

Supervised Learning (Part II)

DS-6030 | Spring 2026

Invalid Date

Table of contents

1	Bias-Variance Trade-off	2
1.1	Data Generating Functions	2
1.2	One Realization	2
1.3	A second realization	3
1.4	Bias, Variance, and Mean Squared Error (MSE)	4
1.4.1	Distribution of $\hat{\theta}$	4
1.4.2	Some properties of an estimator	5
1.5	Estimating the Bias, Variance, and Mean Squared Error (MSE)	6
1.5.1	Simulation	7
1.5.2	Observations	8
1.5.3	Bias, Variance, and MSE at a single input	8
1.5.4	Integrated MSE	9
1.6	What does it all mean	9
2	Training Data Size	12
2.1	Data Generating Process	12
2.2	Model	12
2.2.1	Model Tuning	13
2.2.2	Final Predictive Model	13
2.2.3	Predictive Evaluation	13
2.2.4	Results	15
2.2.5	How good does cross-validation do at estimating the MSE?	16
3	Test Data Size	16

1 Bias-Variance Trade-off

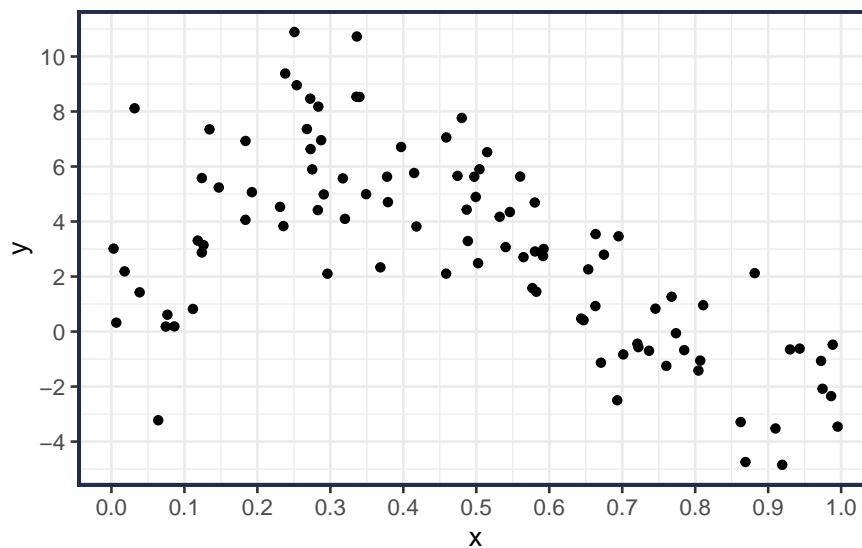
This section explores the bias-variance trade-off for the examples we covered last class. This involves examining the theoretical properties of an estimator.

1.1 Data Generating Functions

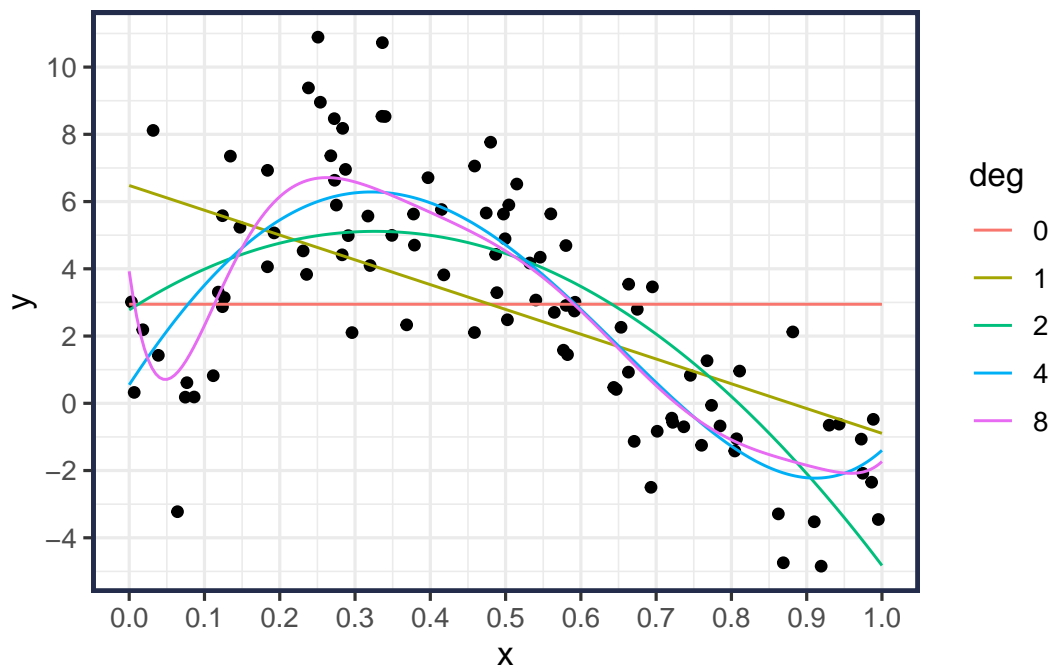
Here we set the data generation functions. $X \sim U[0, 1]$ and $f(x) = 1 + 2x + 5 \sin(5x)$ and $y(x) = f(x) + \epsilon$, where $\epsilon \stackrel{\text{iid}}{\sim} N(0, 2)$.

1.2 One Realization

Last class, we explored one realization from this system.

