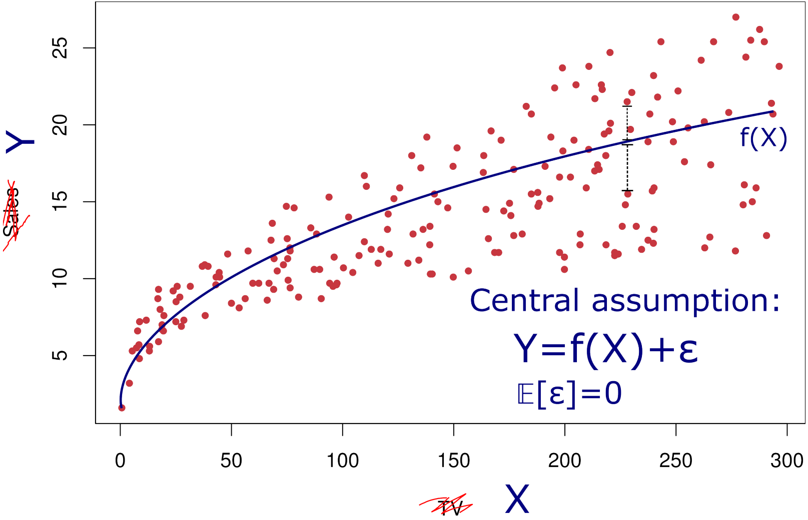
Assessing prediction quality

Remember this?



Remember this?

We assume the setting,

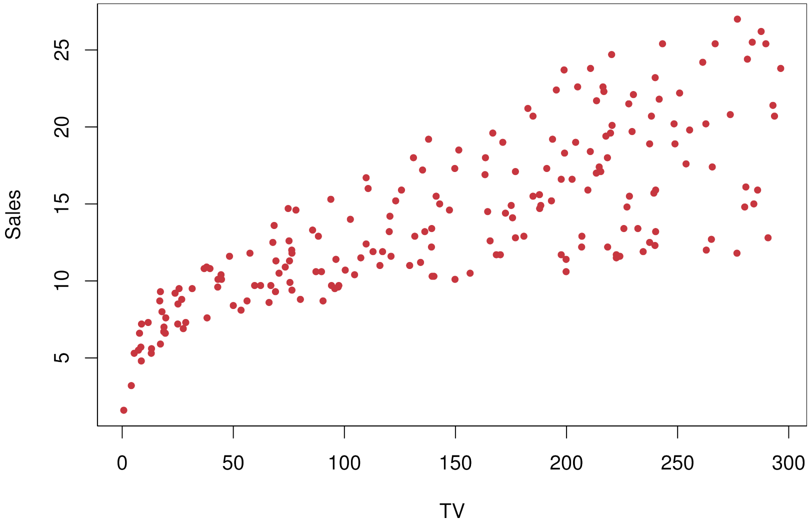
$$Y = f(X_0) + \varepsilon,$$

with ε a mean-zero random variable independent of X_0.

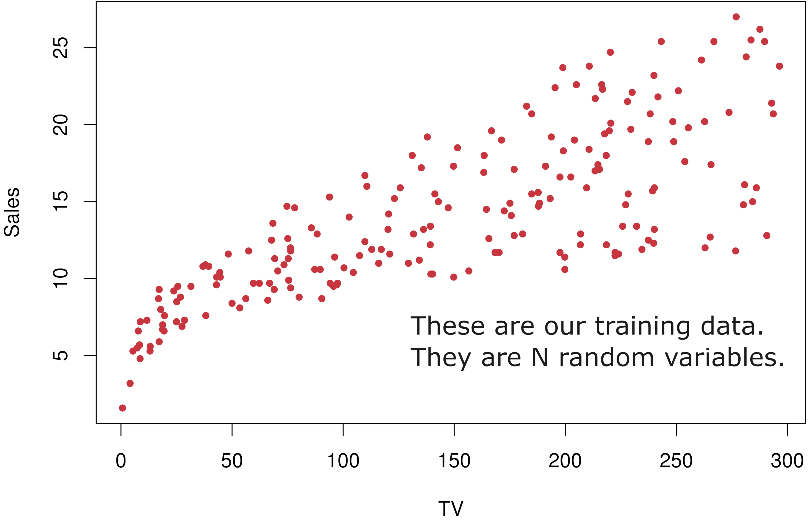
We have discussed methods for fitting a function $\hat{f}$ using a dataset of N observations (X, Y).

- The N observations we refer to as training data
- The algorithm used for fitting the function is called a learner.
- Applying this learner to the training data, we call training.

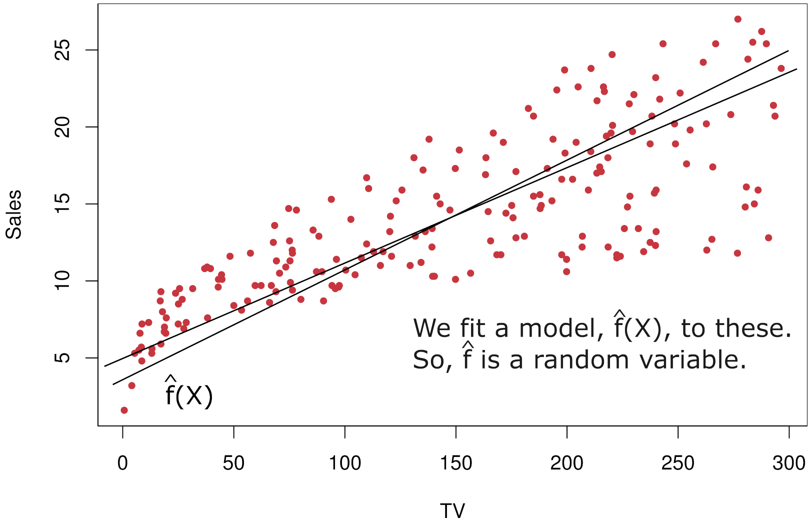
Let's revisit: What is our goal?



Let's revisit: What is our goal?



Let's revisit: What is our goal?



Let's revisit: What is our goal?

What is our goal?

To predict unseen data points: (X_0, Y_0) well!

Q:

Imagine that we have 2 models fit to the data: $\hat{f}_1(X)$ and $\hat{f}_2(X)$. We now get 1 unseen data point (X_0, Y_0).

How do we evaluate which model is best?

Let's revisit: What is our goal?

What is our goal?

To predict unseen data points: (X_0, Y_0) well!

Q:

Imagine that we have 2 models fit to the data: $\hat{f}_1(X)$ and $\hat{f}_2(X)$. We now get n unseen data points (X_i, Y_i) for $i = 1 \ldots n$.

How do we evaluate which model is best?

Let's revisit: What is our goal?

What is our goal?

To predict unseen data points: (X_0, Y_0) well!

