What Is Data Science (DS)?

- There's a saying that a data scientist is someone who knows more statistics than a computer scientist and more computer science than a statistician

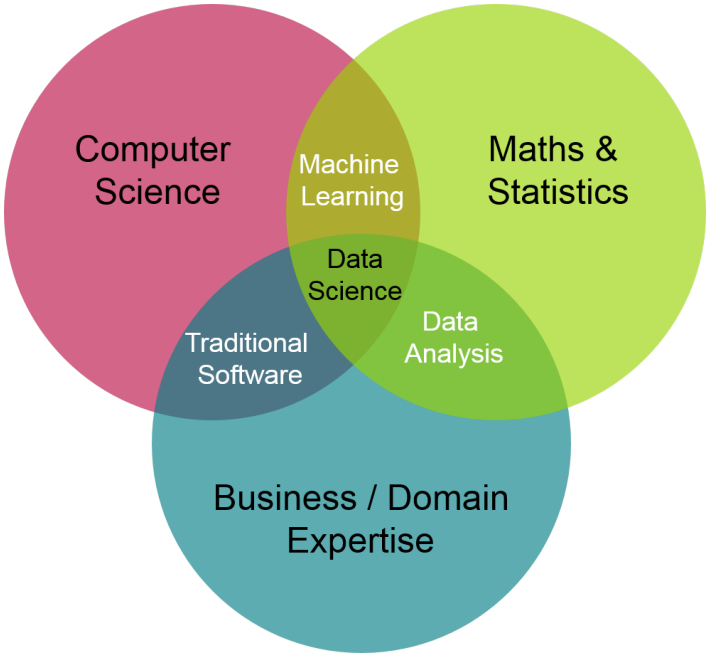
- The truth is that there are different kinds of data scientists
  - some are – for all practical purposes – statisticians
  - others are pretty much indistinguishable from software engineers
  - some are machine-learning experts

  etc.

What is Machine Learning (ML)?

- It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning.

- Application of ML methods has become ubiquitous in everyday life:
  - from automatic recommendations of which movies to watch, to what food to order or which products to buy
  - personalized online radio
  - recognizing your friends in your photos etc. … many modern websites and devices have ML algorithms at their core
  - outside of commercial applications, ML has had a big influence on the way data-driven research is done today, e.g. discovering new particles, analyzing DNA sequences, and providing personalized cancer treatments.



- Diagrams like this are trying to explain the genesis and place of DS and ML. It is clear that a data scientist is someone who extracts insights from messy data and who needs for that many skills

- We are going to focus on the data analysis and ML skills

Machine (Statistical) Learning

Some more examples of machine learning problems:

- Predict whether a patient, hospitalized due to a heart attack, will have a second heart attack. The prediction is to be based on demographic, diet and clinical measurements for that patient
- Predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data
- Identify the numbers in a handwritten ZIP code, from a digitized image

A typical task of learning from data has an outcome measurement (quantitative or categorical) that we wish to predict based on a set of features.

Statistical Learning Principles

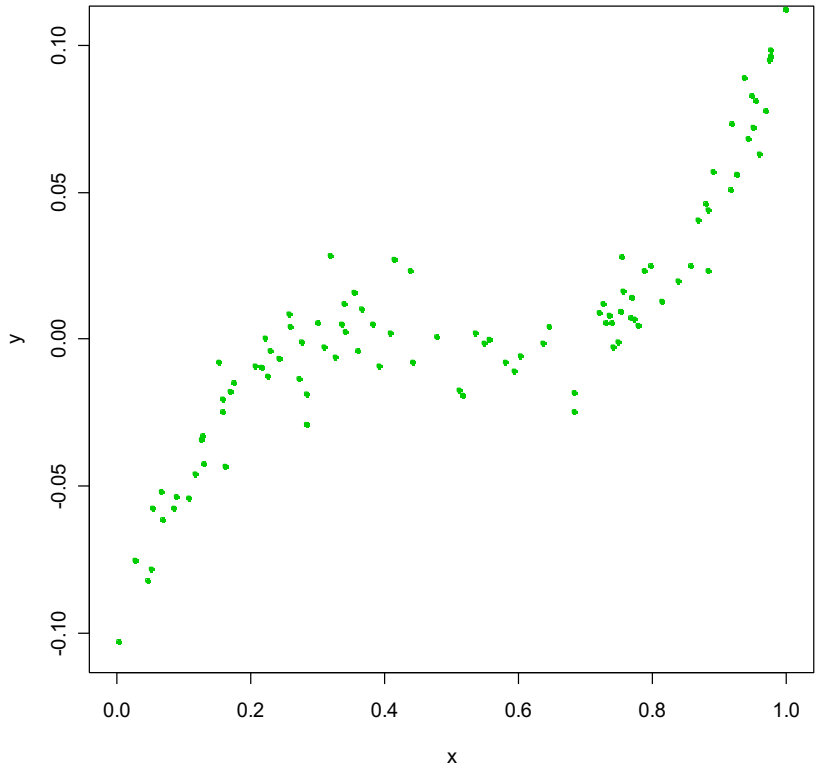
- Suppose we observe a (quantitative) response Y and p predictors (or features) $X_1, X_2, \ldots, X_p$
- Assume that there is a relationship between Y and at least one of the X's.
- We can model the relationship as