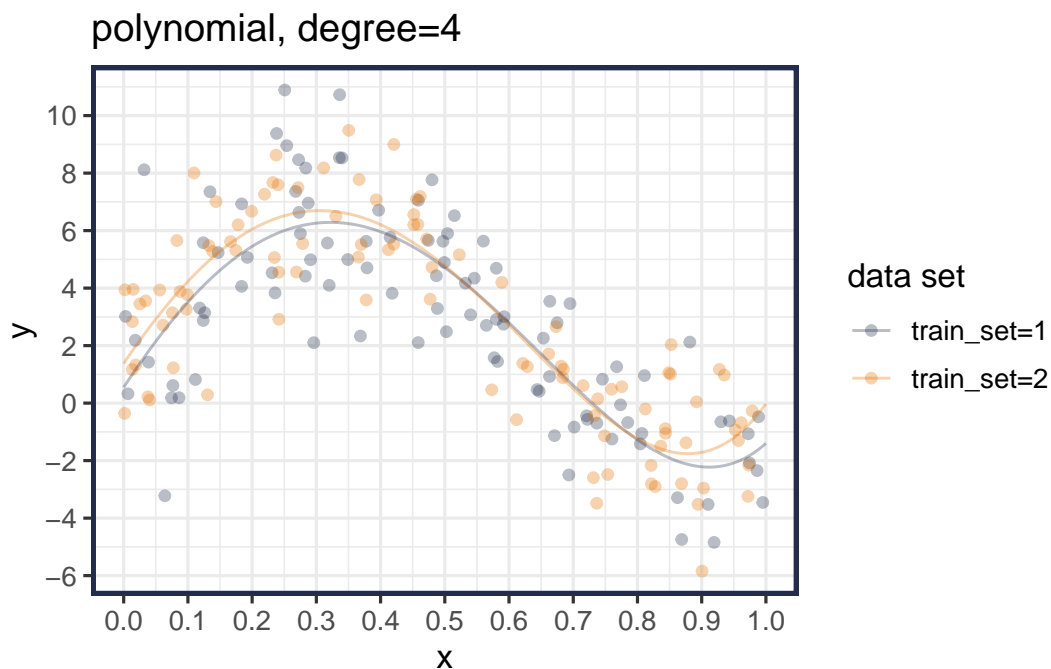


And then fit several polynomial regression models. Recall by polynomial regression I mean using a predictor function $\hat{y}(x) = f(x, d) = \sum_{j=0}^d x^j \hat{\beta}_j$, where $d \in \{0, 1, \dots\}$ is the degree.



1.3 A second realization

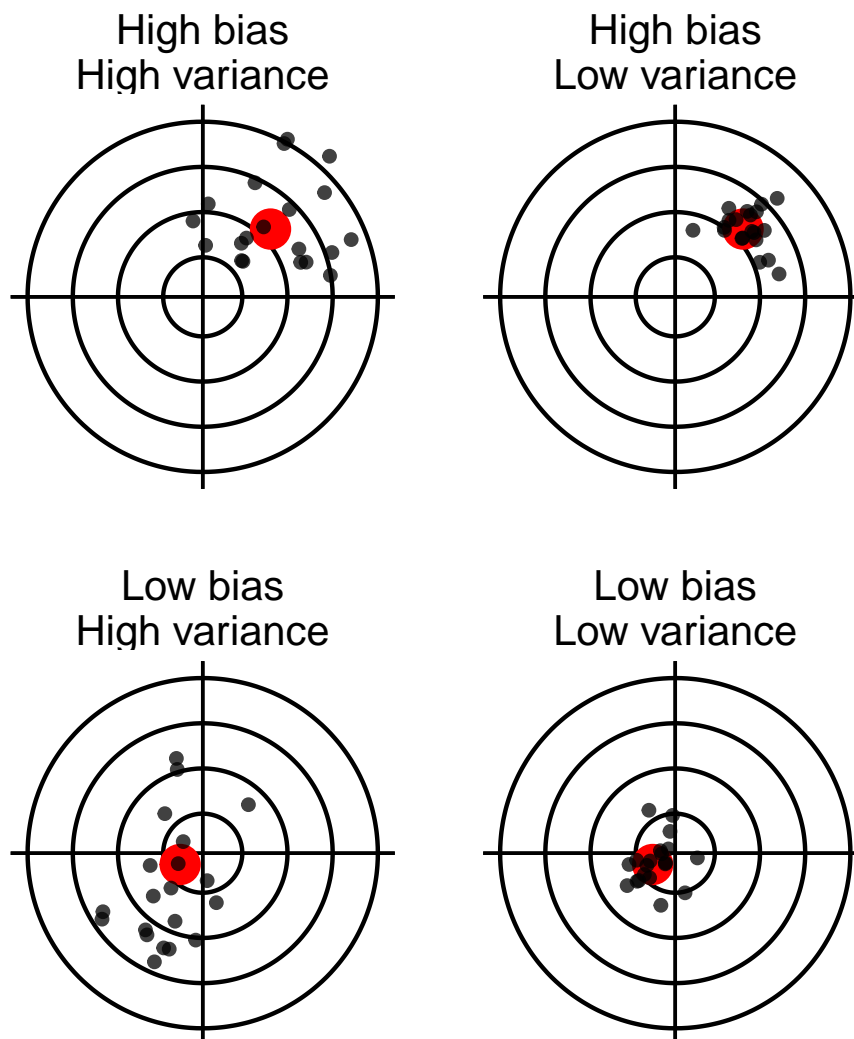
Suppose we drew another training set (using same distributions and sample size n):



- We get another fitted curve using the new training data.
- While the two curves are visually similar, they are not identical.
- If we took more training samples, we would get more fitted curves
- What we want to study in this section is the likelihood that we will happen to get a *good* fit given a single training data set.