In other words:

With access to training data (X, Y), and a chosen model framework, we want our *expected test MSE*,

$$\mathbb{E}[\Big(Y_0 - \hat{f}(X_0)\Big)^2],$$

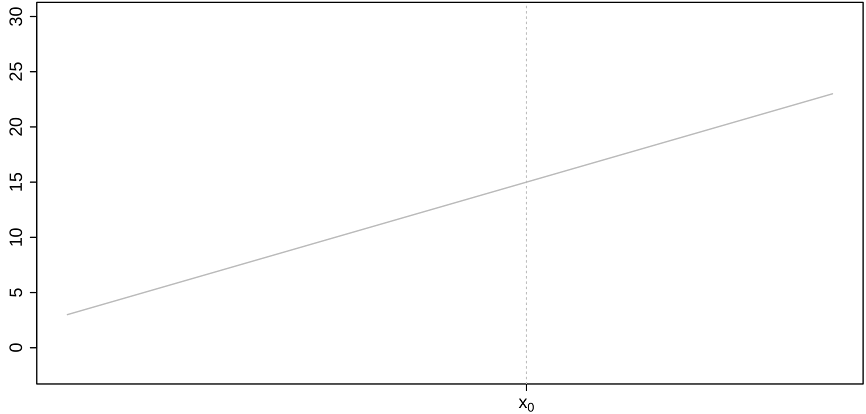
to be as small as possible.

(The unseen test data is what we care about!)

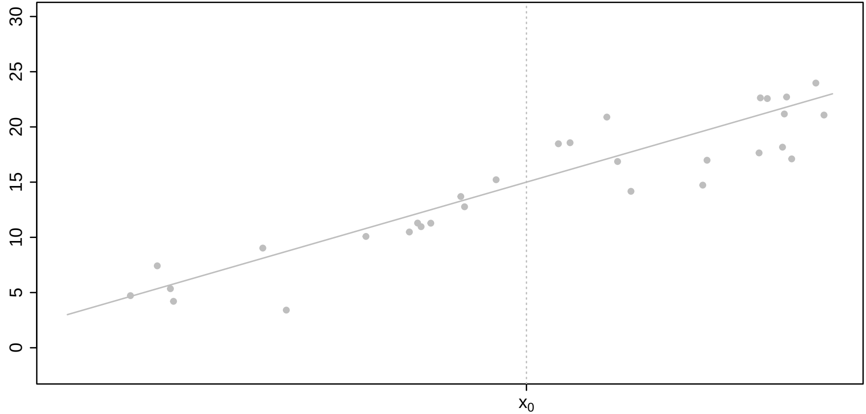
Let us consider now a very general question:

What is the expected squared error that we get from the entire process of first training the learner on some training data and then using it for predicting the response Y_0 *for a new observation* X_0*?*

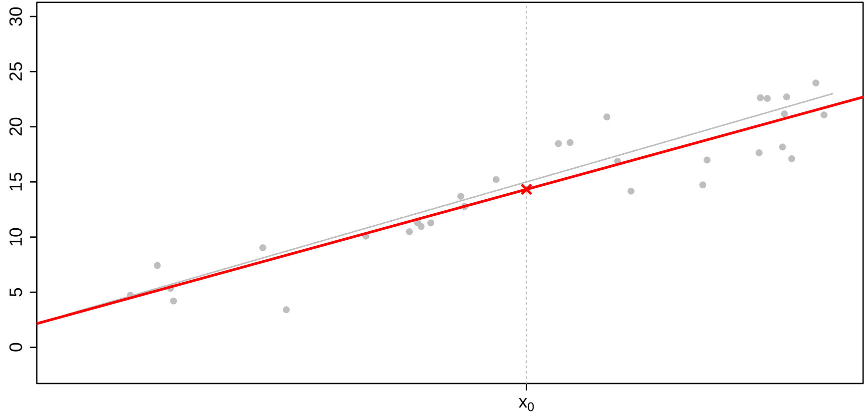
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



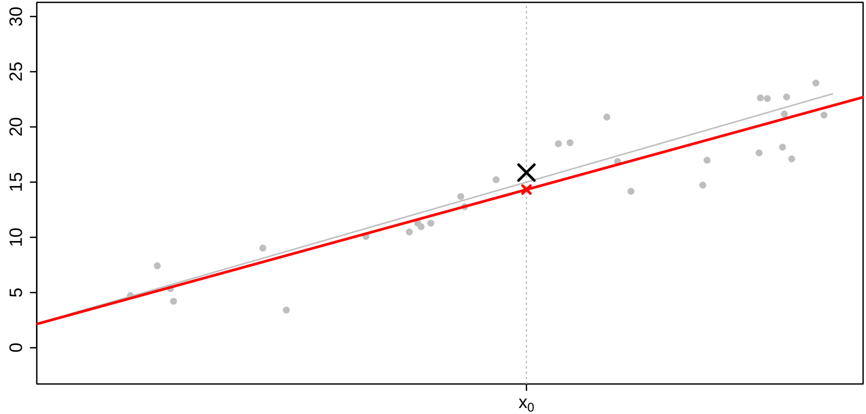
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



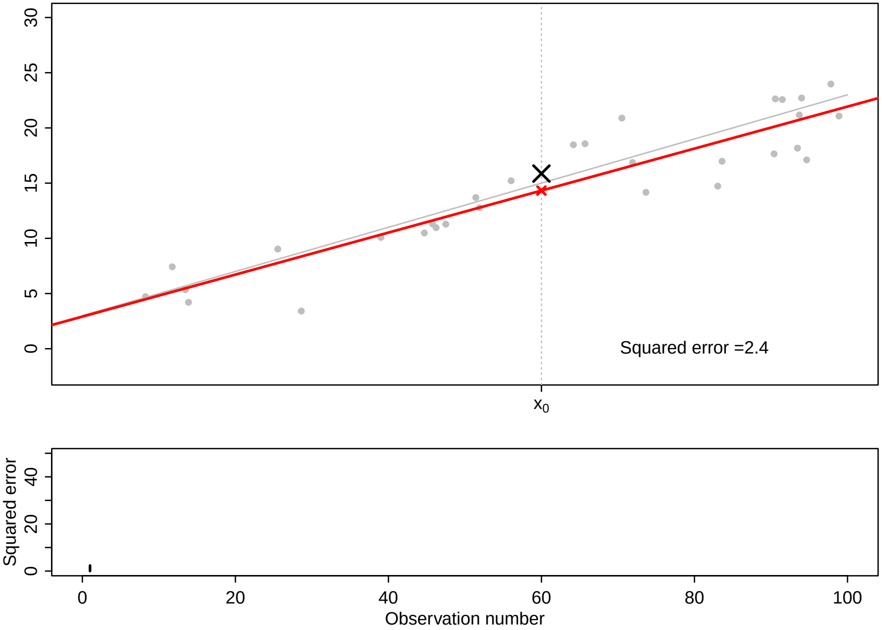
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



