1.0
0.5
0.0
-0.5
-1.0
Y
0
2
4
6
X

The bias at a point is the square of the vertical distance between the true signal and the best linear fit.



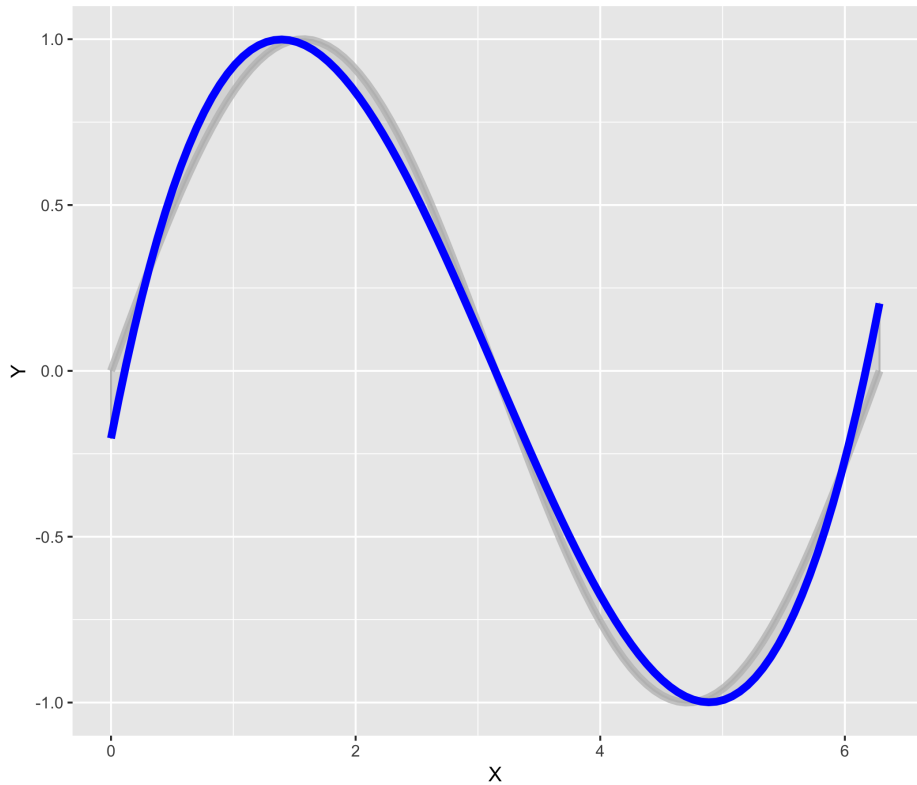
The total bias is the expectation of the pointwise bias. Visually, we can think of the unsigned area between the best linear fit and the true signal:



The total bias can be explicitly calculated in this case (I used a numerical integration routine):

$$\textbf{BIAS}^2 = E_X\left[\left(\mathcal{F}(x) - Ef(x)\right)^2\right] \approx 1.23$$

Bias can be lowered by making our learning algorithm more complex. For example, fitting a *cubic* regression lowers the bias of our model considerably:



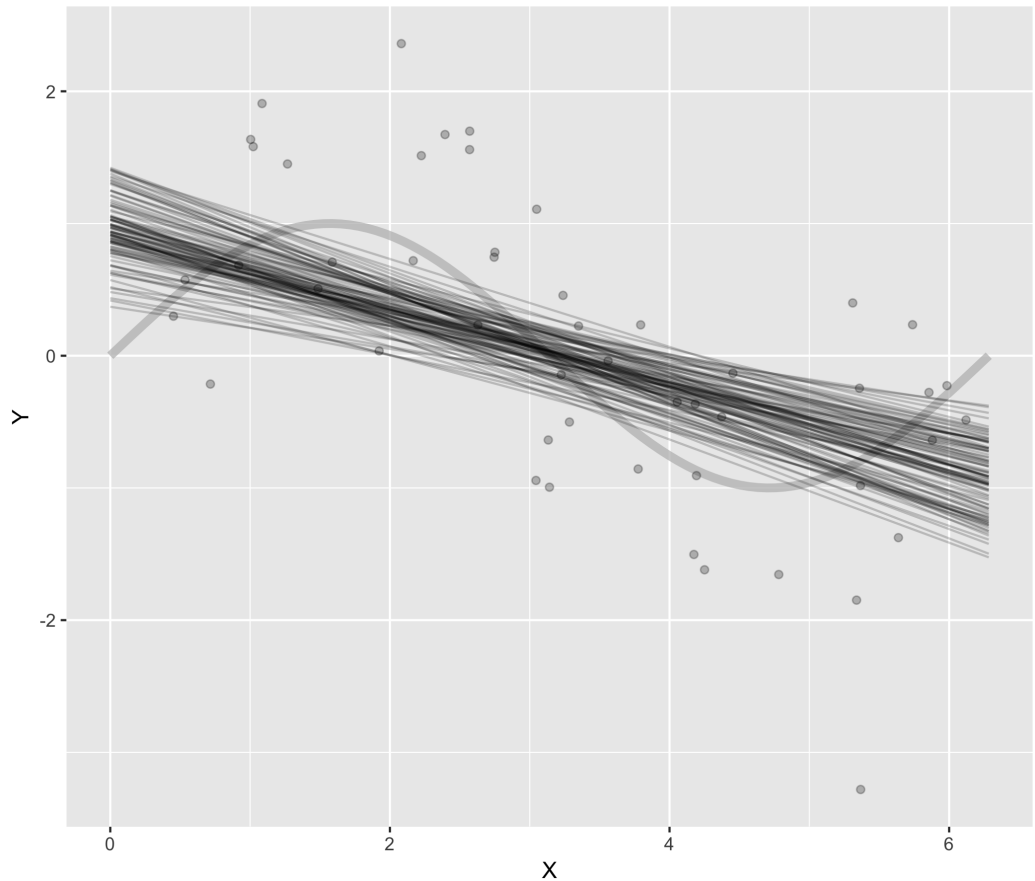
$$\textbf{BIAS}^2 = E_X\left[\left(\mathcal{F}(x) - Ef(x)\right)^2\right] \approx 0.028$$

It is tempting to want to lower the bias as much as possible, as this brings our expected model close to reality, unfortunately, the model variance is a price we pay.