1.4 Bias, Variance, and Mean Squared Error (MSE)

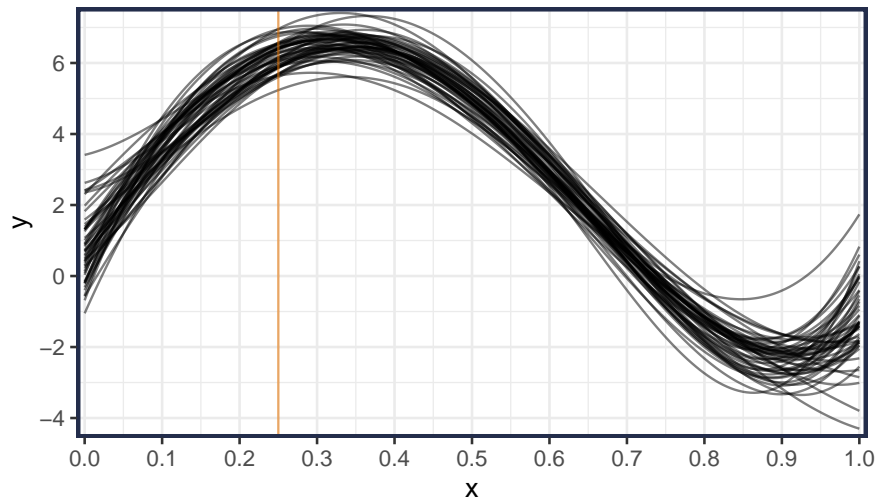
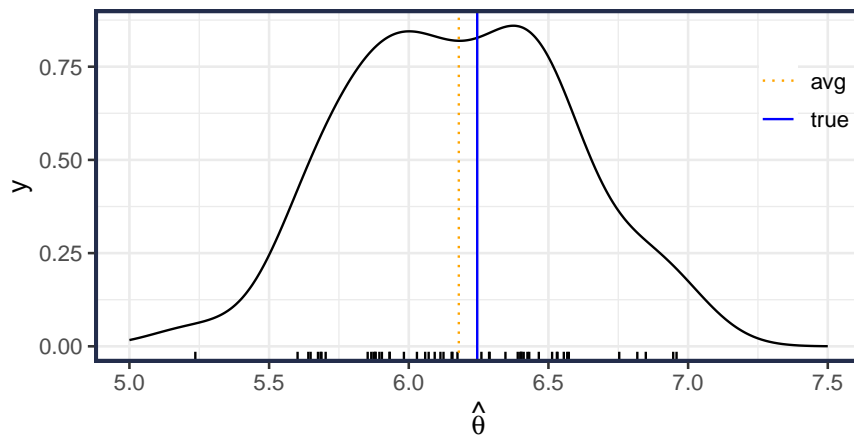
- The statistical properties of an estimator can help us understand its potential performance
- Let $D = [(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)]$ be *training data*
- Let $\hat{\theta} = \hat{\theta}(D)$ be the estimated value *calculated from the training data* D
 - E.g. $\theta = f(x)$, $\hat{\theta} = \hat{f}(x | D)$
 - $\hat{\theta}$ is a *random variable* because we are treating the data as random; it has a distribution.



1.4.1 Distribution of $\hat{\theta}$

- Consider the distribution of $\hat{\theta} = \hat{f}_{\text{poly}}(0.25, d = 4)$.
 - This is the distribution of the fit at $x = 0.25$ from a polynomial (of degree 4) using different *training sets*
- I generated 50 different training data sets (each with $n = 100$), fit a polynomial (deg=4) model to each data set, and recorded the estimate at $x = 0.25$.

Predictions from 50 different training sets

distribution of $\hat{f}(x = 0.25, d = 4)$ Optimal/True $f(x)$ given by blue line, sample mean given by orange

1.4.2 Some properties of an estimator

- **Bias** of an estimator is defined as *the expected value of the estimate minus the true value* or:

$$E_D[\hat{\theta}] - \theta$$

- **Variance** of an estimator is defined as *the variance of the estimate* or:

$$V_D[\hat{\theta}] = E_D[\hat{\theta}^2] - E_D[\hat{\theta}]^2$$

- **MSE** of an estimator is defined as *the expected squared distance of the estimate from the true value* or:

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= E_D[(\hat{\theta} - \theta)^2] \\ &= V_D[\hat{\theta} - \theta] + E_D[\hat{\theta} - \theta]^2 \\ &= V_D[\hat{\theta}] + E_D[\hat{\theta} - \theta]^2 \end{aligned}$$

- Estimators are often evaluated based on MSE, being unbiased, and/or having minimum variance (out of all unbiased estimators)
- These properties are based on the *distribution of an estimate*.
 - Once we observe the training data, the resulting estimate may be great or horrible.
 - However these theoretical properties provide insight into what we can expect and how much confidence we can have in the estimates.

1.5 Estimating the Bias, Variance, and Mean Squared Error (MSE)

- Last class, we examined the Risk/EPE (e.g., MSE) *conditioning on the training data* (See Section 6.2.1)
- Now we will relax this and bring in the uncertainty in the training data D

Under a squared error loss function $L(Y, f(X)) = (Y - f(X))^2$, the *overall* EPE (or EPE before we see any training data) at a particular $X = x$ is

$$\begin{aligned} \text{MSE}_x(f) &= \mathbb{E}_{D|Y|X}[(Y - \hat{f}_D(x))^2 | X = x] \\ &= \mathbb{V}[Y | X = x] + \mathbb{V}[\hat{f}_D(x) | X = x] + \left(\mathbb{E}[\hat{f}_D(x) | X = x] - f(x)\right)^2 \\ &= \text{irreducible error} + \text{model variance} + \text{model squared bias} \end{aligned}$$

where D is the training data, f is the true model, and $\hat{f}_D(x)$ is the prediction at $X = x$ estimated from the training data D .