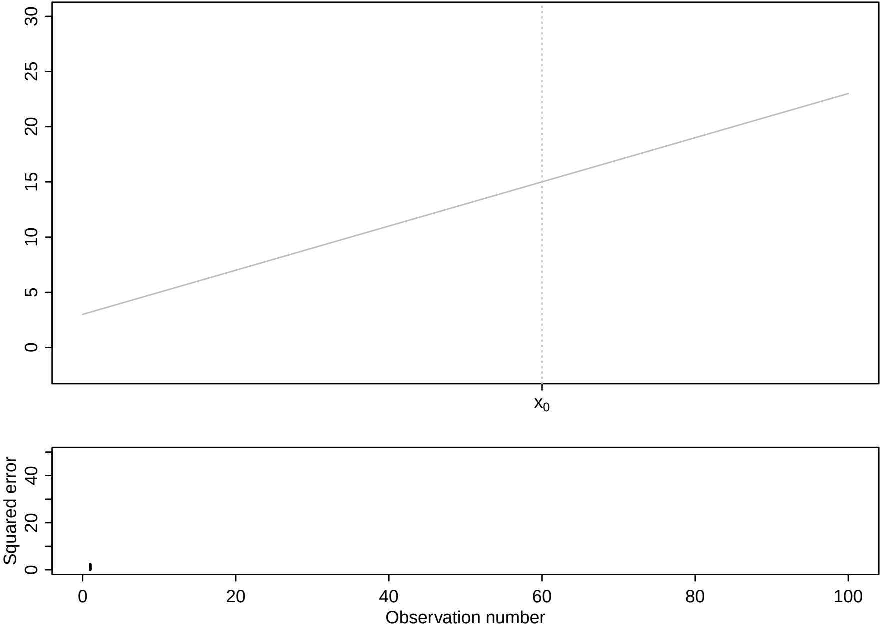
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



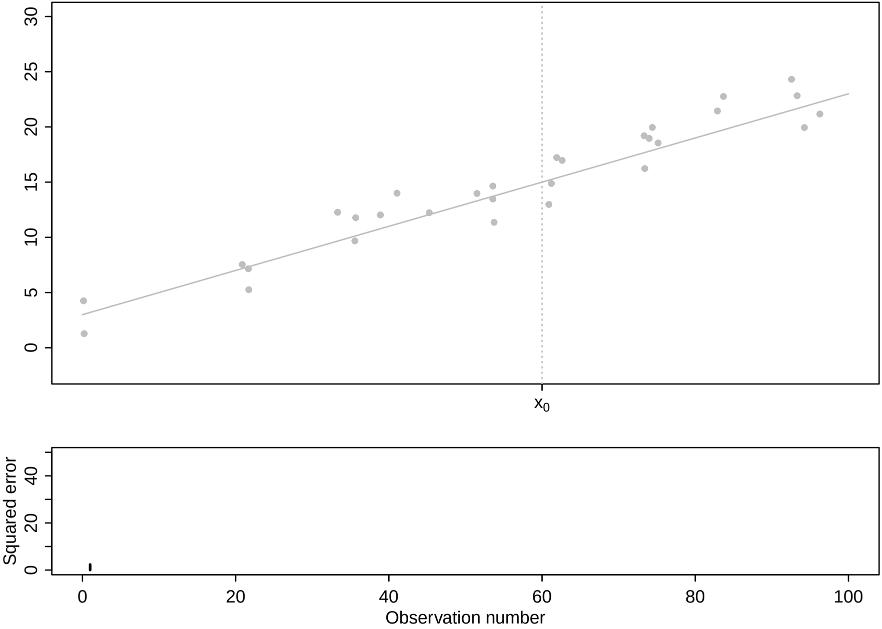
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



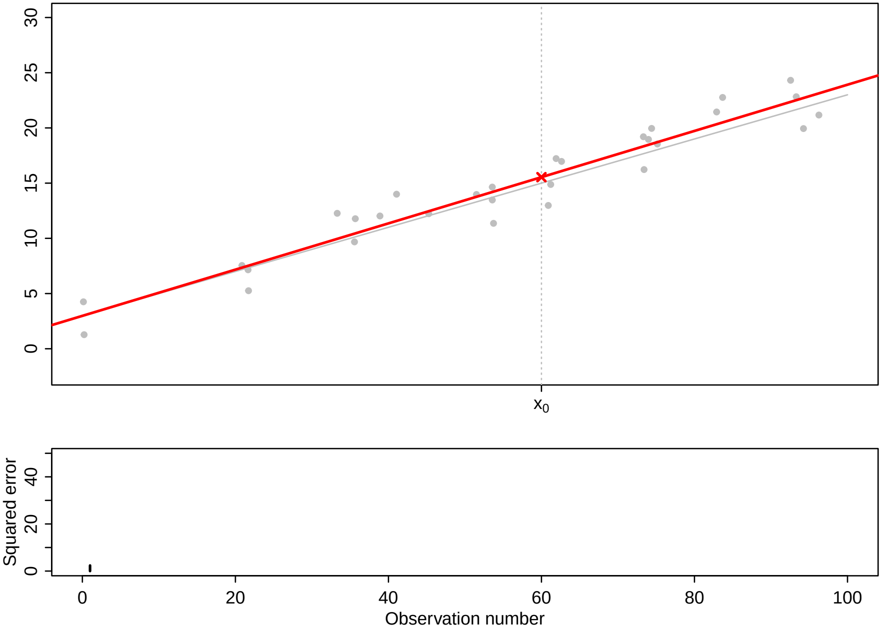
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



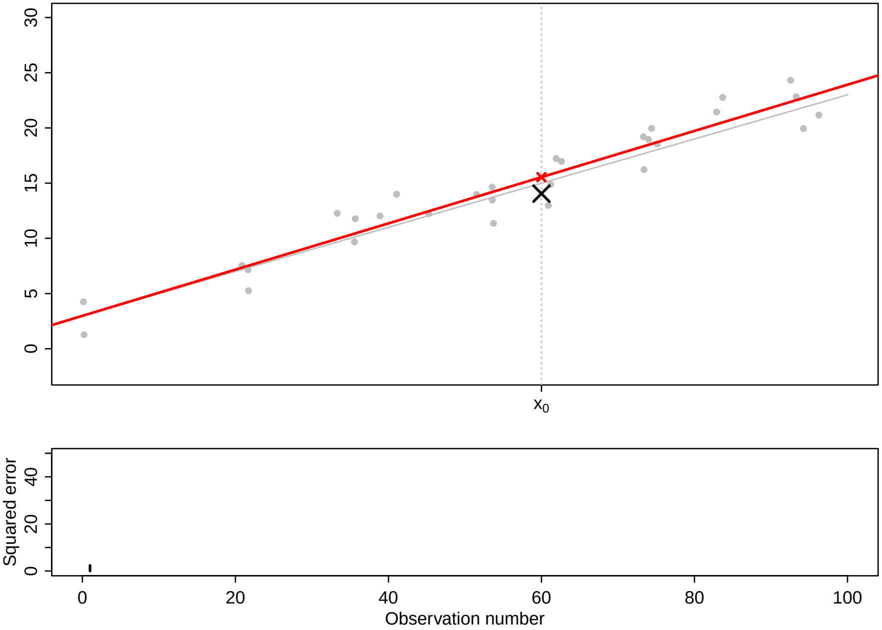
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