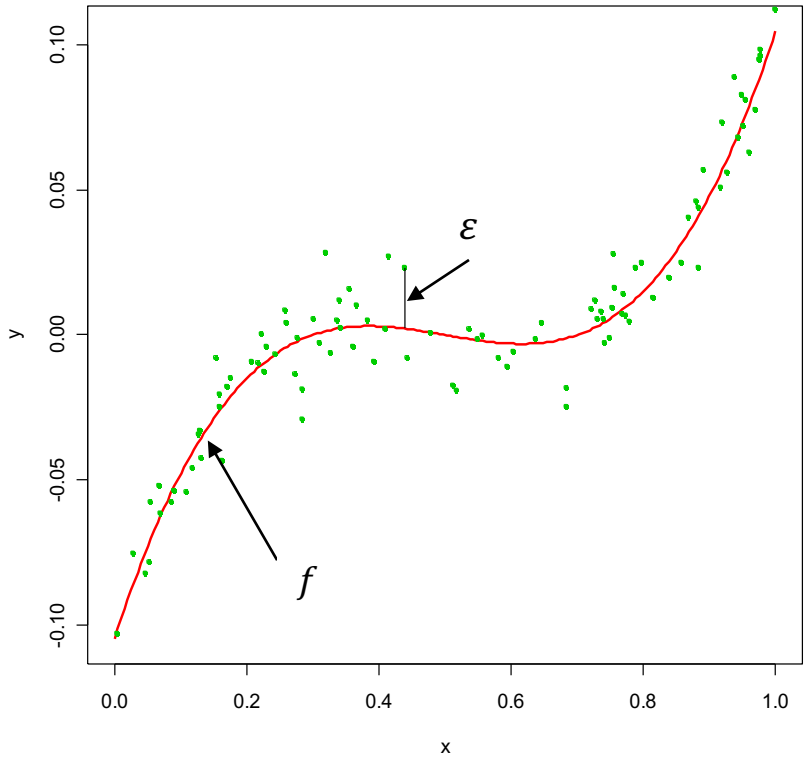
$$Y = f(X) + \varepsilon$$

where f is an unknown function, $X = (X_1, X_2, \ldots, X_p)$ and ε is a random error (independent of X) with mean zero. Here f represent the systematic information that X provides about Y.

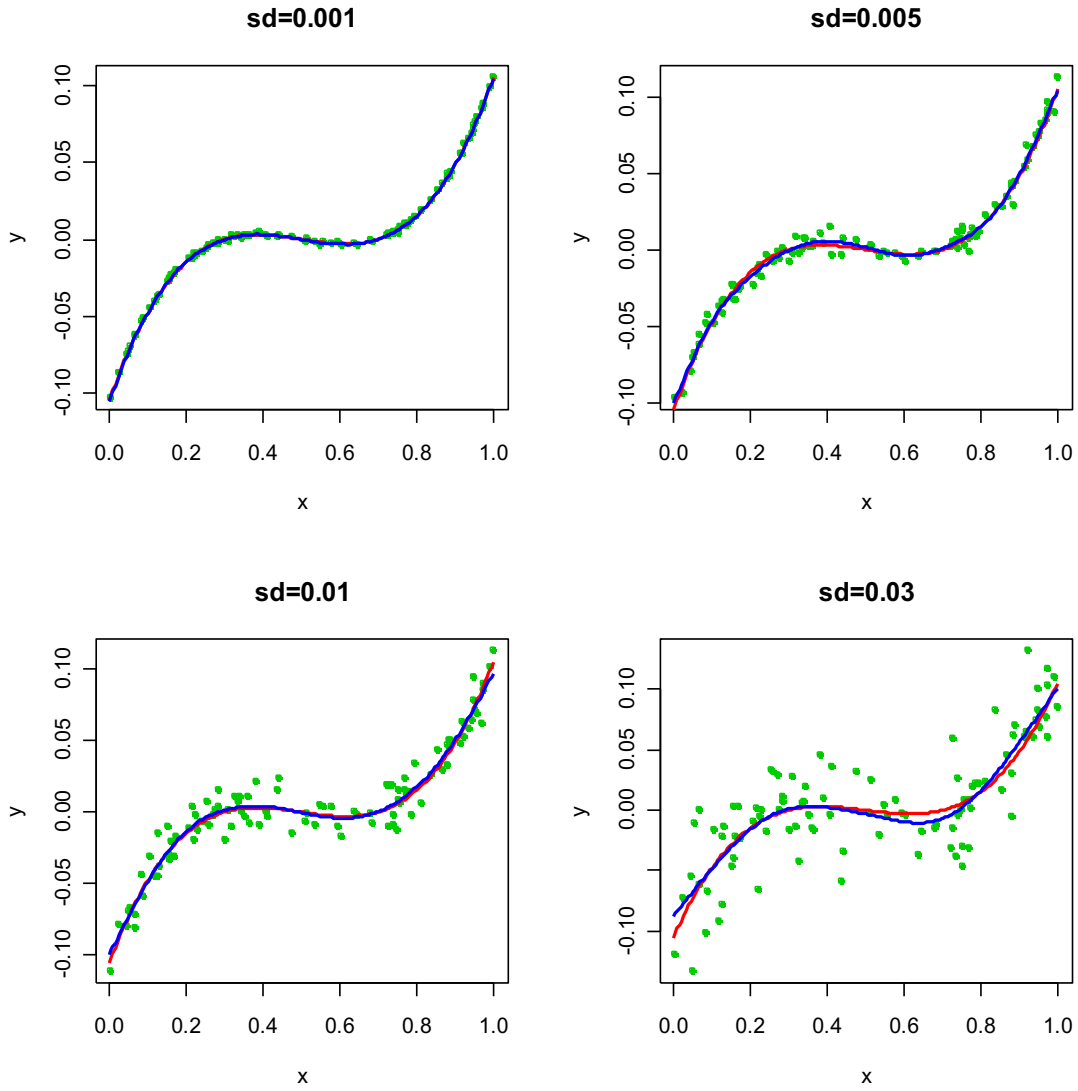
- In essence, statistical learning refers to a set of approaches for estimating f