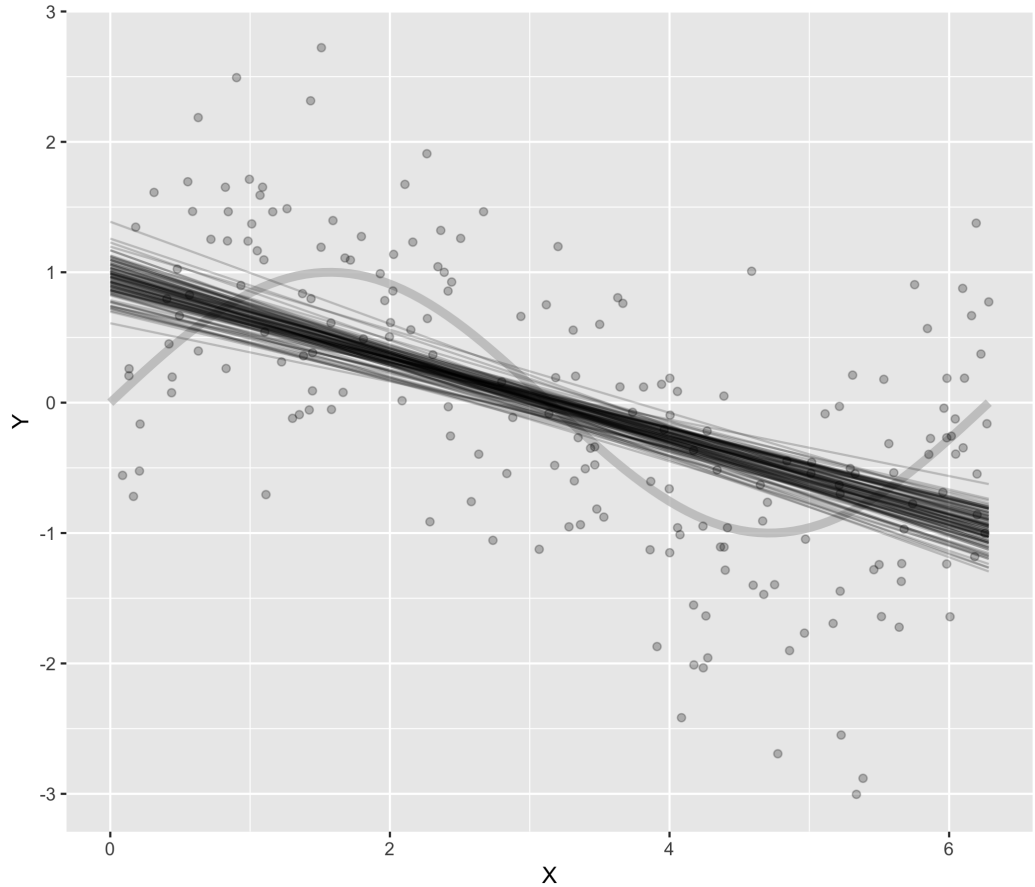
The model variance measures how sensitive our estimate is to the specific training data used:

$$E_{\mathcal{D}}\left[(Ef(x) - f(x, \mathcal{D}))^2 \mid x\right]$$

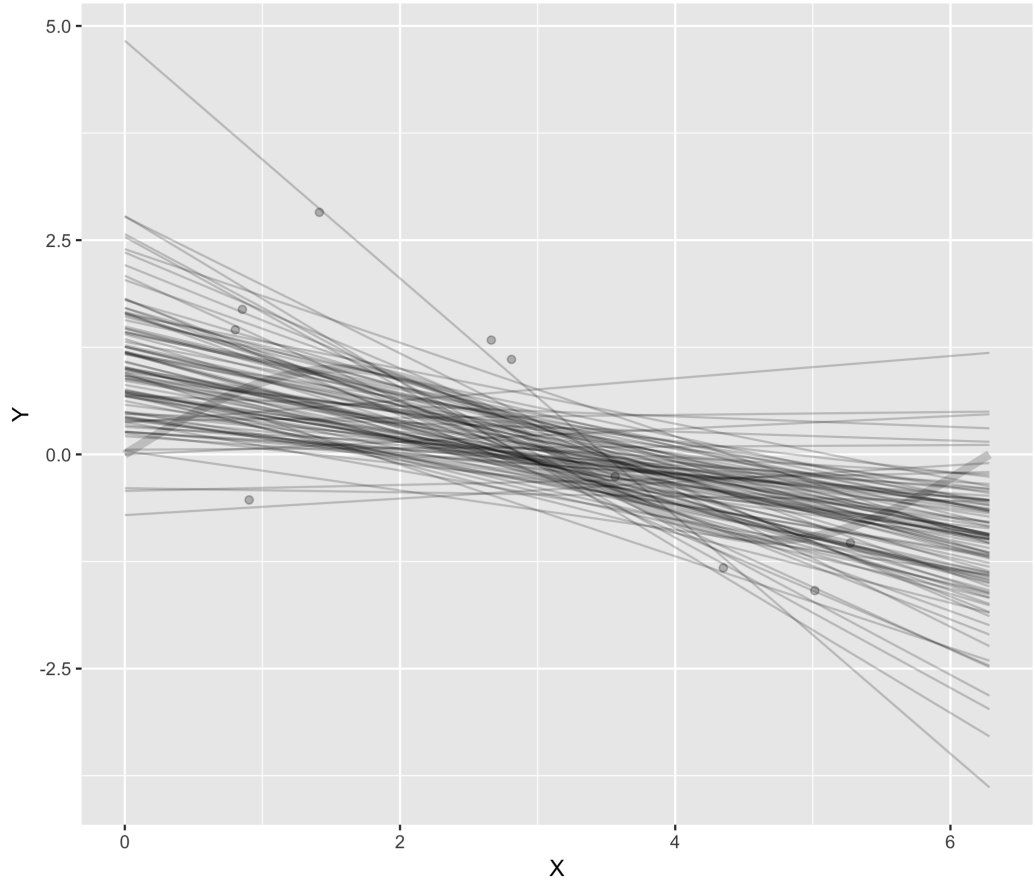
Lines fit to different data sampled from our model distribution tend to cluster around the best linear fit, but there is a fair amount of variance in the clustering:



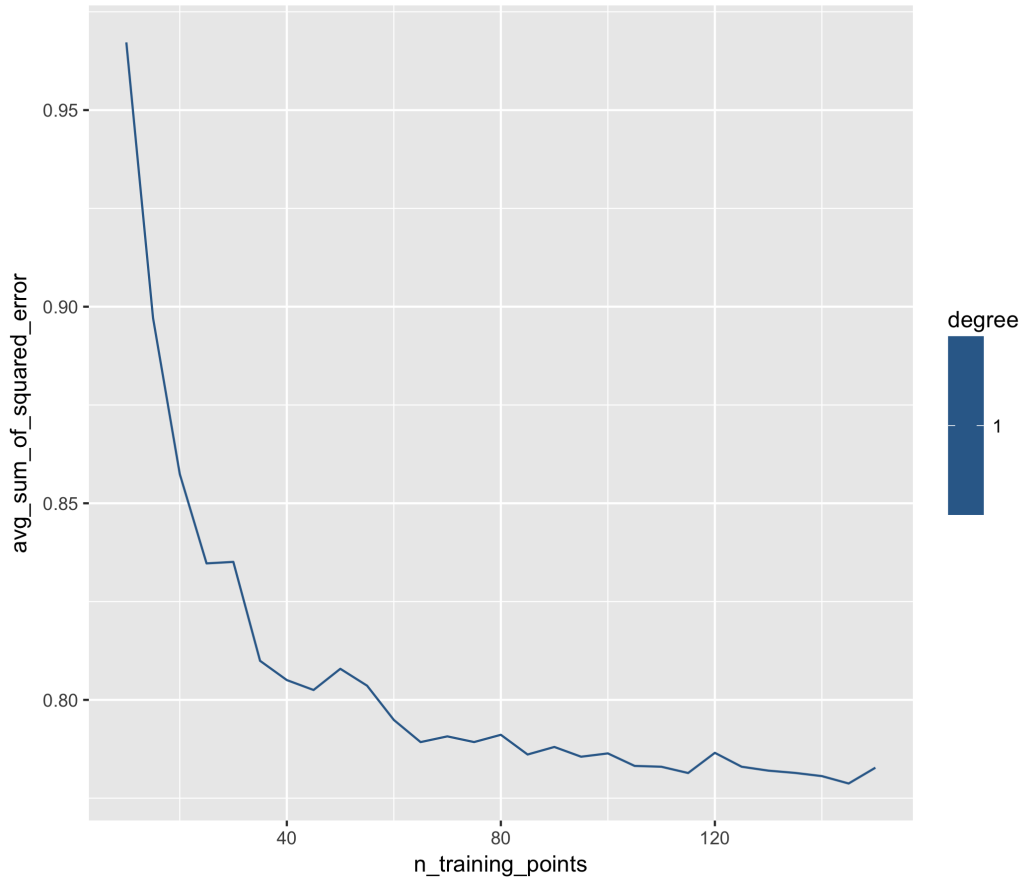
Increasing the amount of data used to fit reduces the variance:



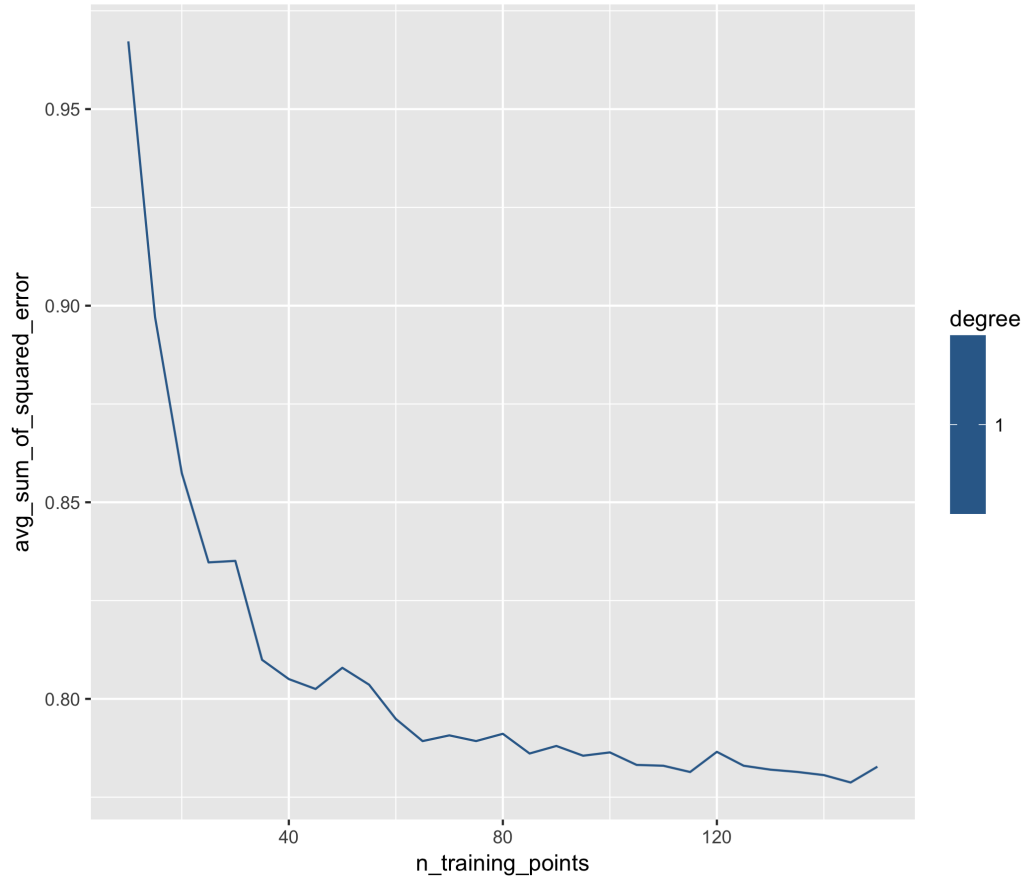
Decreasing the amount of data used to fit increases the variance:



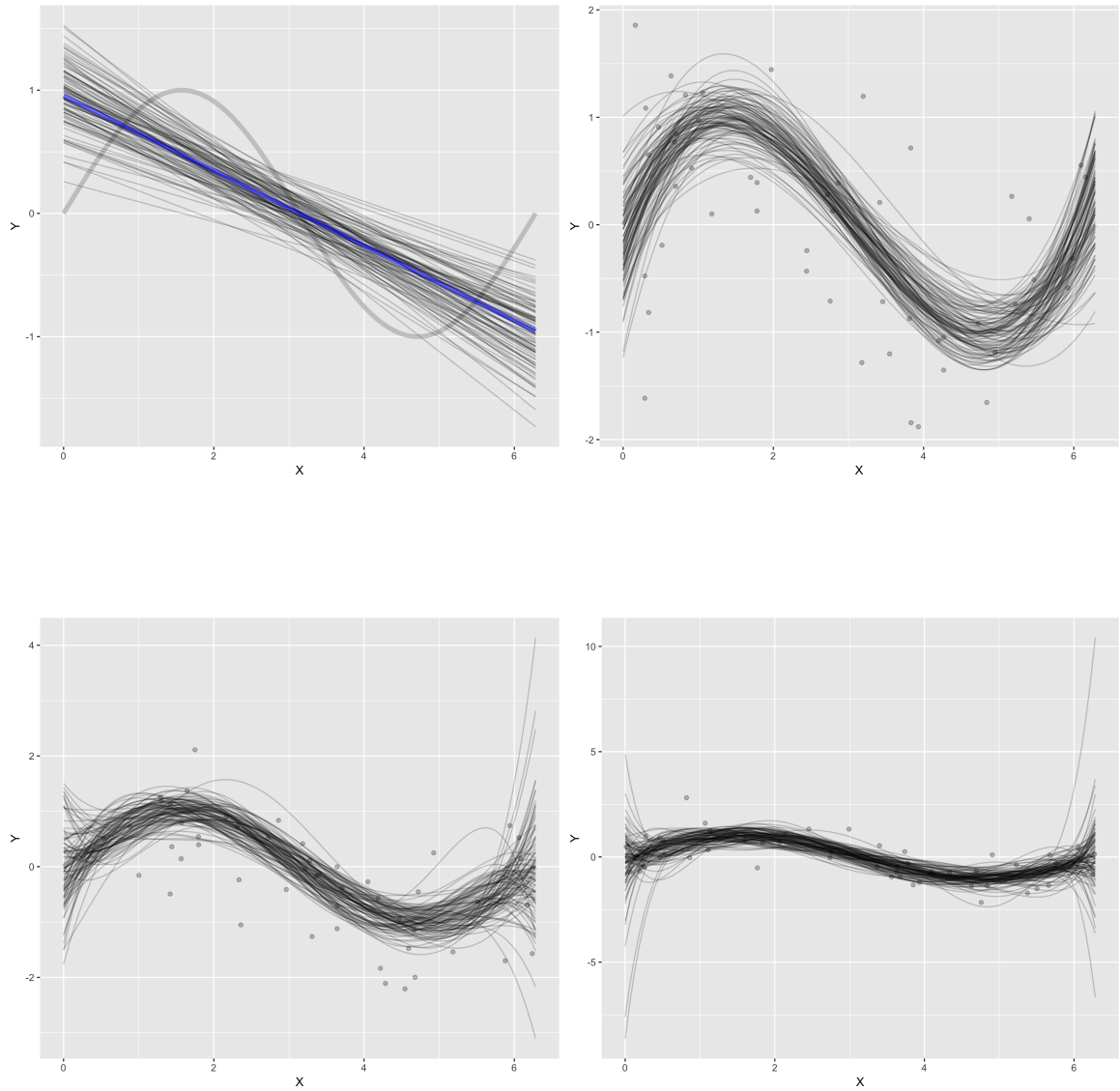
As the amount of data available for training is increased, estimates of expected error tend to approach a limit after which they cannot be decreased:



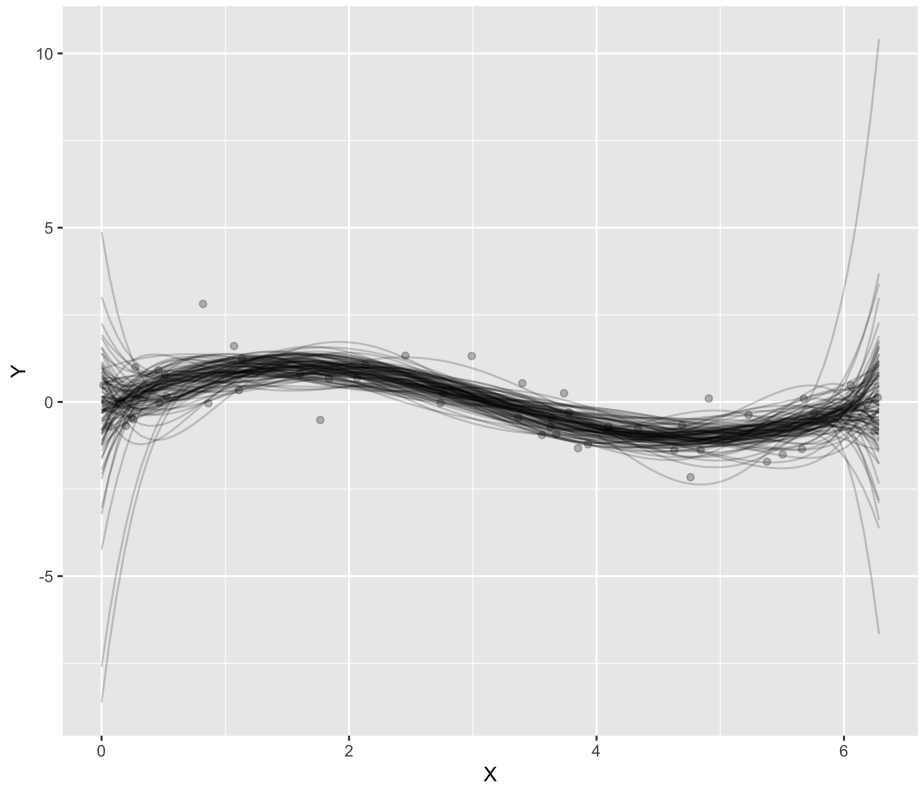
The error remaining after the estimate stabilizes are due to the other error components: the irreducible error and the model bias.



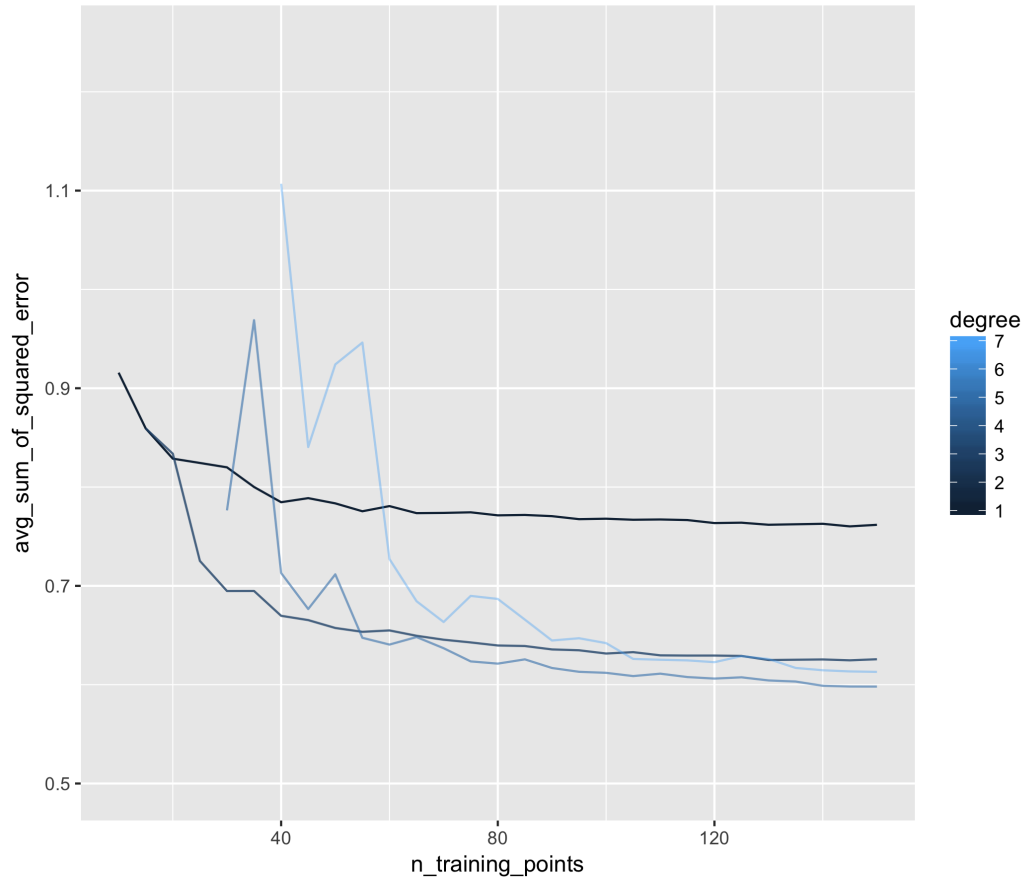
Increasing the complexity of the model also tends to increase the model variance:



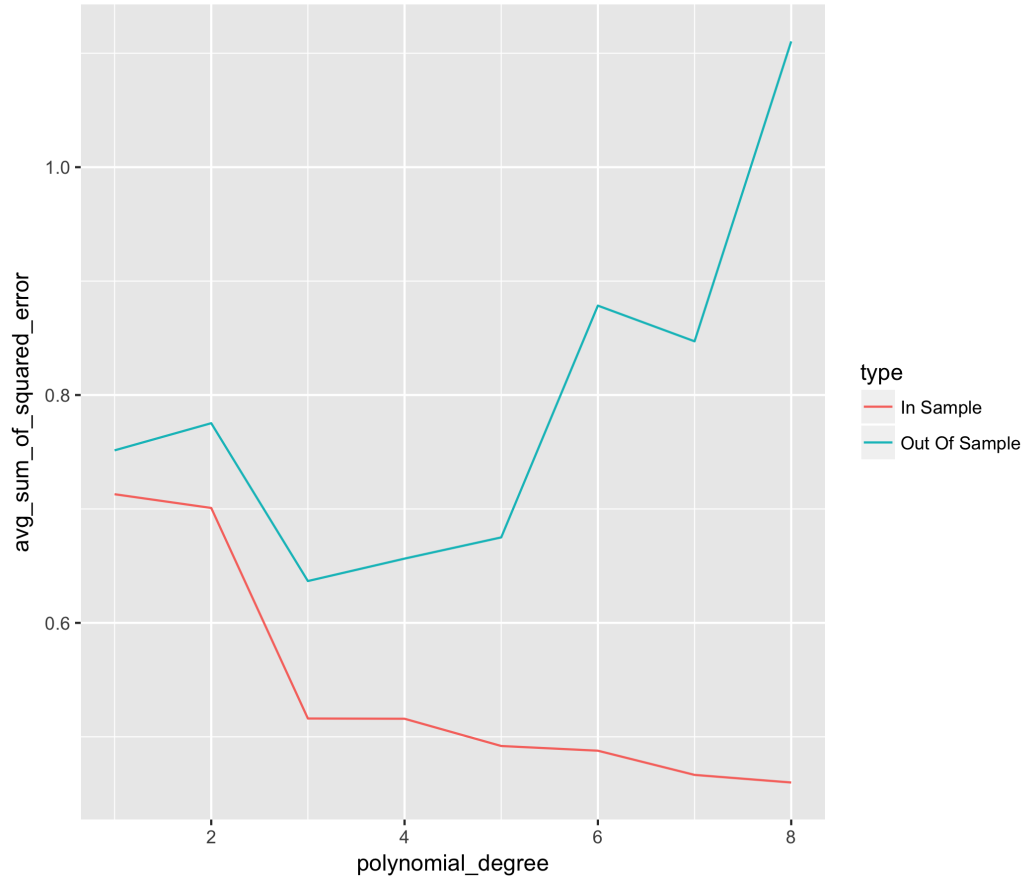
Near the boundaries of the data the pointwise model variance can be extremely large, and dominate the signal:



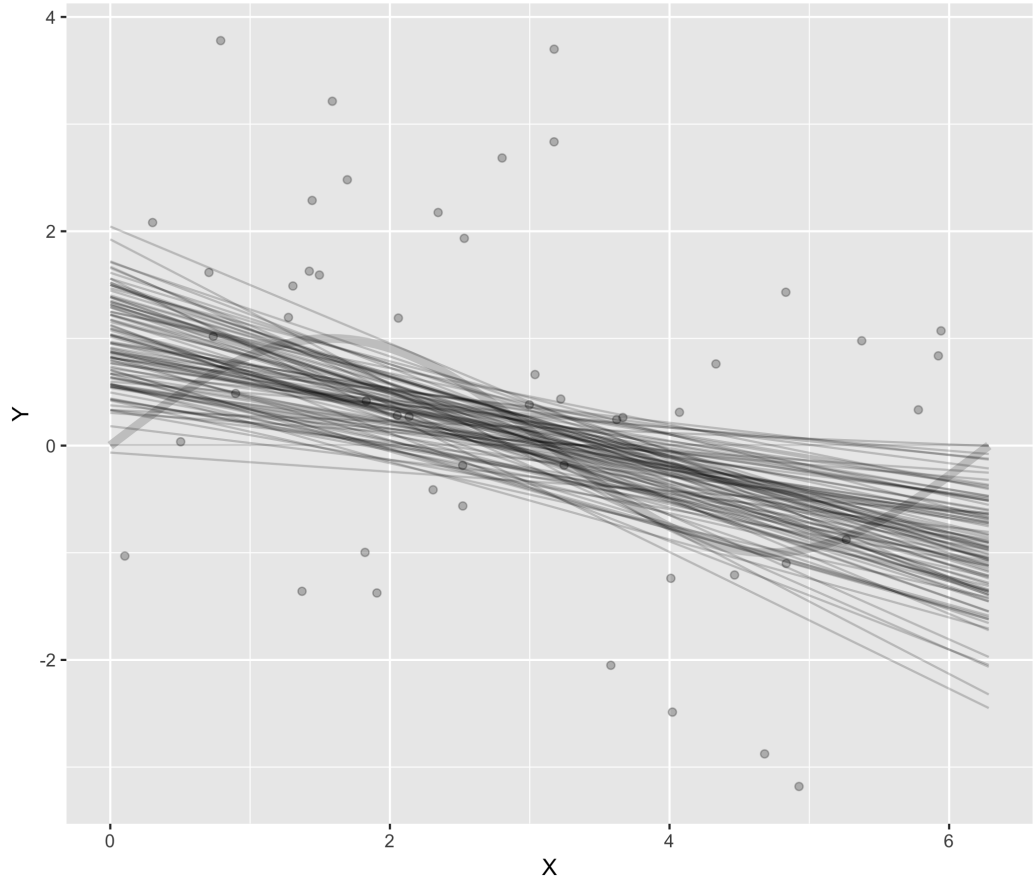
Increasing the complexity of the model can result in lower bias, and hence a lower asymptotic error rate, but enough data is needed to overcome the additional model variance:



If the complexity is increased too much, the variance can dominate, causing the expected error rate to increase



Finally, increasing the irreducible error rate also increases the model variance:

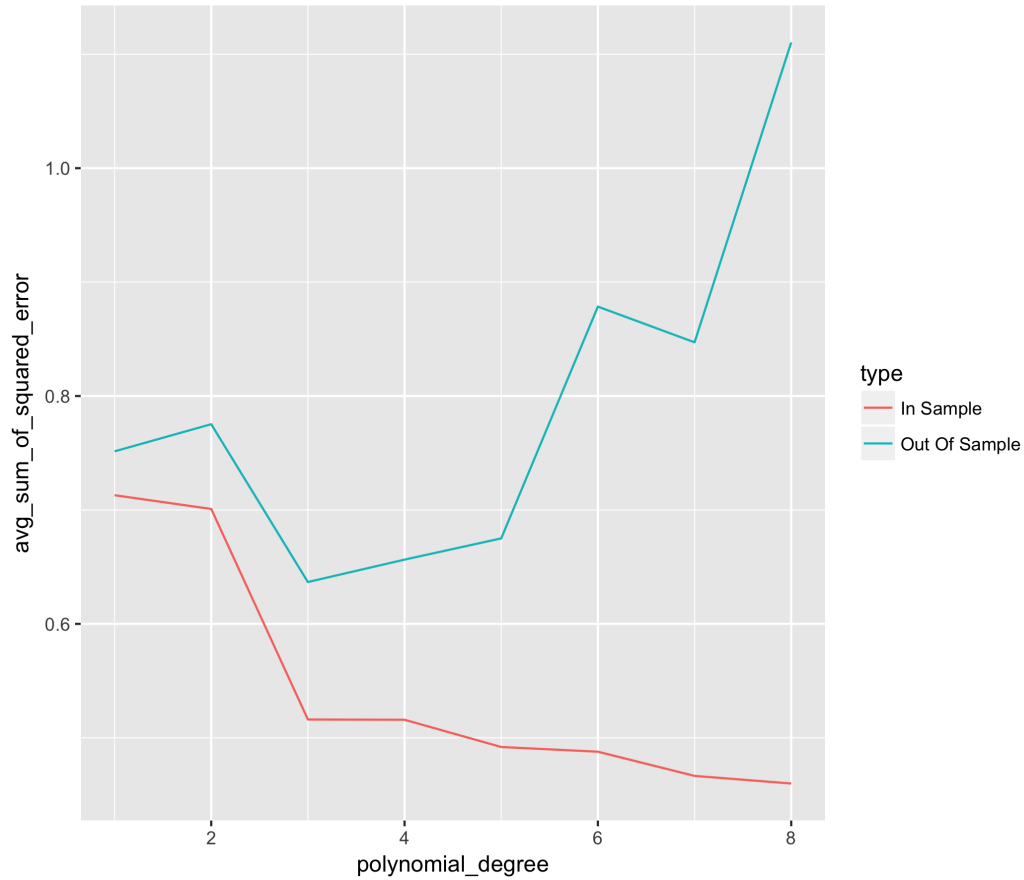


A point derived from this discussion is especially important. The complexity of a model should be a function of the **quantity** and **quality** of the data available.

- **Quantity**: The variance is a function of the number of samples available to train.
- **Quality**: The variance is a function of the irreducible error rate.

The complexity of a model specification should **not** be based on supposed apriori knowledge of the complexity of the signal function, as seductive as this impulse is.

This is demonstrated here:



No polynomial curve can capture the true signal completely, but an ideal fit is closer the higher the degree. None-the-less, the decrease in bias is overcome by the increase in variance.

The Out of Sample Error Rate

The goal in building a predictive model is often to find an appropriate functional form that minimizes the expected error rate:

$$\textbf{ESE}(f) = E\left[(y - f(x))^2\right]$$

In this section, we turn our attention to estimating this quantity.

The first difficulty is that there are different ways to interpret the error, depending on what is or is not considered at random.

We have already discussed the error when randomizing over both a training set and an out of training sample, called the **expected out of sample error**:

$$\textbf{ESE} = E_{X,Y,\mathcal{D}}\left[(y - f(x;\mathcal{D}))^2\right]$$

It is this quantity that we were able to decompose with the **bias-varaince** decomposition.

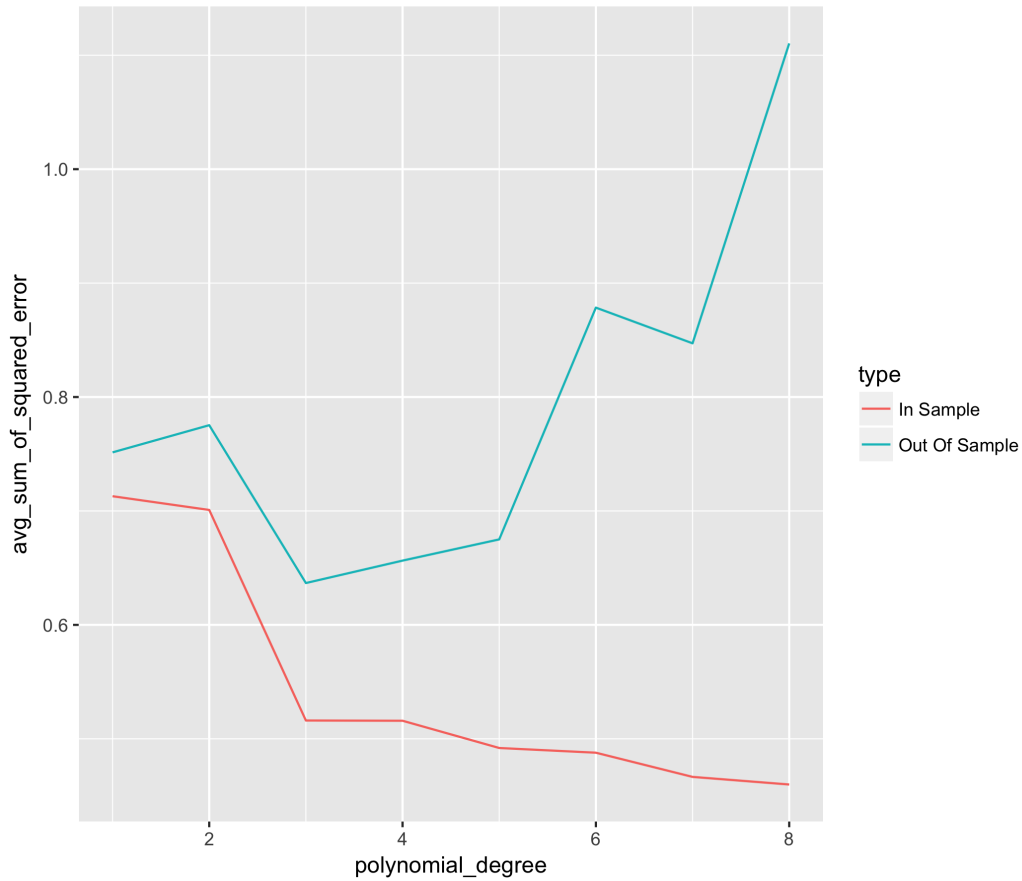
Possibly more relevant, is the error when randomizing over an out of training sample, but for a *fixed* training set:

$$\mathbf{ESE}(\mathcal{T}) = E_{X,Y}\left[(y - f(x;\mathcal{T}))^2\right]$$

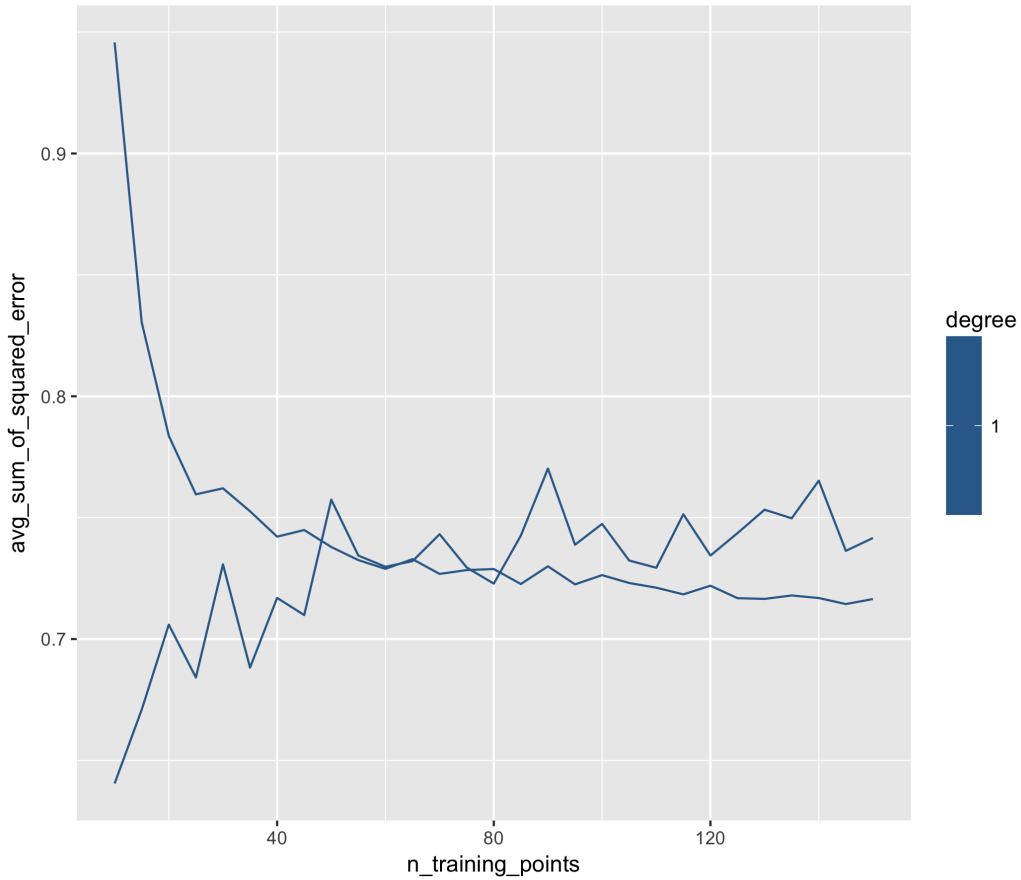
Also convenient is the **expected training error**, where the sample error is computed on the same dataset used to train the model:

$$\textbf{ETE} = E_{\mathcal{T}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{T}}(y - f(x;\mathcal{T}))^2\right]$$

The expected training error is *not* a good estimator of the expected out of sample error, because the modeling algorithm incentives fitting the training data as closely as possible:



The effect is amplified in high model variance situations (small data sets or noisy data):



The bias of the in sample error is so important, it's worth putting some math behind it:

$$\textbf{ESE} = E_{X,Y,\mathcal{T}}\left[(y - f(x;\mathcal{T}))^2\right]$$

The bias of the in sample error is so important, it's worth putting some math behind it:

$$\begin{aligned}\textbf{ESE} &= E_{X,Y,\mathcal{T}}\left[(y - f(x;\mathcal{T}))^2\right] \\ &= E_{\mathcal{D},\mathcal{T}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{D}}(y - f(x;\mathcal{T}))^2\right]\end{aligned}$$

We can replace the expectation of distribution with the expectation of its sample means.

The bias of the in sample error is so important, it's worth putting some math behind it:

$$\begin{aligned}
\textbf{ESE} &= E_{X,Y,\mathcal{T}}\left[(y - f(x;\mathcal{T}))^2\right] \\
&= E_{\mathcal{D},\mathcal{T}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{D}}(y - f(x;\mathcal{T}))^2\right] \\
&\geq E_{\mathcal{D}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{D}}(y - f(x;\mathcal{D}))^2\right]
\end{aligned}$$

This is the definition of $f(X,\mathcal{D})$, it has smaller error than all other possible f when the sample error is evaluated on $\mathcal{D}$.

The bias of the in sample error is so important, it's worth putting some math behind it:

$$\begin{aligned}
\textbf{ESE} &= E_{X,Y,\mathcal{T}}\left[(y - f(x;\mathcal{T}))^2\right] \\
&= E_{\mathcal{D},\mathcal{T}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{D}}(y - f(x;\mathcal{T}))^2\right] \\
&\geq E_{\mathcal{T}}\left[\frac{1}{N}\sum_{x,y\in\mathcal{T}}(y - f(x;\mathcal{T}))^2\right]
\end{aligned}$$

Changing the name of a free variable.

Since the training error is not a good estimate of the out of sample error, we need an alternative way to get at this quantity. There are two major approaches:

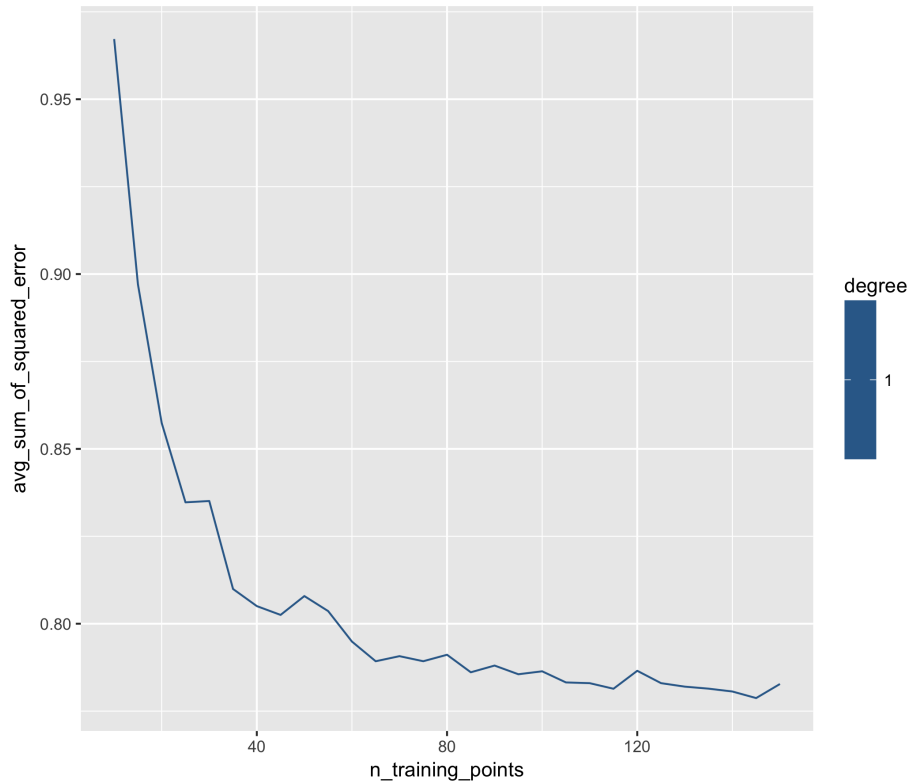
- Using a **hold out set**.
- Using **cross validation**.