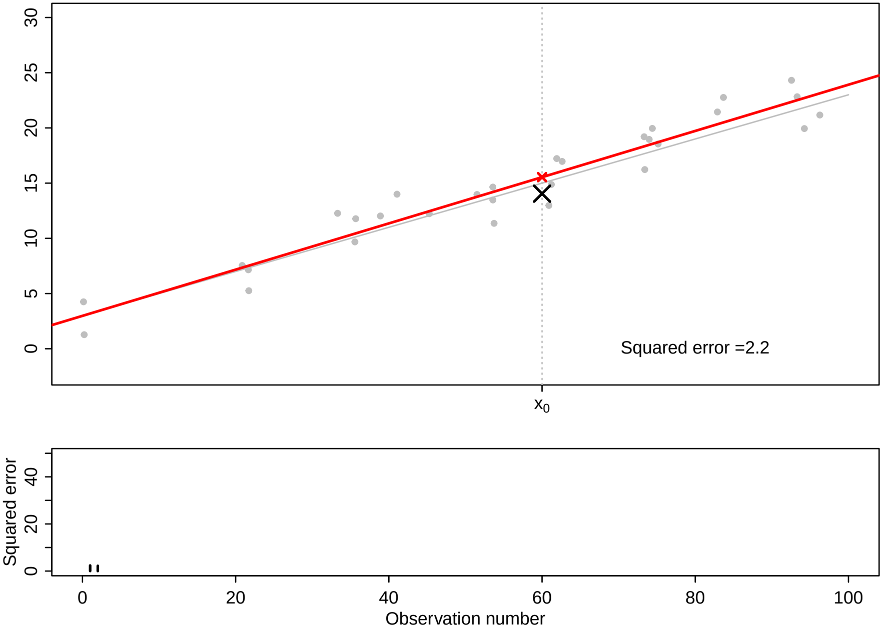
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



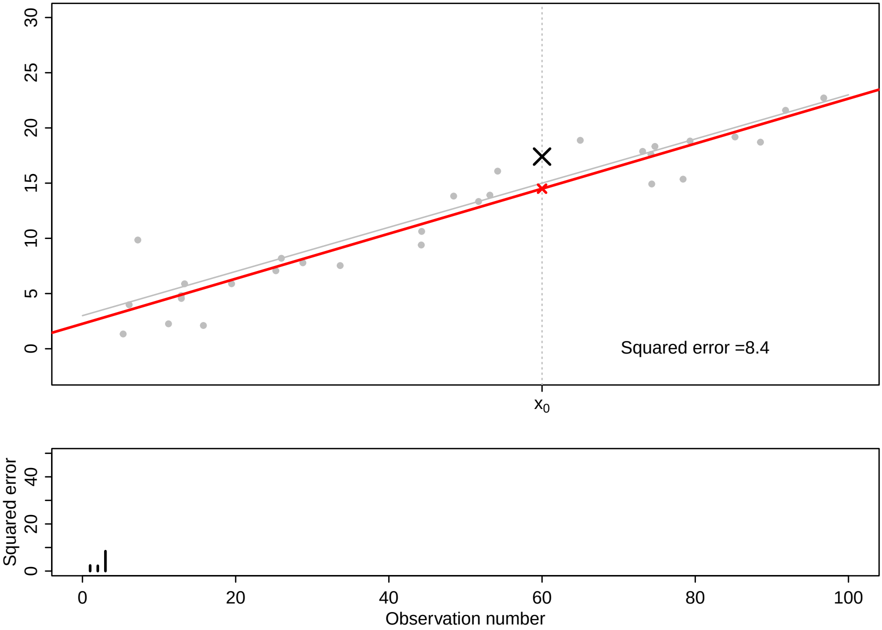
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



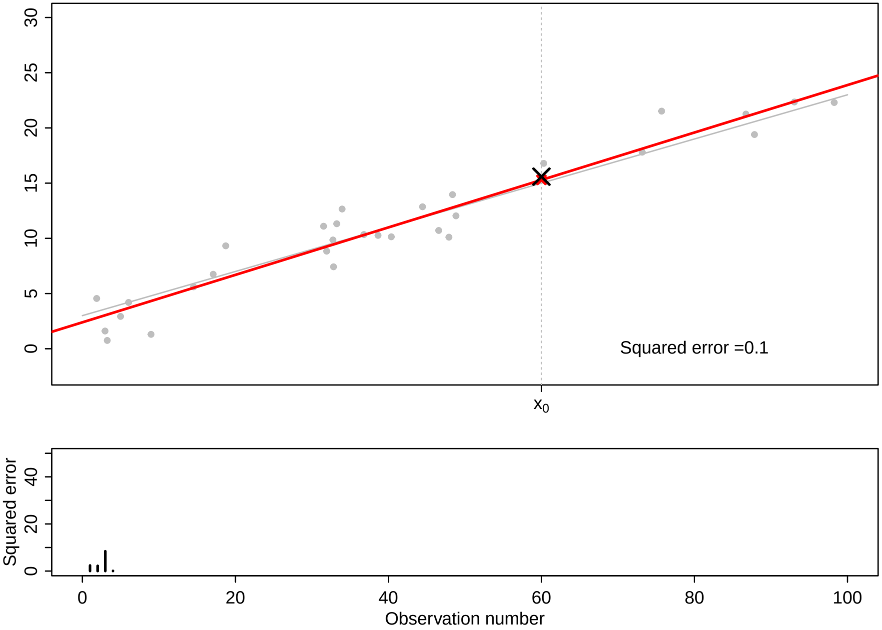
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