Example 1 (p=1) – data and a smooth f modeling the data

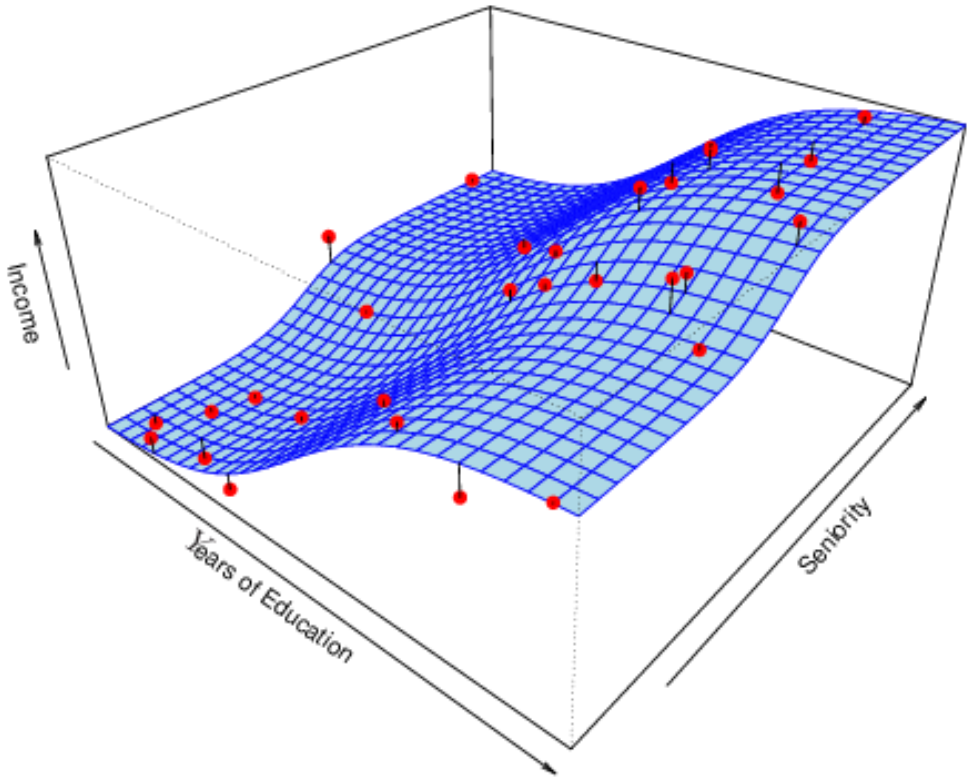




The amount of noise (the standard deviation of ε) affects the choice of f



Example 2 (p=2) – **Income** is modeled as a function of Education & Seniority



The surface represents the true relationship (the function f) which is known since the data is simulated. The dots indicate 30 observed values.

Why model f?

- Statistical Learning is about how to estimate (or "learn") f
- We use data in the process
- The reasons for estimating f are
  - Prediction
  - Inference
  - Mixture of the two above

Prediction

- In many situations, the predictors X are known but not the outcome Y which is hard or impossible to obtain
- With a good estimate $\hat{f}$ of f (and if the variance of ε is not too large) we can predict Y accurately by

$$\hat{Y} = \hat{f}(X)$$

- In this situation, $\hat{f}$ is often treated as a black box – we don't care what the exact form of f is as soon as it is providing good predictions

Example 3 – Direct mailing prediction

- Interested in predicting how much money an individual will donate based on observations from 90,000 people on which we have recorded over 400 different characteristics
- Don't care too much about each individual characteristic
- Just want to know: For a given individual should I send out a mailing?

Inference

- Alternatively, we may also be interested in the type of relationship between Y and the X's. In this case $\hat{f}$ can not be treated as a black box
- The following question might be of interest in this case
  - Which particular predictors actually affect the response? Identify the important predictors
  - Is the relationship positive or negative?
  - Is the relationship a simple linear one or is it more complicated etc.?

Example 4 – Inference (and Prediction)

- Wish to predict median house price based on 14 variables
- Probably want to understand which factors have the biggest effect on the response and how big the effect is (inference)
- For example how much impact does a river view have on the house value etc. (inference)

Supervised vs. Unsupervised Learning

We can divide all learning problems into Supervised and Unsupervised

- Supervised Learning
  - Supervised Learning is where both the predictors, X_i, and the response, Y, are observed.
  - Most of this course will also deal with supervised learning.
- Unsupervised Learning
  - In this situation only the X_i's are observed.
  - We could use the X_i's to guess what Y would have been and build a model from there
  - A common example is market segmentation where we try to divide potential customers into groups based on their characteristics
  - A common approach is clustering

Regression vs. Classification

- Supervised learning problems can be further divided into regression and classification problems.
- Regression covers situations where Y is continuous/numerical, e.g.
  - Predicting the value of the Dow in 6 months
  - Predicting the value of a given house based on various inputs
- Classification covers situations where Y is categorical, e.g.
  - Will the Dow be up (U) or down (D) in 6 months?
  - Is this email a SPAM or not?
- We will deal with both types of problems
- Now, we continue with the estimation problem in Supervise Learning

Estimating f

- We will assume we have observed a set of training data with p predictors and n observations

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_p)\},$$

  Here ${x_i}^T = (x_{i1}, x_{i2}, \dots, x_{ip}), i = 1,2, \dots, n$ is the i-th observation (row of data)

- We then use the training data and a Statistical Learning method to estimate f

- We classify the Statistical Learning methods broadly into
  - Parametric Methods
  - Non-parametric Methods

Parametric Methods

- They reduce the problem of estimating f down to one of estimating a set of parameters.