If a dataset $\mathcal{H}$ has participated in neither the training of the model, or the decision making process, then:

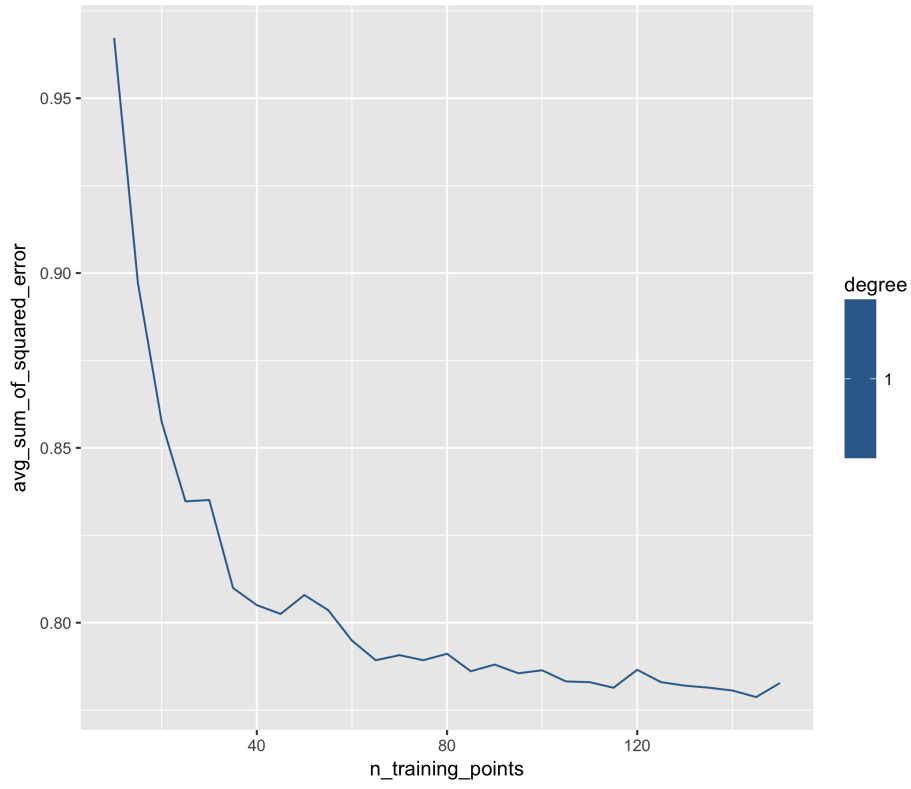
$$\frac{1}{N} \sum_{x,y \in \mathcal{H}} (y - f(x; \mathcal{T}))^2$$

is an unbias estimate of $\mathbf{ESE}(\mathcal{T})$ (the expected out of sample error with a fixed training set).

Holding out data is *not* costless, as there is less data available for training and a smaller data set will increase the variance of the estimated model:



How costly holding out data is depends on the shape of the learning curve for the model being estimated.



This shape is affected by:

- The overall level of noise in the data (irreducible error).
- The complexity of the model being estimated.

Noisier data and more complex models may be more adversely affected by holding out data.

In **cross validation** the data is available is partitioned into a disjoint union of *folds*:

$$\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_k$$

For each i, the out of fold data is formed by removing the fold $\mathcal{T}_i$ from the training data set:

$$\hat{\mathcal{T}}_i = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_{i-1} \cup \mathcal{T}_{i+1} \cup \cdots \cup \mathcal{T}_k$$

The cross validation estimate of the out of sample error is:

$$\frac{1}{k}\sum_{i=1}^{k}\frac{1}{\#\mathcal{T}_i}\sum_{x,y\in\mathcal{T}_i}(y-f(x,\hat{\mathcal{T}}_i))^2$$

Since we are randomizing over a training set, this is actually an estimate of **ESE**.

That is, while the hold out sample error is an estimate of:

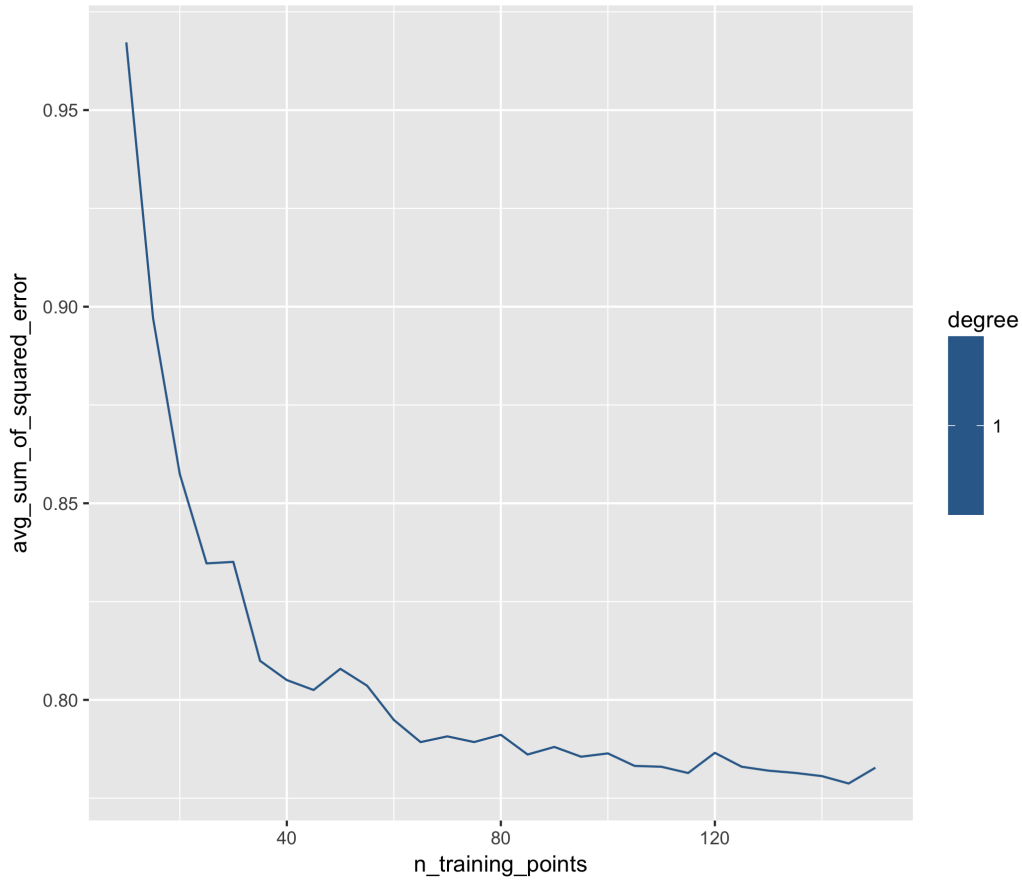
$$\textbf{ESE}(\mathcal{T}) = E_{X,Y}\left[(y - f(x;\mathcal{T}))^2\right]$$

The cross validation estimate is of:

$$\textbf{ESE} = E_{X,Y,\mathcal{D}}\left[(y - f(x;D))^2\right]$$

The number of folds to use is yet another example of balance and compromise in statistical modeling.

If a small k is chosen, much data is held out from each training set, depending on how much total data is available, this can increase the variance of the estimated model:



At the other extreme, the $k = N$ case is called **leave one out cross validation**. In this case the models trained on the complementary sets are highly correlated, as the various training sets differ at only one point:

$$f(x, \hat{\mathcal{T}}_i) \approx f(x, \hat{\mathcal{T}}_j)$$

This makes the cross validation estimator relatively insensitive to model variance, as the training data is not sufficiently averaged out.
Said another way, if the cross validation estimate is repeated over various *full* training sets, *this estimate* will have high variance.