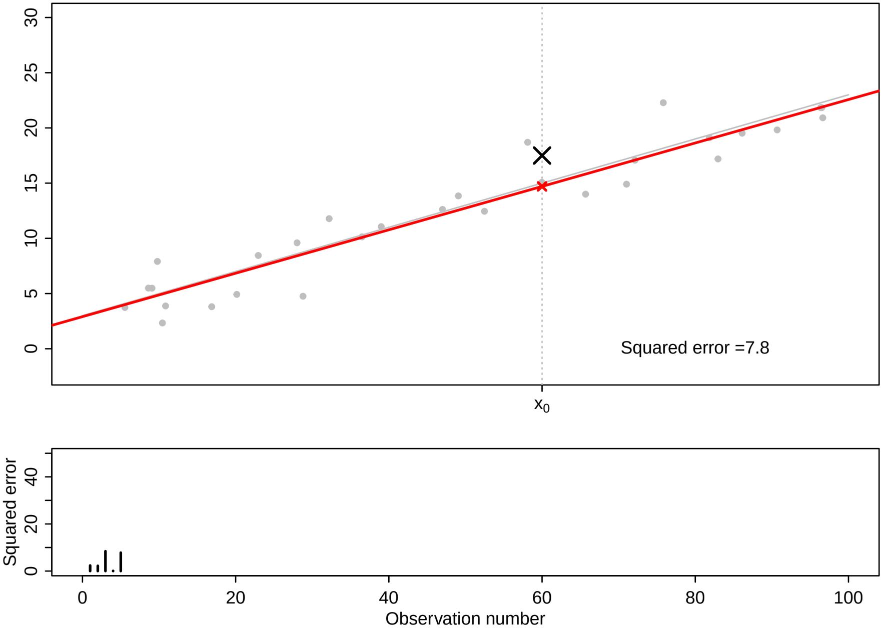
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



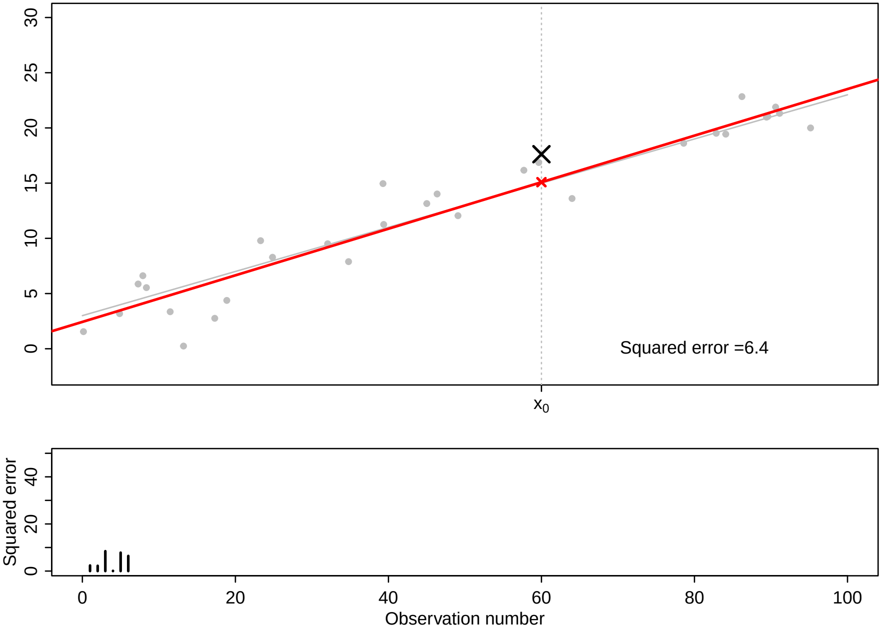
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



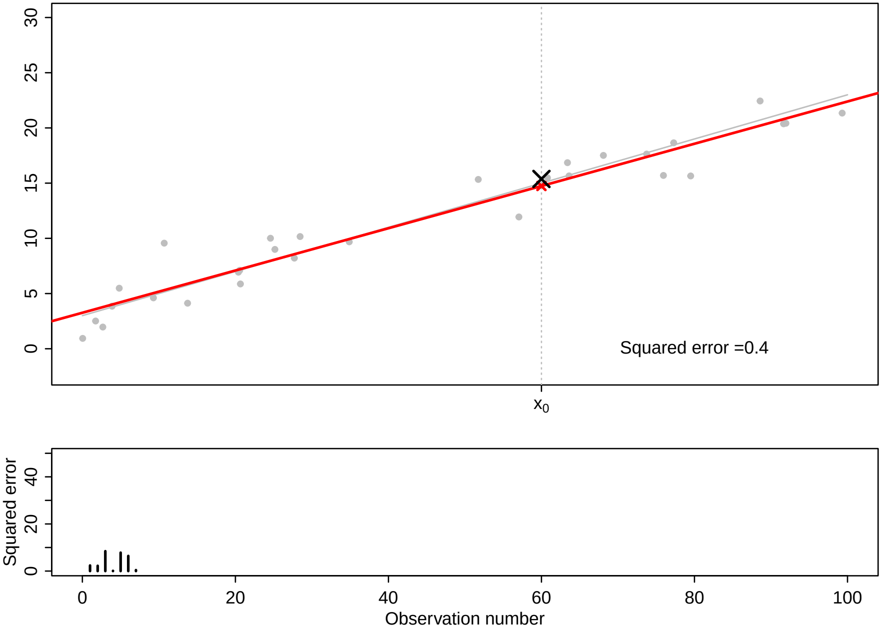
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



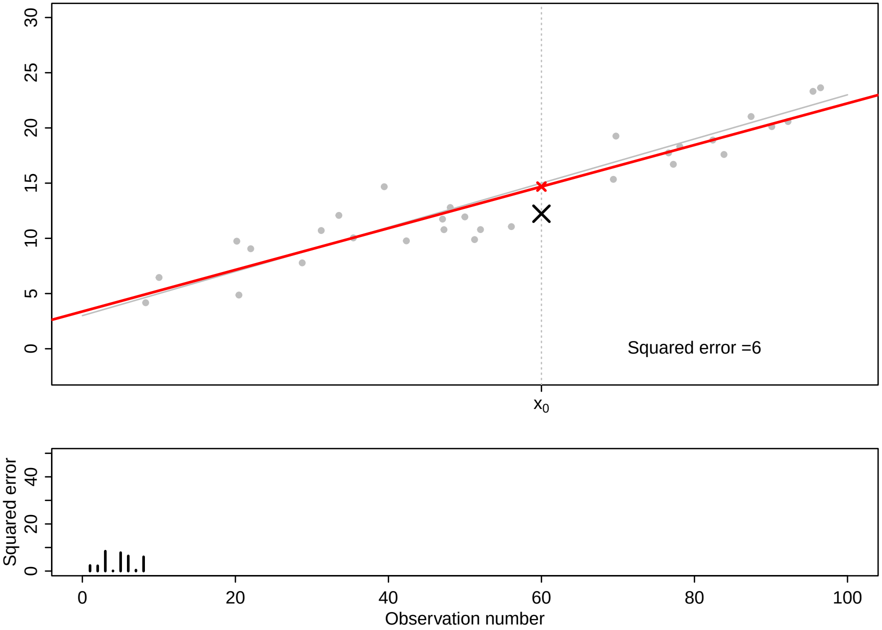
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