- Parametric models involve a two-step model based approach

STEP 1:

Make some assumption about the functional form of f, i.e. come up with a model. The most common example is a linear model i.e.

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

However, we will examine far more complicated, and flexible, models for f. In a sense the more flexible the model the more realistic it is.

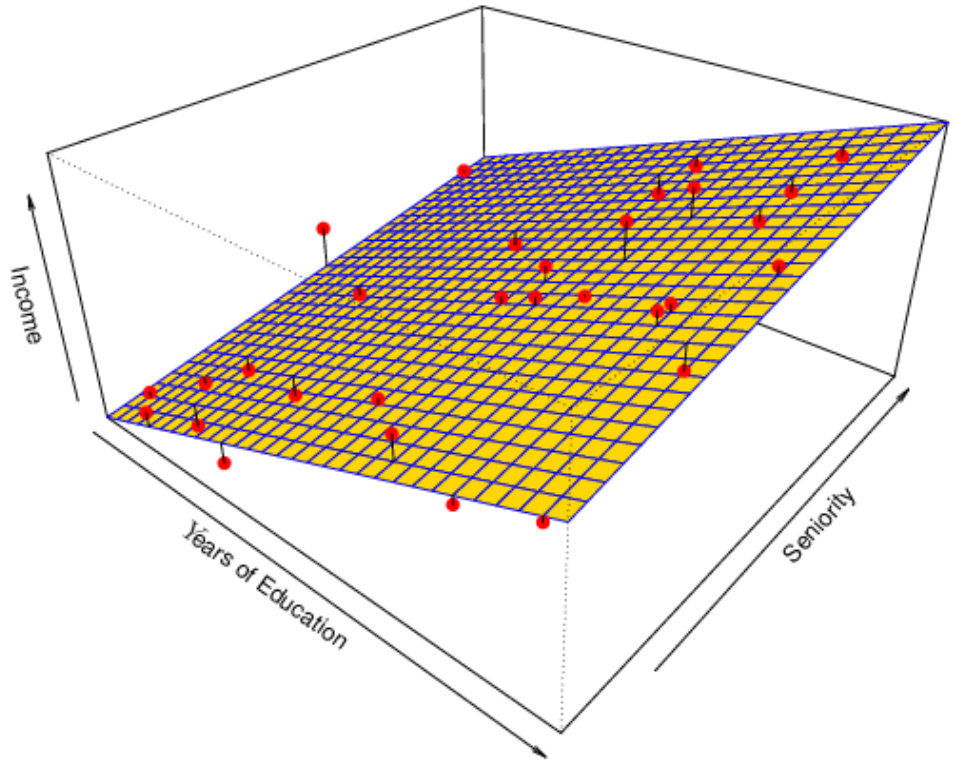
STEP 2:

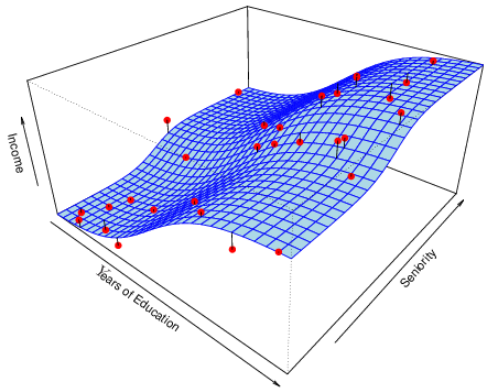
Use the training data to fit the model i.e. estimate f or equivalently the unknown parameters such as $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$

- The most common approach for estimating the parameters in a linear model is Ordinary Least Squares (OLS)
- However, this is only one way
- We will see in the course that there are often superior approaches

Example 2 again – The data is synthetic so we know the actual f (the blue surface on the right). The linear model:

$$f = \beta_0 + \beta_1 \times Education + \beta_2 \times Seniority$$





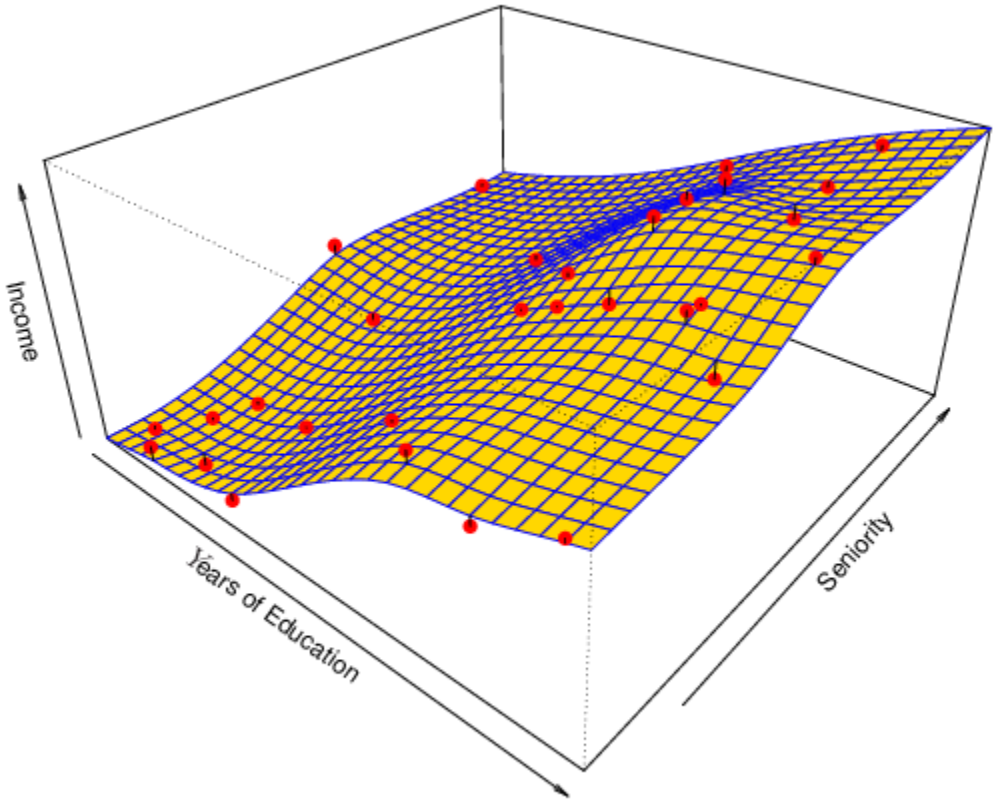
generates a (hyper)plane – the orange surface.

Non-Parametric Methods

They do not make explicit assumptions about the functional form of f

- Advantages: They accurately fit a wider range of possible shapes of f
- Disadvantages: A very large number of observations is required to obtain an accurate estimate of f

Example 2 again – **Tin-plate spline** estimate is one non-parametric method. It produces an estimate $\hat{f}$ very close to the real f



The danger with the more flexible methods is that they can overfit.

In short this means the model gets too close to the training data and can't predict that well on new data.

Some additional considerations referring to the above two models are:

- A simple method such as linear regression produces a model which is much easier to interpret (Inference is better). For example, in a linear model, β_j is the average increase in Y for one unit increase in X_j holding all other variables constant

- Even if you are only interested in prediction, so the interpretability is not really an issue, it is often possible to get more accurate predictions with a simple, instead of a complicated, model. This seems counter intuitive but has to do with the fact that it is harder to fit a more flexible model

- So there is also a tradeoff between simple and more complex models that is relevant to prediction accuracy. It will be explained in detail after we introduce ways to access the accuracy of the model.

Assessing Model Accuracy

Measuring the quality of fit

- Suppose we have a regression problem
- One common measure of accuracy is the mean squared error (MSE) i.e.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the prediction our method gives for the outcome value y_i in our training data

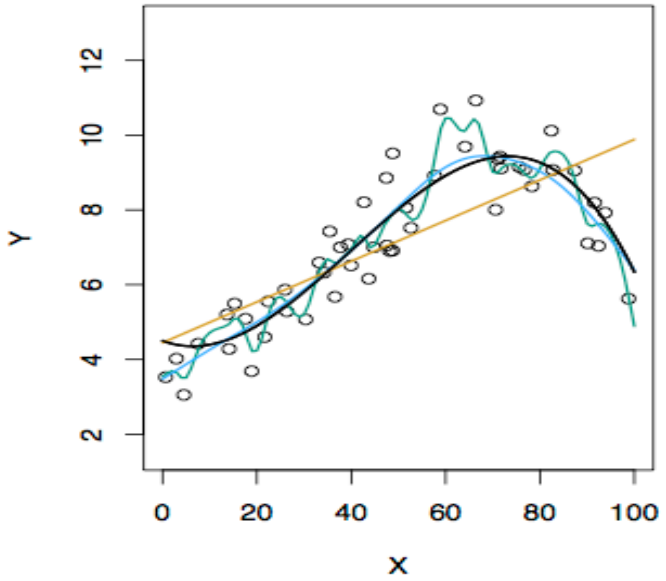
Caveats:

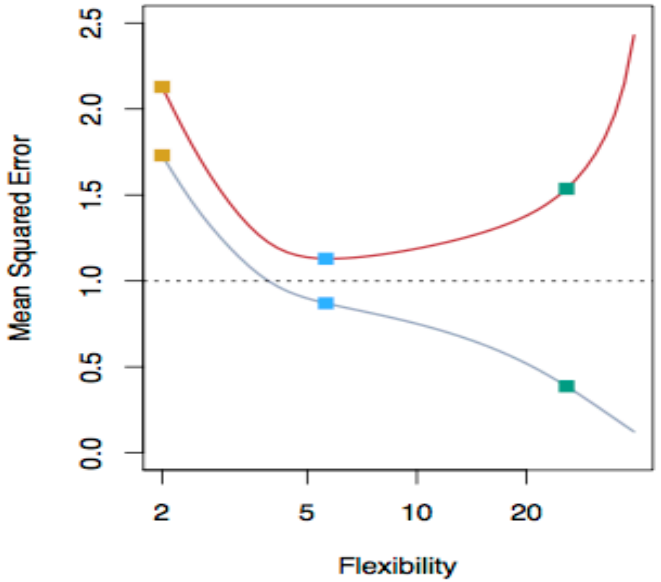
- The methods have generally been designed to make MSE small on the training data. E.g. with linear regression we choose the line such that MSE is minimized