Note

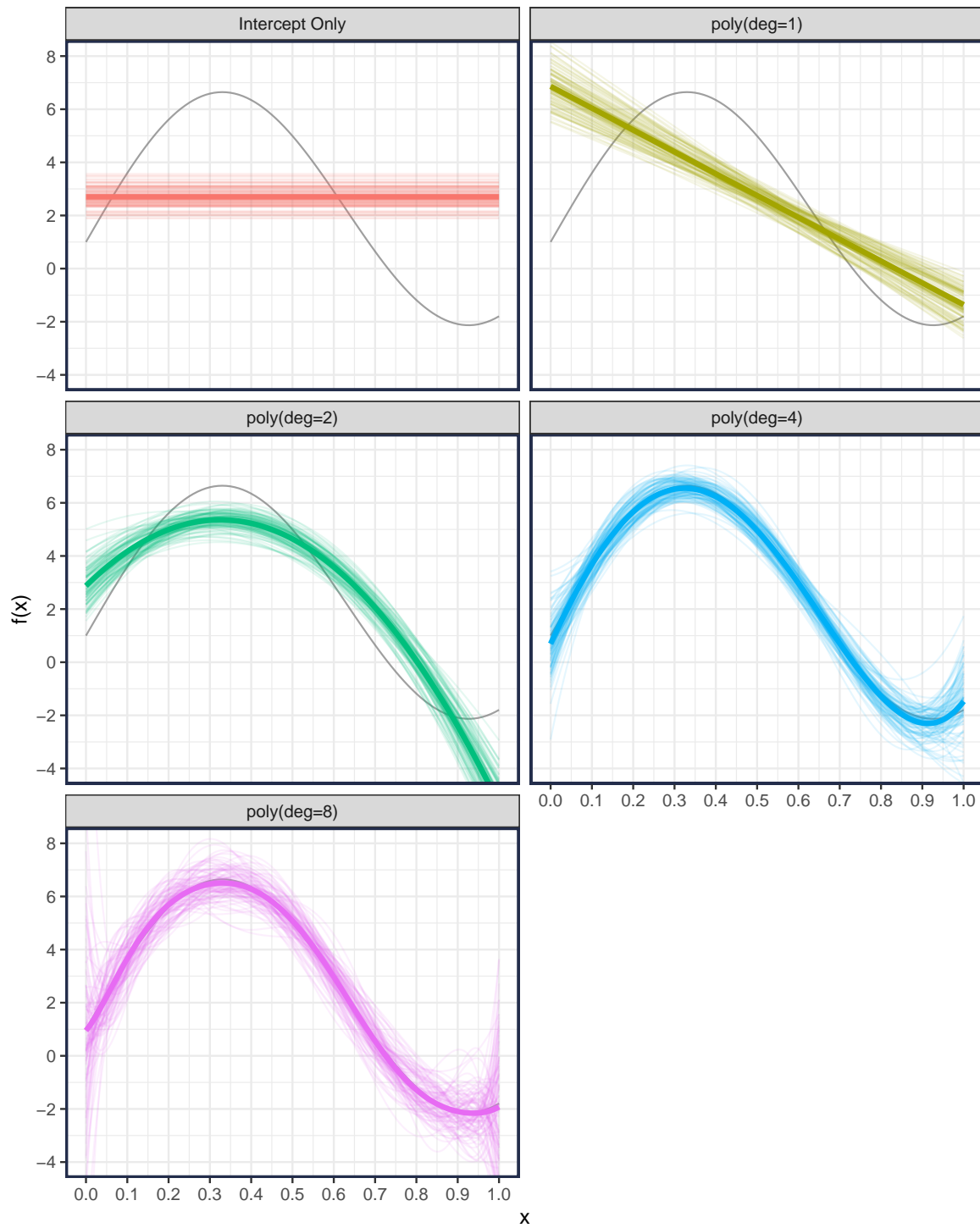
$$\begin{aligned} \text{MSE}_x(f) &= \mathbb{E}_{D|Y|X}[(Y - \hat{f}_D(x))^2 | X = x] \\ &= \mathbb{E}_{D|Y|X}[(Y - f(x) + f(x) - \hat{f}_D(x))^2 | X = x] \\ &= \mathbb{E}_{D|Y|X}[(Y - f(x))^2] + \mathbb{E}_{D|Y|X}[(f(x) - \hat{f}_D(x))^2] + \mathbb{E}_{D|Y|X}[2(Y - f(x))(f(x) - \hat{f}_D(x))] \\ &= \mathbb{V}[Y | X = x] + \mathbb{E}_{D|Y|X}[(f(x) - \hat{f}_D(x))^2] + 0 \\ &= \mathbb{V}[Y | X = x] + \mathbb{V}_{D|Y|X}(\hat{f}_D(x)) + \mathbb{E}_{D|Y|X}(f(x) - \hat{f}_D(x))^2 \end{aligned}$$

- We can estimate the model variance and bias with simulation
 - Generate new data $D_m = \{(Y_i, X_i)\}_{i=1}^n$ for simulations $m = 1, 2, \dots, M$ (use the same sample size n)
 - Fit the models with data D_m getting $\hat{f}_{D_m}(\cdot)$
 - Now we can estimate the items of interest:

$$\begin{aligned} \mathbb{E}[\hat{f}_D(x)] &\approx \bar{f}(x) = \frac{1}{M} \sum_{m=1}^M \hat{f}_{D_m}(x) \\ \mathbb{V}[\hat{f}_D(x)] &\approx s_f^2(x) = \frac{1}{M-1} \sum_{m=1}^M (\hat{f}_{D_m}(x) - \bar{f}(x))^2 \end{aligned}$$