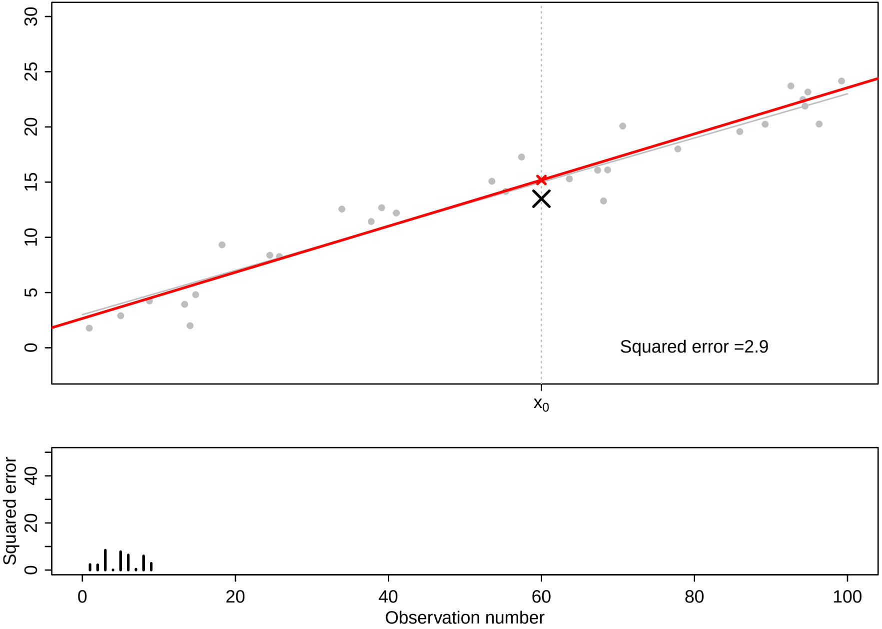
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



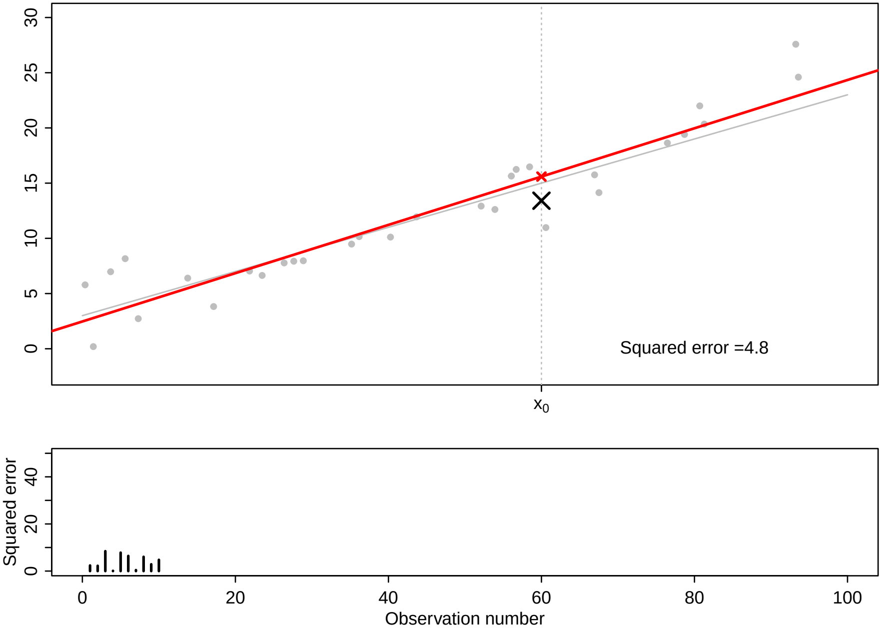
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



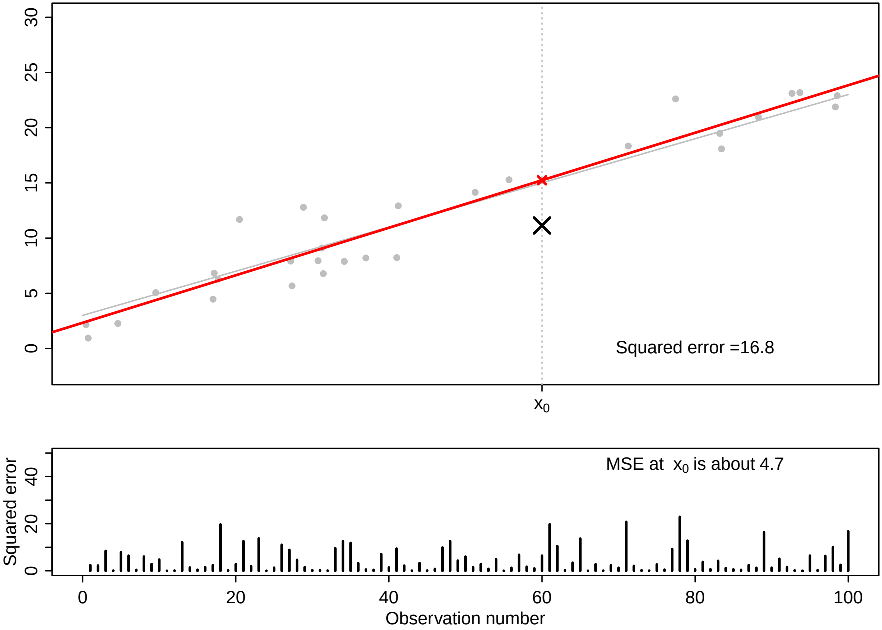
Expected MSE at x_0: Average over $\hat{f}$ and Y_0



Expected MSE at x_0: Average over $\hat{f}$ and Y_0



Expected MSE at x_0: Average over $\hat{f}$ and Y_0



Let us consider now a very general question:

What is the expected squared error that we get from the entire process of first training the learner on some training data and then using it for predicting the response Y_0 *for a new observation* X_0*?*

Remember:

- The function $\hat{f}$ is a random variable, because it is obtained from applying a learner to a dataset of N random variables (X, Y).
- A new observation (X_0, Y_0) is a random variable.

The bias-variance decomposition

The expected test MSE at x_0 can be decomposed as

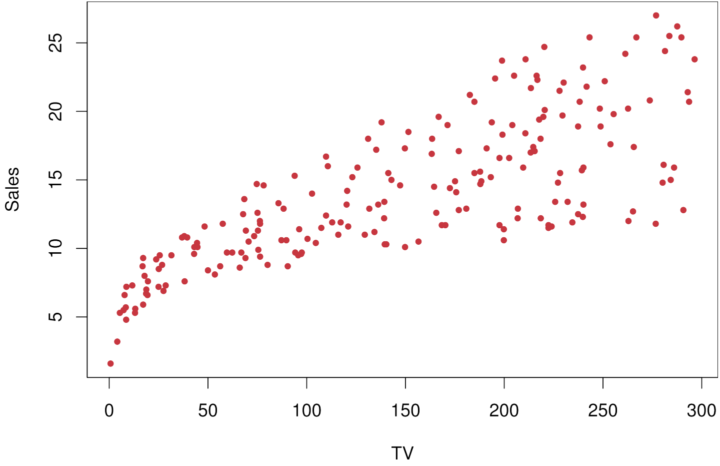
$$\mathbb{E}\left(Y_0 - \hat{f}(x_0)\right)^2 = \underbrace{\mathbb{E}\left(\hat{f}(x_0) - \mathbb{E}\hat{f}(x_0)\right)^2}_{\text{Variance of } \hat{f}(x_0)} + \Big(\underbrace{\mathbb{E}\hat{f}(x_0) - f(x_0)}_{\text{Bias of } \hat{f}(X_0)}\Big)^2 + \mathrm{Var}(\epsilon)$$