- What we really care about is how well the method works on new data. We call this new data "test data"

- There is no guarantee that the method with the smallest training MSE (the most flexible one perhaps) will have the smallest test (i.e. new data) MSE

Examples with Different Levels of Flexibility





LEFT

Black: Truth

Orange: Linear Estimate

Blue: Smoothing spline

Green: Smoothing spline (more flexible)

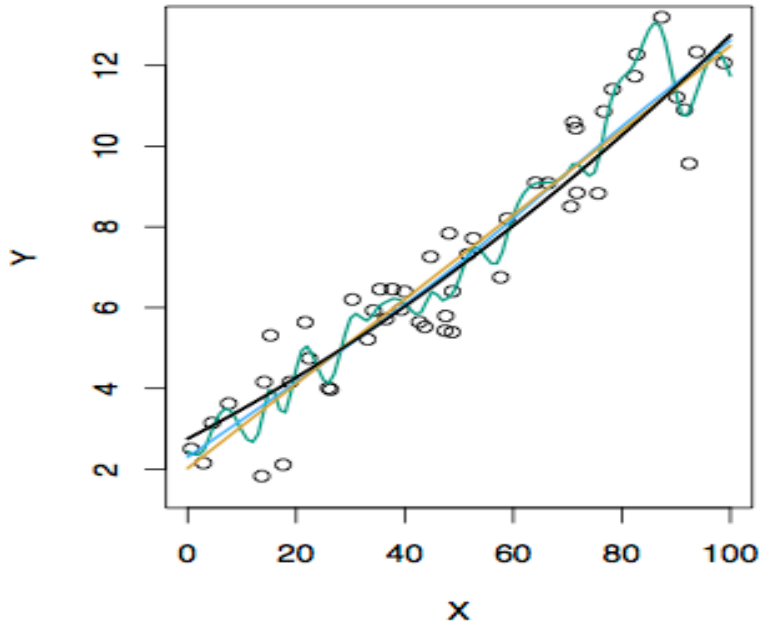
RIGHT

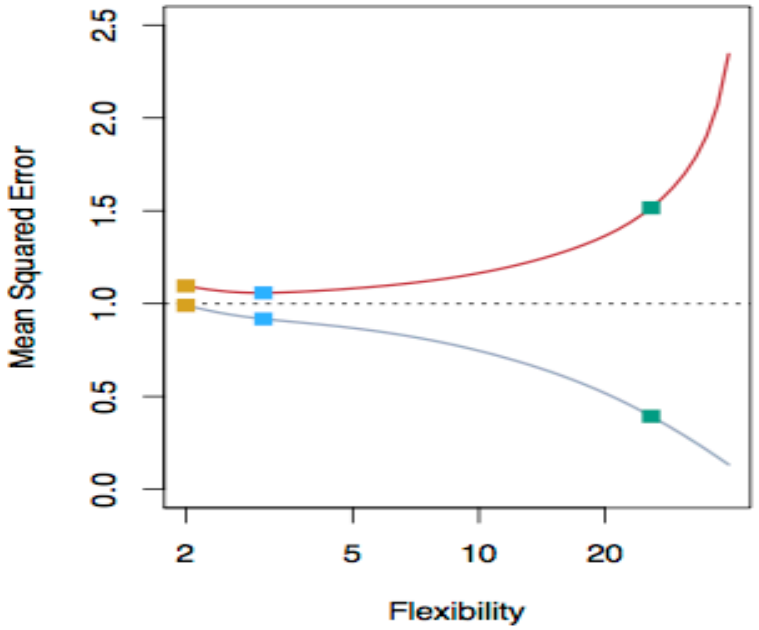
RED: Test MES

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

***Squares**: Training and test MSEs for the 3 fits shown in the left-hand panel.*





LEFT

Black: Truth

Orange: Linear Estimate

Blue: Smoothing spline

Green: Smoothing spline (more flexible)