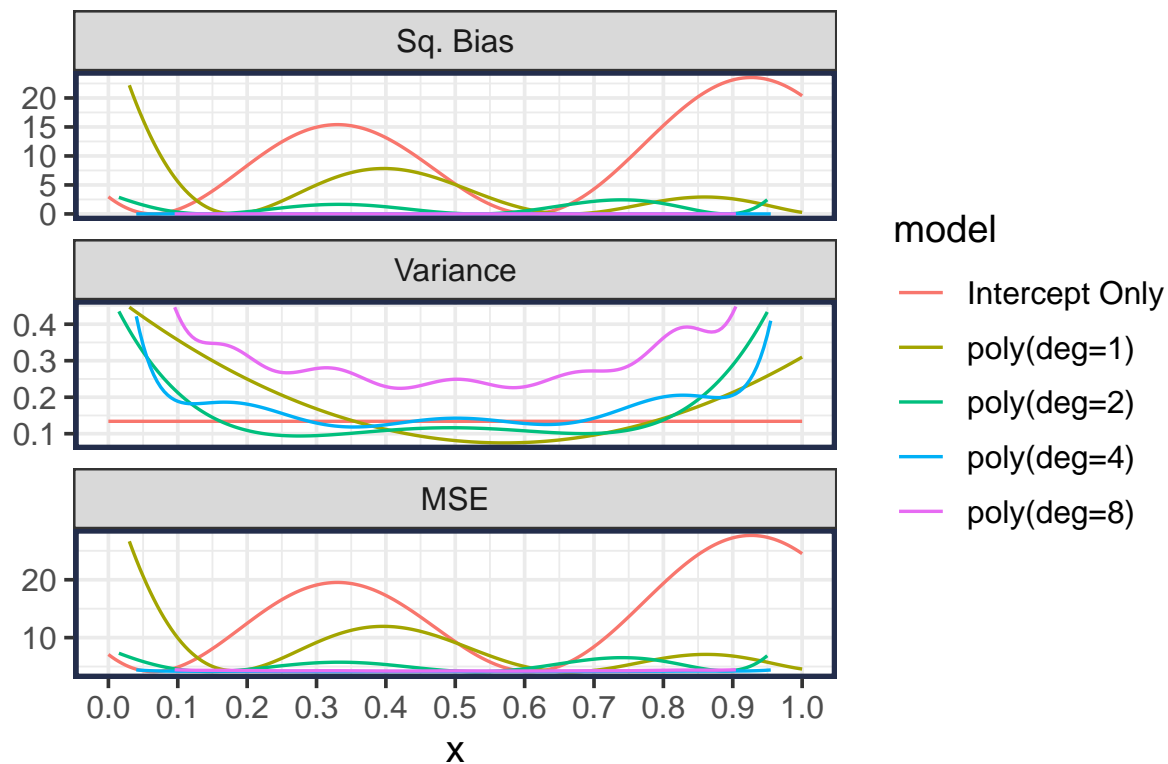
1.5.1 Simulation

I ran 2000 simulations to generate $\{\hat{f}_m(x, \text{deg} = d) : d \in \{0, 1, 2, 4, 8\}, m \in \{1, 2, \dots, 2000\}\}$.



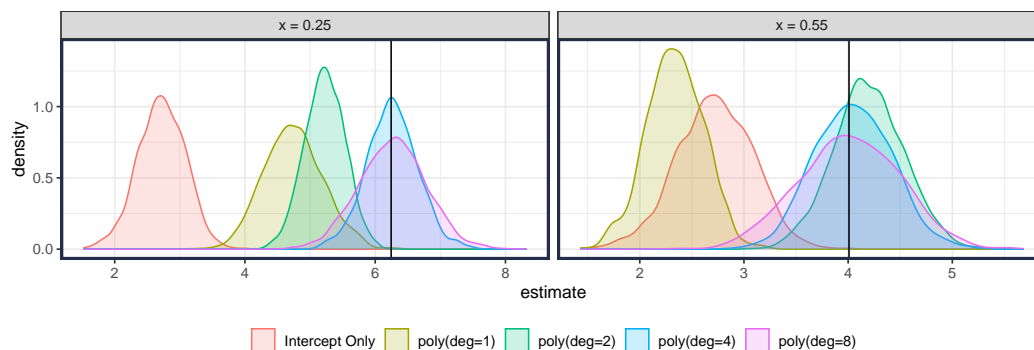
1.5.2 Observations

- This shows the bias and variance of each model.
- You can see that as the flexibility (e.g., degree) of the model increases, the bias decreases but the variance (especially at the edges) increases.
 - The **bias** is the difference between the true regression function (dark gray line) and the model mean (dark colored line).
 - The **variation** is seen in the width of the transparent curves, one for each simulation.



1.5.3 Bias, Variance, and MSE at a single input

- Notice that model variance and model bias vary over x .
- To help see what is going on, we now look at the distributions at $x = 0.25$ and $x = 0.55$.



1.5.4 Integrated MSE

The above analysis examines the $\text{MSE}_x(f)$ over a set of x 's. However, in a real setting, the overall test error will be based on *all* of the actual test X values. So we are usually more interested in the *integrated* MSE:

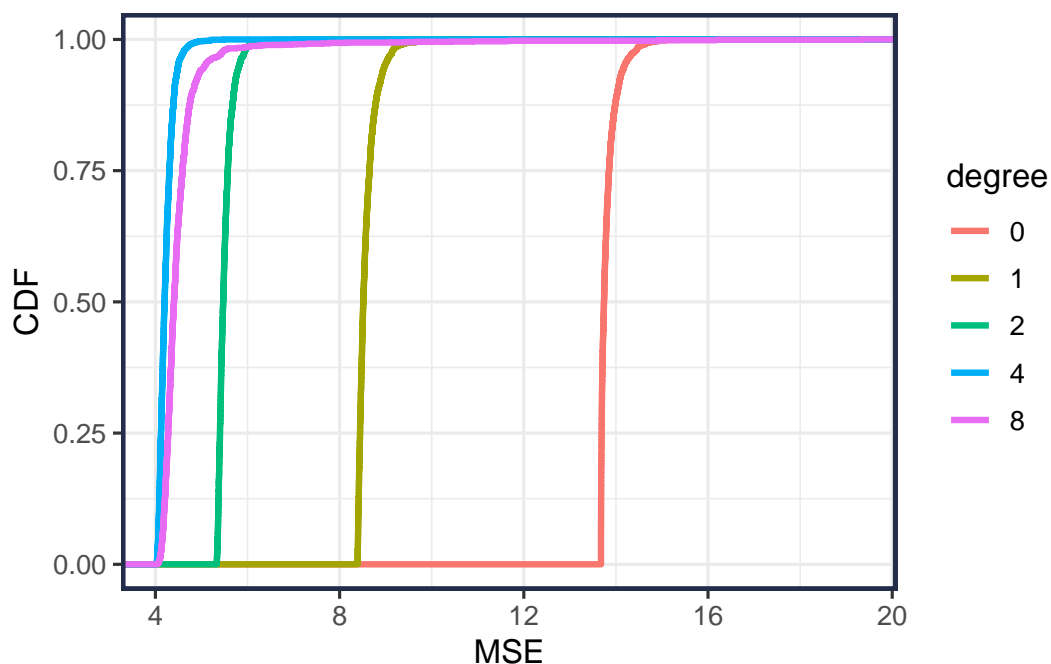
$$\begin{aligned}\text{MSE}(f) &= E_{D^{YX}}[(Y - \hat{f}_D(X))^2] \\ &= E_X[\text{MSE}_X(f)] \\ &= \int \text{MSE}_x(f) \Pr(dx)\end{aligned}$$

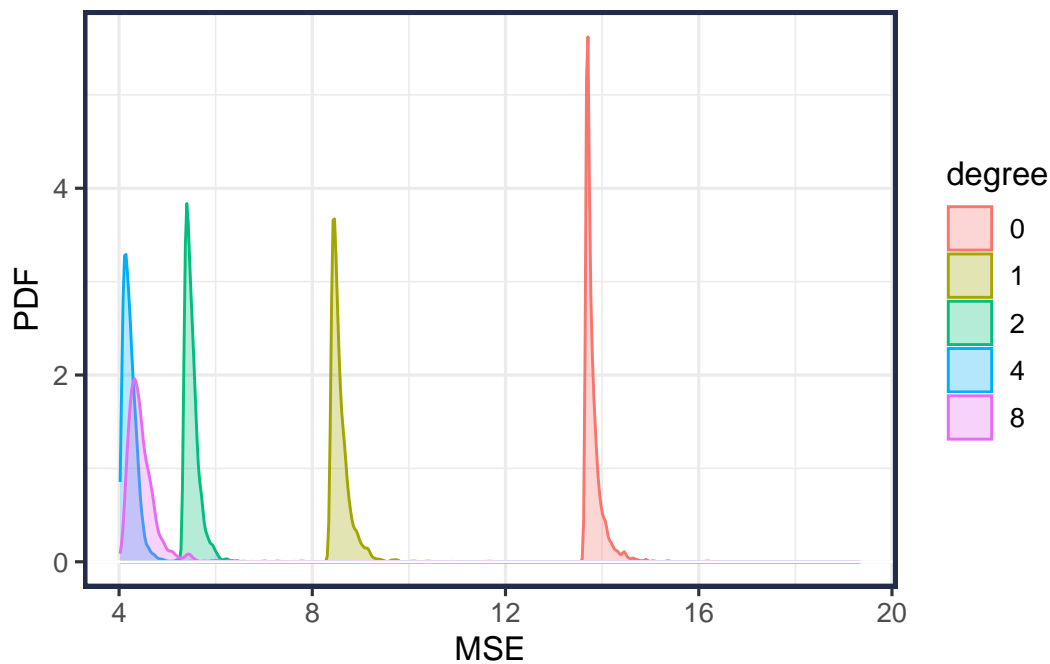
deg	bias.sq	var	mse
0	9.68	0.13	13.81
1	4.39	0.19	8.58
2	1.33	0.18	5.51
4	0.01	0.22	4.23
8	0.00	0.53	4.53

1.6 What does it all mean

The main point is that we desire to find a predictive model that has just the right *flexibility*. This will depend on both the true flexibility of the data as well as the sample size. A model that is too complex will have low bias, but potentially high variance. A model that is not complex enough will have high bias, but lower variance.

In a real setting, we will only observe one training data (a single curve) and will have to decide the optimal flexibility. The following plot shows the estimated distribution of MSE values.





While it's possible that we could just happen to get a particular training data realization that favors a model other than the globally optimal model, this is unlikely for the bad models. However, it is not uncommon for “close” models.

Below is a table of the number of simulations that each model had the best MSE:

degree	n
0	0
1	0
2	0
3	397
4	1006
5	486
6	75
7	25
8	4
9	6
10	1

- While the polynomial of degree=4 is most often best model (under the squared error loss), under some training data sets, the other degree models predict better.
- **Conclusion 1:** In our toy example, a polynomial with degree=4 is the best model, in principal. However, for some data (i.e., some training data sets) the models with degree>4 and degree<4 would predict better.
 - Why do more higher degree models (degree > 4) predict better than lower degree models (degree < 4)?
- **Conclusion 2:** The above analysis is what is meant by the “bias-variance trade-off”.