First term: The mathematical definition of variance of $\hat{f}(x_0)$. (So "Variance")

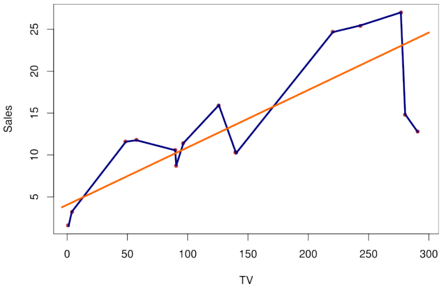
Second term: The expected deviation of our model prediction from the true value. (So "Bias")

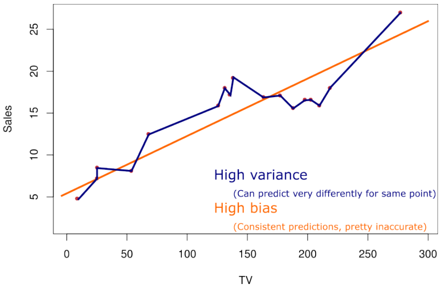
Bias and variance of $\hat{f}(x_0)$?

Let's illustrate this by sampling 2 sets of data points from the advertising data and fitting 2 kinds of models on them.



Bias and variance of $\hat{f}(x_0)$



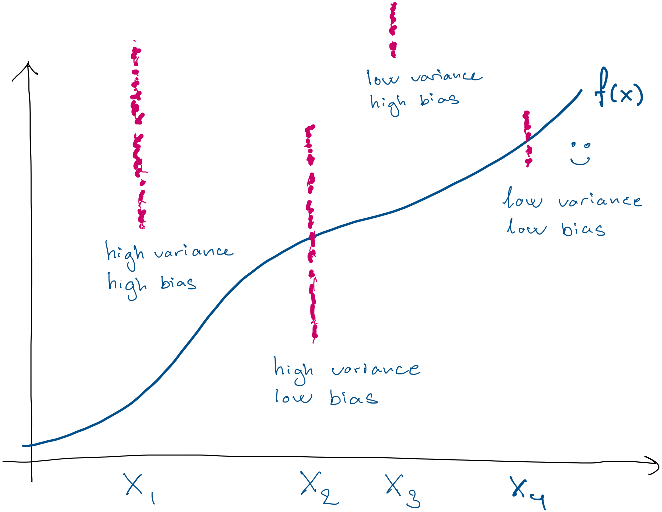


From ISL:

"*Variance* refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set."

"*Bias* refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model."

Bias and variance of $\hat{f}(x_0)$



(In this illustration, red dots are predictions made by different instances of $\hat{f}$)

Bias-variance tradeoff

$$\mathbb{E}\left(Y_0 - \hat{f}(x_0)\right)^2 = \underbrace{\mathbb{E}\left(\hat{f}(x_0) - \mathbb{E}\hat{f}(x_0)\right)^2 + \left(\mathbb{E}\hat{f}(x_0) - f(x_0)\right)^2}_{\text{Reducible error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible error}}$$

All three terms are non-negative, so if any is large, the MSE is large.

The reducible error can be lowered by using an estimator $\hat{f}$ that has both low variance and low bias.

The irreducible error is a lower bound on the accuracy of our prediction for Y. (The bound would typically be unknown.)

Bias-variance tradeoff

Think about the bias and variance of these two learners:

- Fit a constant line through the training data.

- Fit a curve that interpolates all points in the training data.

Bias-variance tradeoff

Think about the bias and variance of these two learners:

- Fit a constant line through the training data.
  – Low variance, high bias

- Fit a curve that interpolates all points in the training data.
  – high variance, low bias

Bias-variance tradeoff

Think about the bias and variance of these two learners:

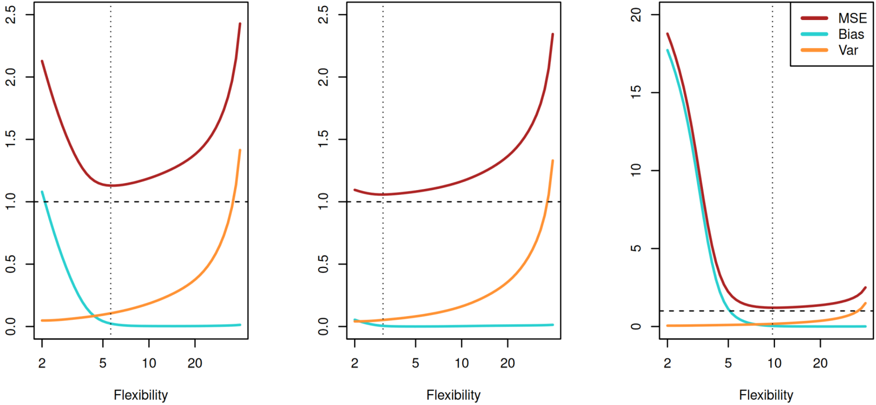
- Fit a constant line through the training data.
  – Low variance, high bias

- Fit a curve that interpolates all points in the training data.
  – high variance, low bias

It is easy to find an estimator with *either* very low bias or very low variance, but less straightforward to find one with *both*.

The problem is referred to as the *the bias-variance tradeoff*.

Example: Bias, variance, and MSE as model complexity increases



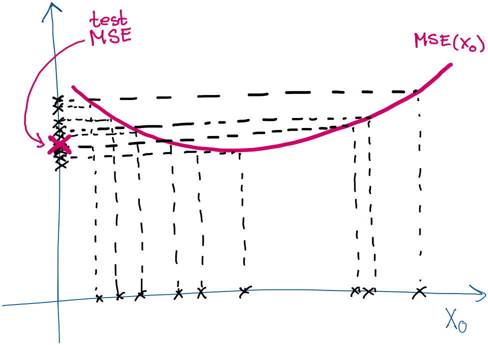
"Flexibility" is model complexity.

Panels are 3 different data sets.

Q: What do these 3 panels have in common?

Expected test MSE

So far, we have considered the test MSE at an individual point x_0. To get a great model fit, we are not satisfied with doing well at one point. We want the expected test MSE to be good across all possible values of x_0, not just *for a specific value* of the feature.



Expected test MSE averages also over all x_0 in the test data.