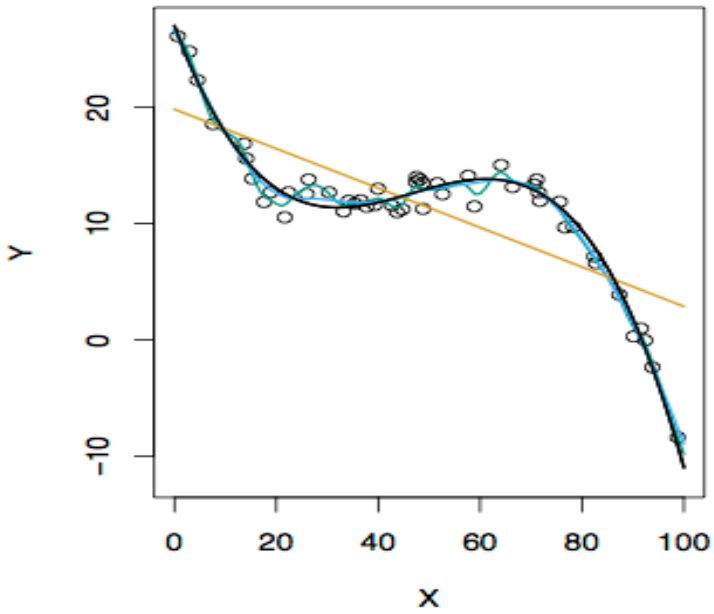
RIGHT

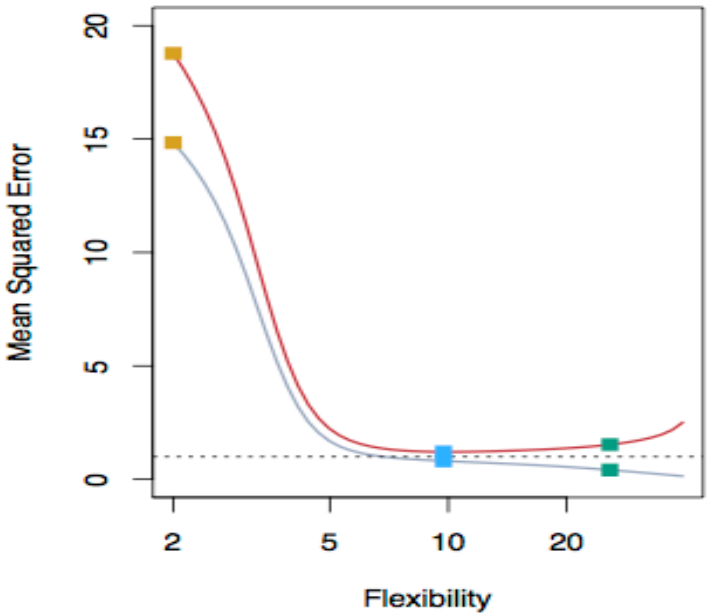
RED: Test MES

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

***Squares**: Training and test MSEs for the 3 fits shown in the left-hand panel.*





LEFT

Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

RIGHT

RED: Test MES

Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

***Squares**: Training and test MSEs for the 3 fits shown in the left-hand panel.*

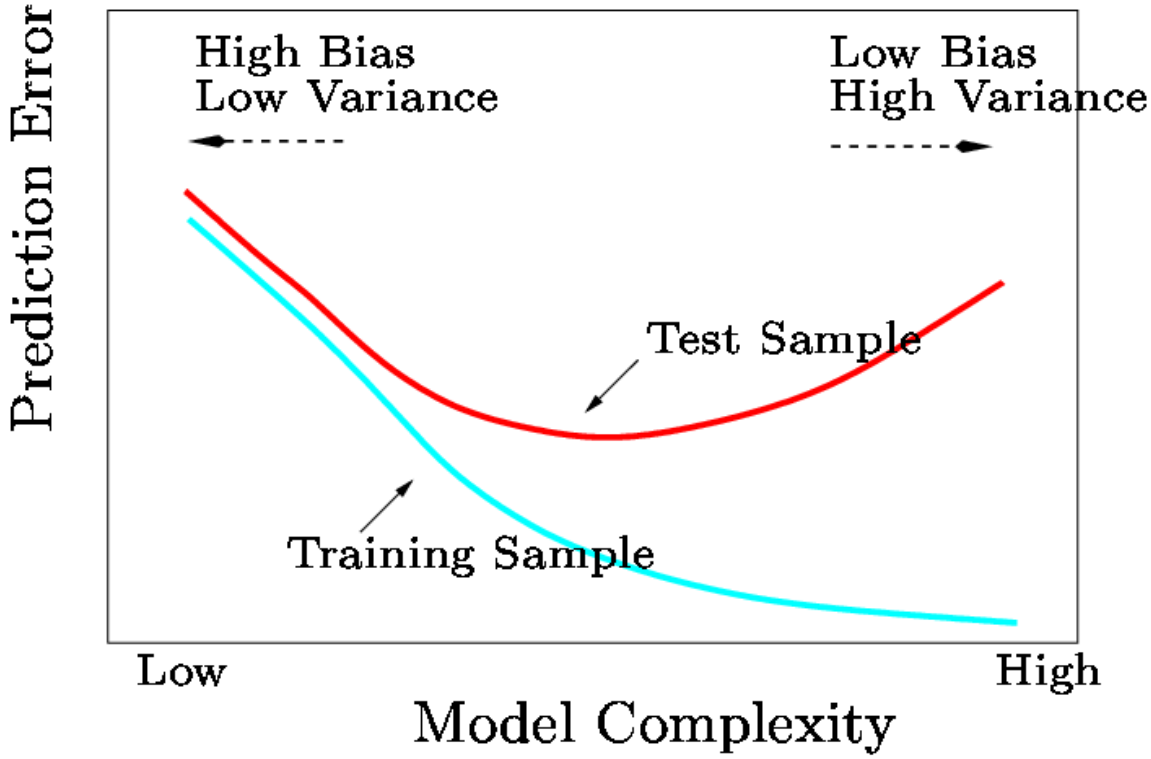
Bias-Variance Tradeoff

- The previous graphs of test versus training MSE's illustrates a very important tradeoff that governs the choice of statistical learning methods
- There are always two competing forces that govern the choice of learning method, i.e. bias and variance
- Bias refers to the error that is introduced by modeling a real life problem (that is usually extremely complicated) by a much simpler problem
- For example, linear regression assumes that there is a linear relationship between Y and X. It is unlikely that, in real life, the relationship is exactly linear so some bias will be present
- The more flexible/complex a method is the less bias it will generally have
- Variance refers to how much your estimate for f would change by if you had a different training data set
- Generally, the more flexible a method is the more variance it has

Fundamental Principle

In general training errors will always decline

However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).