- In reality, we only get to observe one realization of the training data so we can never actually estimate the bias and variance the way we did above
 - But we can still estimate the Risk (e.g., MSE) by using resampling methods like cross-validation or statistical methods like BIC.
 - More loosely, when people mention bias-variance trade-off they are referring to the principal that the best model is one that has just the right flexibility.
 - If the model is too complex, it is unlikely to produce a good estimate (across the entire range of inputs) because it is likely to stray far from the expected mean at certain values.
 - If the model is not complex enough, then it will not track with the expected mean across the range of input values and thus produce poor overall performance.
- **Conclusion 3:** Performance of a model can vary across the input features X .
 - If you are only concerned about performance in a specific range of X , then emphasize these during training (e.g., weight observations close to X more heavily during model estimation).
 - If the test X values are coming from a different distribution than the training X values, then your model may not be optimal.

2 Training Data Size

2.1 Data Generating Process

To help us analyze how the size of the training data set influences predictive performance, we will create some new simulated data.

- There are p multivariate normal predictors

$$X \sim \text{MVN}(0, \Sigma)$$

$$\Sigma_{ij} = \exp(-|i - j|/\rho)$$

where $\rho > 0$ controls the correlation between predictors.

- The outcome is generated as

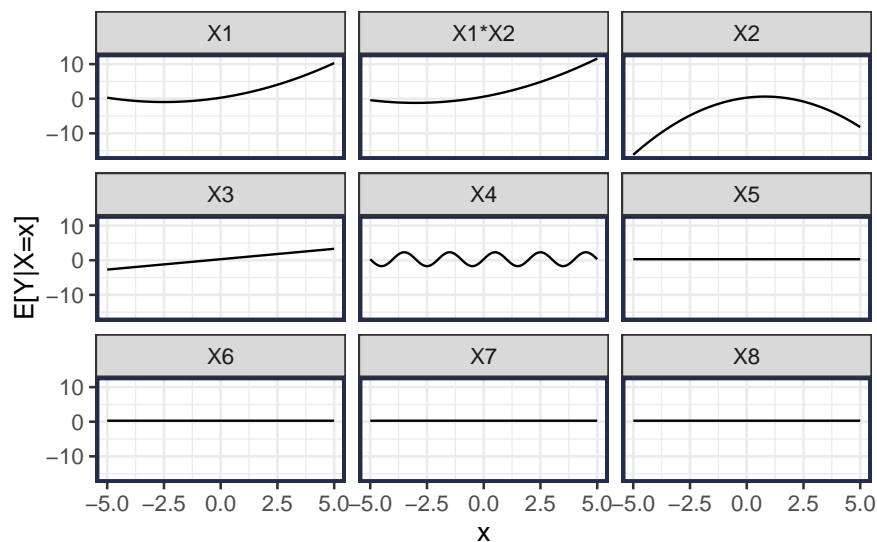
$$y = f(x) + \epsilon$$

where $\epsilon \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma)$.

The mean response (true regression function) is

$$f(x) = 1x_1 + .8x_2 + .6x_3 + .2(x_1^2 - 1) - .5(x_2^2 - 1) + 2\sin(\pi x_4) + .2x_1x_2$$

We are simulating $p = 8$ predictor variables (only 4 influence outcome) and using $\rho = 1$ and $\sigma = 3$.



2.2 Model

Pretending that we don't know the true model, we need determine the model structure to use. For this example, I'll use an *ridge* penalized linear model. The α close to 0 operates similar to the ridge penalty, but can still allow feature selection. The model will use all 8 first order features, all two-way interactions, and 6 degree polynomial expansion for features 1 through 4.

Your Turn #1 : Write down the prediction formula

- Notice that the true model only uses 2nd degree polynomials for features 1 and 2 and a single two-way interaction. Feature 4 would be best captured by a higher degree polynomial. We are specifying a model that can have a much higher effective degrees of freedom due to so many predictors.

2.2.1 Model Tuning

I'm using 10-fold cross-validation to estimate the penalty strength (λ) in the ridge penalty $P_\lambda(\beta) = \lambda \|\beta\|_2^2$.