Expected test MSE

In probabilistic terms, what we want to do is take expectation over new values X_0 of the feature as well as over Y_0 and $\hat{f}$.

It can be done sequentially, by

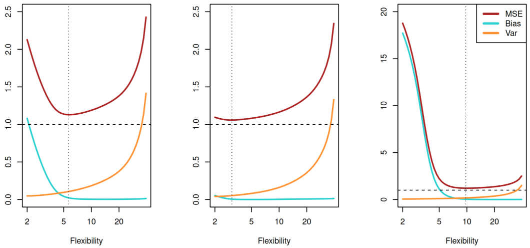
- first finding the *MSE at* x_0, which is then a function of X_0,
- and then finding the expectation of the MSE at x_0 when x_0 varies:

$$\mathbb{E}(MSE(\hat{f}, Y_0, X_0)) = \mathbb{E}_{X_0}\left(\mathbb{E}_{\hat{f},Y_0}(MSE(\hat{f}, Y_0, X_0) \mid X_0 = x_0)\right)$$

Expected test MSE

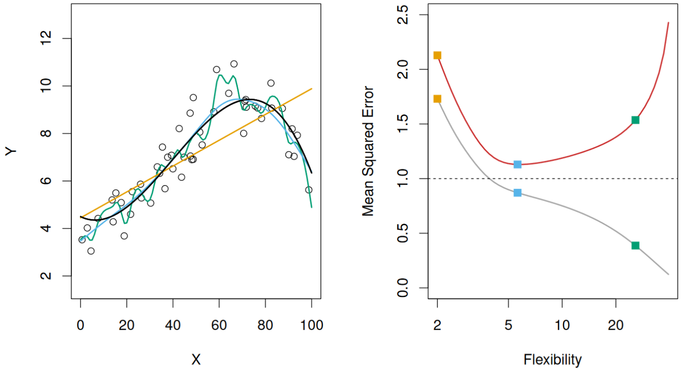
Taking expectation over X_0 in the bias-variance decomposition gives a decomposition of the Expected test MSE into three perhaps less directly interpretable terms:

$$\text{MSE} = \underbrace{\mathbb{E}\left(\text{Var}(\hat{f}(X_0))\right)}_{\substack{\text{“Average” variance of} \\ \hat{f}(x_0) \text{ across the range} \\ \text{of } X_0.}} + \underbrace{\mathbb{E}\left(\text{Bias}(\hat{f}(X_0))\right)^2}_{\substack{\text{“Average” squared bias} \\ \text{of } \hat{f}(x_0) \text{ across the} \\ \text{range of } X_0.}} + \underbrace{\text{Var}(\epsilon)}_{\substack{\text{Residual} \\ \text{variance}}}$$



Test MSE vs training MSE

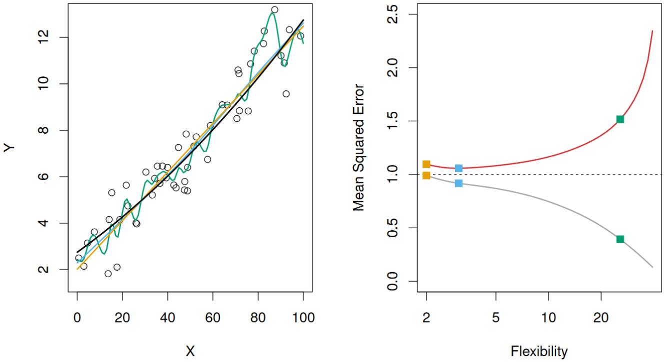
The MSE is a theoretical quantity that we can estimate from data. We use data points that were not used during training (the *test set*). [Below: Test MSE (red) and training MSE (grey)]



This is because ML models can *overfit* to the training data: If we fit too well to the training data, our model does not generalize well (general problem!). Aim for a sweet spot where the test MSE is low.

Test MSE vs training MSE

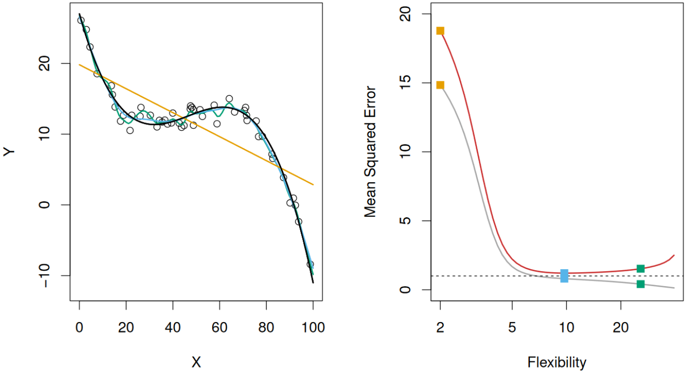
The MSE is a theoretical quantity that we can estimate from data. We use data points that were not used during training (the *test set*). [Below: Test MSE (red) and training MSE (grey)]



This is because ML models can *overfit* to the training data: If we fit too well to the training data, our model does not generalize well (general problem!). Aim for a sweet spot where the test MSE is low.

Test MSE vs training MSE

The MSE is a theoretical quantity that we can estimate from data. We use data points that were not used during training (the *test set*). [Below: Test MSE (red) and training MSE (grey)]



This is because ML models can *overfit* to the training data: If we fit too well to the training data, our model does not generalize well (general problem!). Aim for a sweet spot where the test MSE is low.

Test MSE vs training MSE

We should keep in mind a distinction between the following two problems

Model selection:	estimating the prediction error of different models with the purpose of choosing the best one.
Model assessment:	having chosen a final model, estimating its prediction error.

Holding out data for testing

In the linear regression model, we used a single dataset to train the models and selected between models using p-values or AIC.

This is great: We evaluate our model on data that the model has never seen!

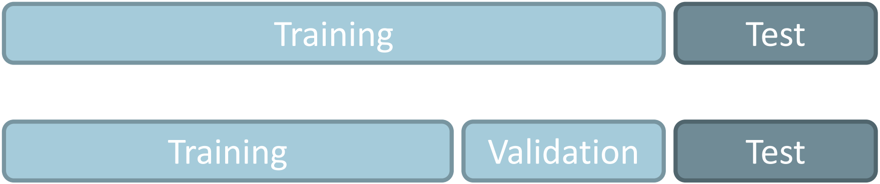
Sometimes, we have *hyperparameters* to specify in our model.

Perhaps we have a $\lambda \in [0, 1]$ where a specific value gives the best-performing model.

Q: How can we train the model for different λ and choose the best value without "fitting" the model to our test data?

Holding out data for testing

In the linear regression model, we used a single dataset to train the models and selected between models using p-values or AIC.



If we wish to use the MSE to guide the model building – selecting features, tuning *hyperparameters* – then we should preferably create a validation set for estimating the MSE and leave the test set for assessing performance.

A typical split could be 60-20-20 for the three datasets, but it depends on how much data you have.