Moral: A more complex model is not always better

The Bias-Variance Tradeoff Explained

- It can be shown (see **Chap 7.3, p.233 in [ESLII]**) that for any given, $X = x_0$, the expected test MSE for a new Y at x_0 will be equal to

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}_{\mathcal{T}}\left(\hat{f}(x_0)\right) + \left[\mathrm{Bias}\left(\hat{f}(x_0)\right)\right]^2 + \underbrace{\mathrm{Var}(\varepsilon)}_{\text{irreducible error}}$$

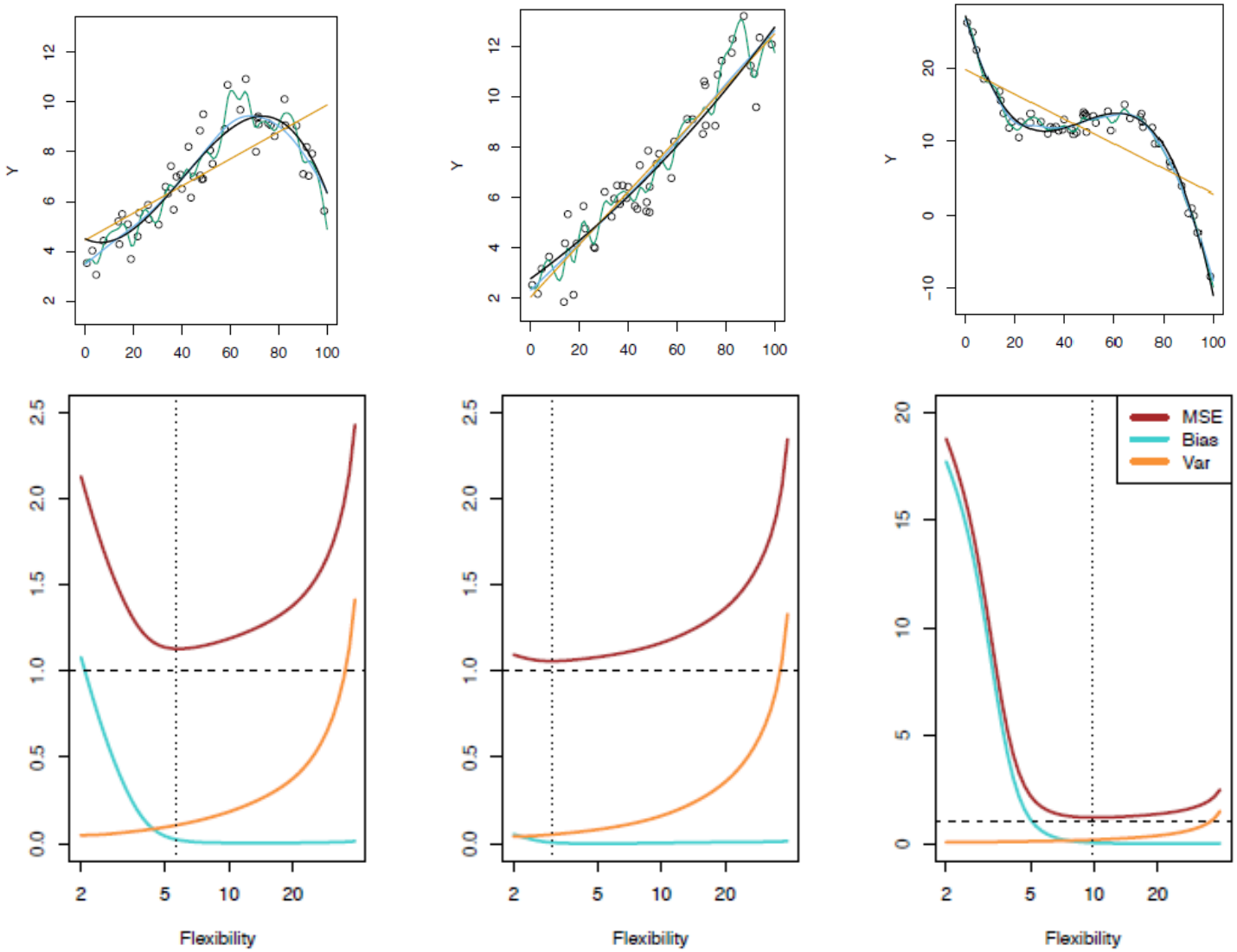
  where the expression on the left is average test MSE that we would obtain if we repeatedly estimated f and using many training datasets and tested each at x_0. The 1st term reflects the variance in the estimate $\hat{f}$ due to using different train sets $\mathcal{T}$ (indicated by subscript $\mathcal{T}$)

- What this means is that as a method gets more complex the bias will decrease and the variance will increase but expected test MSE may go up or down

Reading:

[**ISLR**]: Chapter 1, Chapter 2 (2.1-2.2)

Test MSE, bias and variance



Python sources

https://www.codecademy.com/articles/install-python3

Step1: If you are interested in data science or machine learning then starting with Miniconda is a good choice.

https://docs.conda.io/en/latest/miniconda.html

Step 2: Install Jupyter Notebook (JupyterLab)

USING MINICONDA
Follow the below instructions to install the Jupyter Notebook package using the Miniconda package manager conda.

```
conda install jupyter
```

```
conda install -c conda-forge jupyterlab
```

Step 3: Once complete, we can check that Jupyter Notebook was successfully installed by running jupyter notebook / jupyter lab from a Terminal (Mac) / Command Prompt (Windows):

```
$ jupyter lab
```

This will startup the Jupyter Notebook server, print out some information about the notebook server in the console, and open up a new browser tab to http://localhost:8888

Some introductory Python sources

https://nbviewer.jupyter.org/github/jakevdp/WhirlwindTourOfPython/blob/master/00-Introduction.ipynb

https://diveintopython3.net/index.html