2.2.2 Final Predictive Model

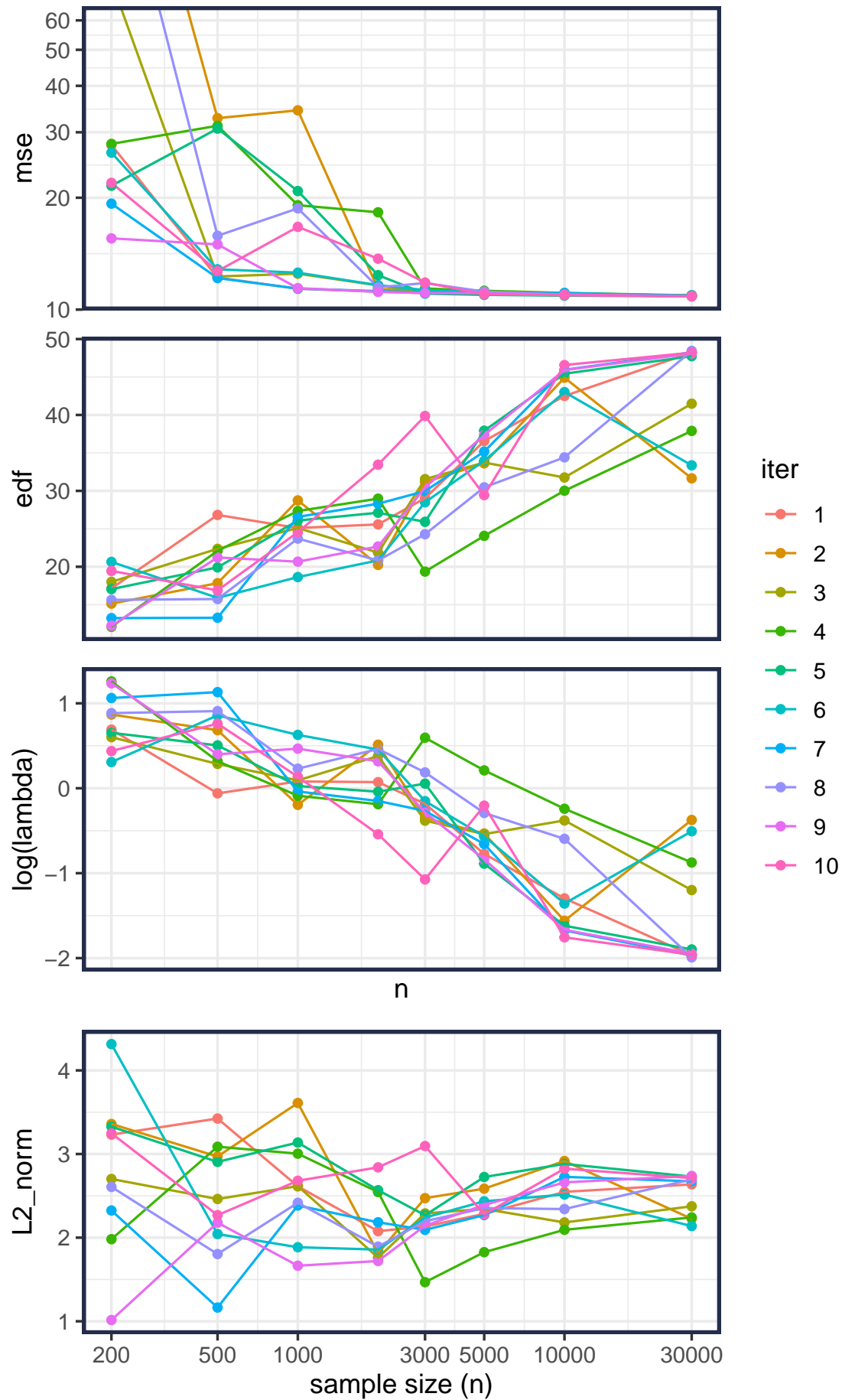
After selecting the optimal tuning parameter ($\hat{\lambda}$) using cross-validation, fit a final model using all `n_train` observations and optimal penalty.

2.2.3 Predictive Evaluation

I generated `n_test = 100,000` observations which are reserved for predictive evaluation.

Then I generated `n_train` observations to get a final prediction model (recording the edf, selected λ , and other model properties), made predictions on the test data, and calculated the MSE. I added additional training data so my total `n_train` was `{200, 500, 1000, 2000, 3000, 5000, 10,000, 30,000}`. This entire process was repeated 10 times.

2.2.4 Results



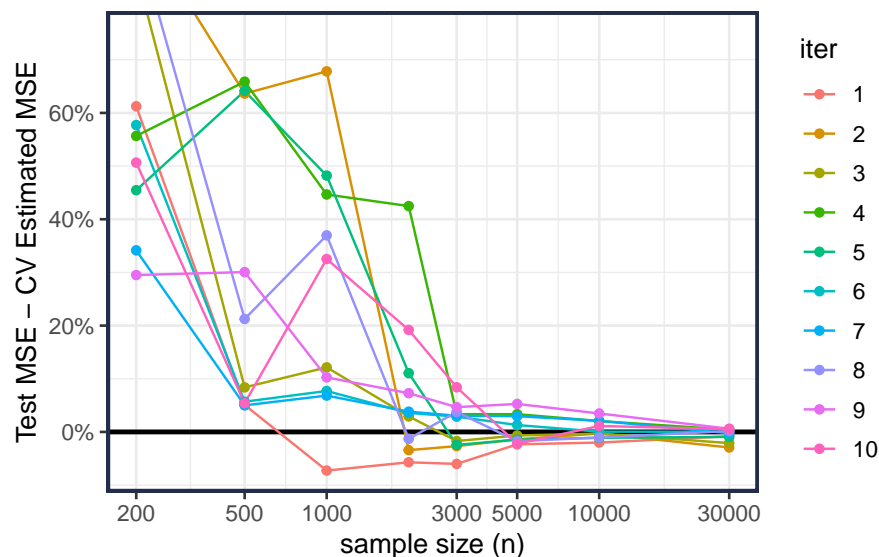
Observations:

- By around $n_{\text{train}} = 3000$, the MSE settles
- As more data is available, the model can support a higher edf (complexity).
- The strength of the optimal ridge penalty (λ) decreases.
- The L2 norm of the estimated (normalized) model parameters seems to stabilize between 2 and 3.
- These observations are only based on 10 replications.

Check out a few individual paths:

- Iteration 2: predictive performance gets worse when $n_{\text{train}} = 1000$.
- Iteration 7: made really good predictions with only $n_{\text{train}} = 500$.

2.2.5 How good does cross-validation do at estimating the MSE?



- With not enough data, cross-validation tends to underestimate the true MSE.
- However this bias decreases as the data size increases.

Your Turn #2 : Cross-Validation

In 10-fold cross-validation, how much data is used to train and test a model?

3 Test Data Size

In the previous analysis, we used a very large $n_{\text{test}} = 100,000$ test set. How many observations do we really need in the test set?

Let's think about what we are doing. We are trying to estimate the Expected Prediction Error (EPE) of a fitted model. That is,

$$\text{EPE}(\hat{f}) = E_{XY}[Y - \hat{f}(X)]$$

The test data is what we use to estimate the EPE:

$$\widehat{\text{EPE}}(\hat{f}) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (y_i - \hat{f}(x_i))^2$$

Remember confidence intervals? And standard errors?

- The *standard error* is the standard deviation of the test statistic. In this case the test statistic is the squared error $\theta = (y_i - \hat{f}(x_i))^2$.
- $\text{SE}(\theta) = \sqrt{\text{V}(\theta)} / \sqrt{n_{\text{test}}}$.
- A 95% confidence interval is approximately estimate ± 2 SE:

