**Statistical Learning: Chapter 1 & 2**
**Chapter 1: Examples of statistical learning problems**

## 1. Wage data

This data is described in ISLR

Wages for 3000 males in "Middle Atlantic" USA
12 variables recorded:

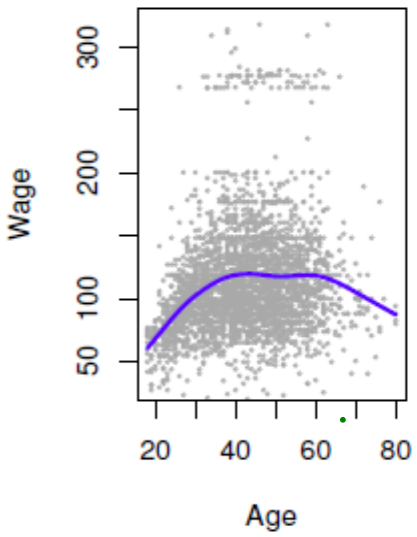
not very useful variable

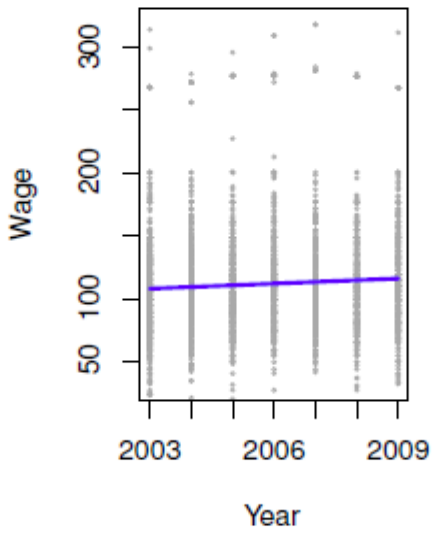
```
> summary(Wage)
      year           age               sex                    maritl           race
 Min.   :2003   Min.   :18.00   1. Male  :3000   1. Never Married: 648   1. White:2480
 1st Qu.:2004   1st Qu.:33.75   2. Female:   0   2. Married      :2074   2. Black: 293
 Median :2006   Median :42.00                    3. Widowed      :  19   3. Asian: 190
 Mean   :2006   Mean   :42.41                    4. Divorced     : 204   4. Other:  37
 3rd Qu.:2008   3rd Qu.:51.00                    5. Separated    :  55
 Max.   :2009   Max.   :80.00

                education                     region               jobclass
 1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544
 2. HS Grad        :971   1. New England       :   0   2. Information:1456
 3. Some College   :650   3. East North Central:   0
 4. College Grad   :685   4. West North Central:   0
 5. Advanced Degree:426   5. South Atlantic    :   0
                          6. East South Central:   0
                          (Other)              :   0
            health      health_ins      logwage           wage
 1. <=Good      : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09
 2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38
                                     Median :4.653   Median :104.92
                                     Mean   :4.654   Mean   :111.70
                                     3rd Qu.:4.857   3rd Qu.:128.68
                                     Max.   :5.763   Max.   :318.34
```

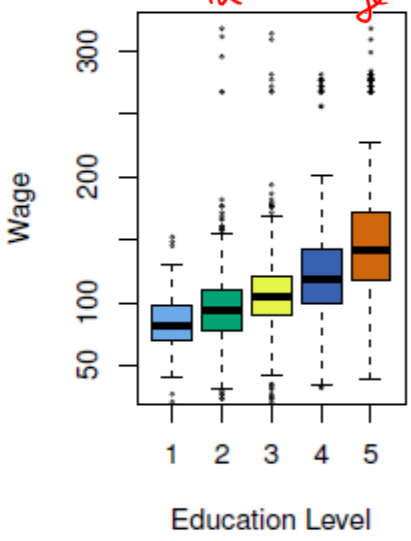
How is "wage" related to age, education, calendar year?

"Regression" - supervised learning with numeric response.

as Education ↑ median & variability in Wage ↑



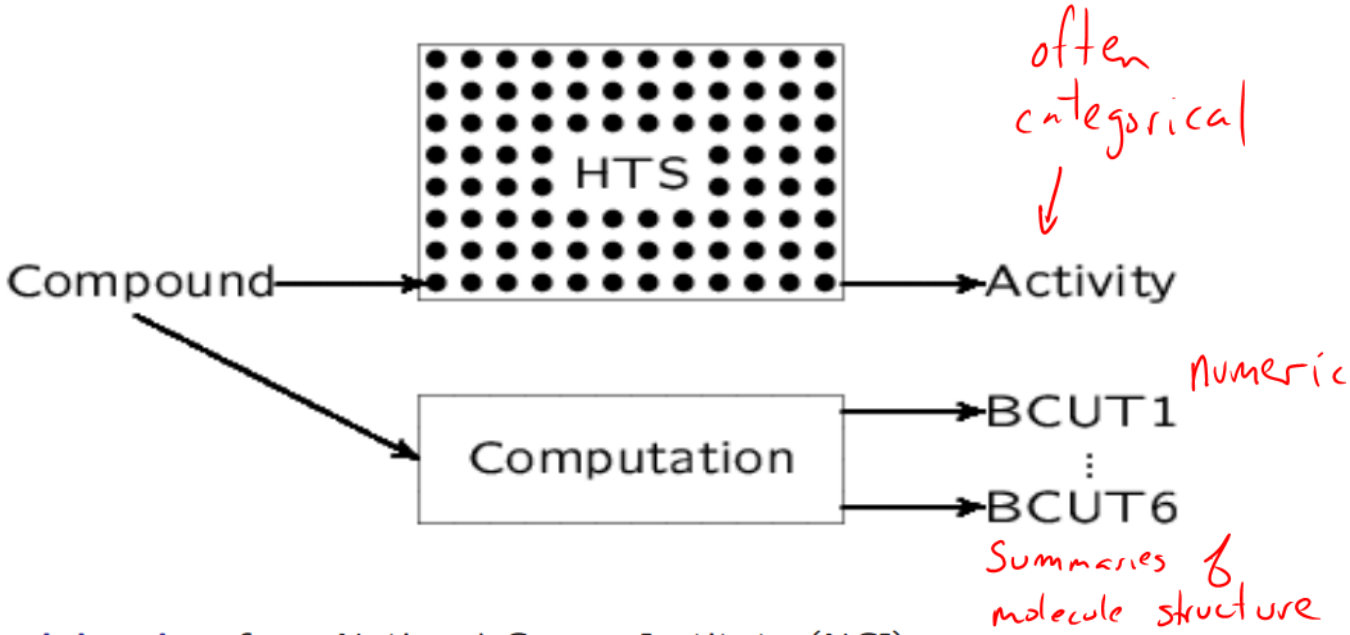

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

## 2. Drug Discovery

High throughput screening (HTS)

Response variable: Activity
(% inhibition, IC50 concentration, inactive/active)

Explanatory variables: Molecular descriptors (e.g., BCUT's)




**Example datasets** - from National Cancer Institute (NCI)

1. AIDS Antiviral Data (Inactive/Active)
   - Response: 0/1 inactive/active (active = highly active or mildly active)

| | Training data | Test data | Total |
|---|---|---|---|
| Active | 304 | 304 | 608 |
| Inactive | 14,602 | 14,602 | 29,204 |
| Total | 14,906 | 14,906 | 29,812 |

   - 6 BCUT descriptors
2. A similar dataset with a continuous response exists: -log(EC50). EC50 = compound concentration that protects infected cells by 50% (-log(EC50) is a larger-the-better response).

Classification/Regression modelling:

- Given a training set with activity and descriptors, construct a model to predict activity using the descriptors.
- This allows "virtual screening" of compounds - identify most likely actives without testing them all.

Both the wage data and the drug discovery problem are examples of **supervised learning**.
- We want to construct a model to predict a "response variable" or "output" Y, when given values of "inputs" (a vector X).
For example...

| Y | X |
|---|---|
| wage | Educ., age, ... |
| Activity | BCUT1, ..., BCUT6 |

- In supervised learning, we have a ``training set" where X and Y are observed for each object in the data.

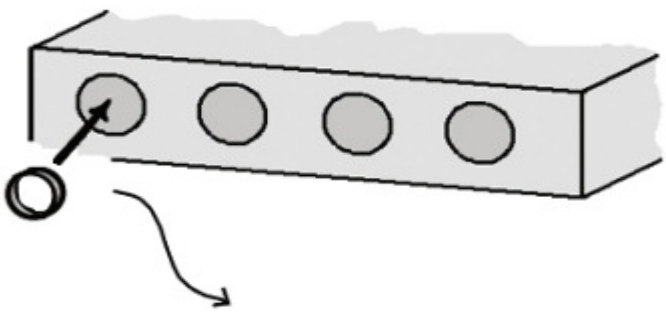
- Using the training data we estimate or learn a statistical model that can predict Y when given a new X.

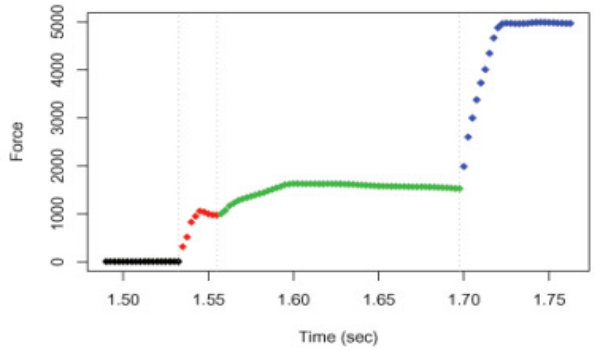
- In **unsupervised learning**, there is no output "Y". Instead we want to describe or summarize the X values.

**3. A manufacturing example (unsupervised learning)**

## Example - Valve Seat Insertion

- Source: Truck engine assembly plant.
- Steel valve seats force-fitted atop the cylinder head.
- V8 engines with 4 intake and 4 exhaust valve seats inserted simultaneously.
- Data: force exertion $f(t)$ observed over time.
- 6,000 cylinders; 41 days in Jan and Feb of 2000.

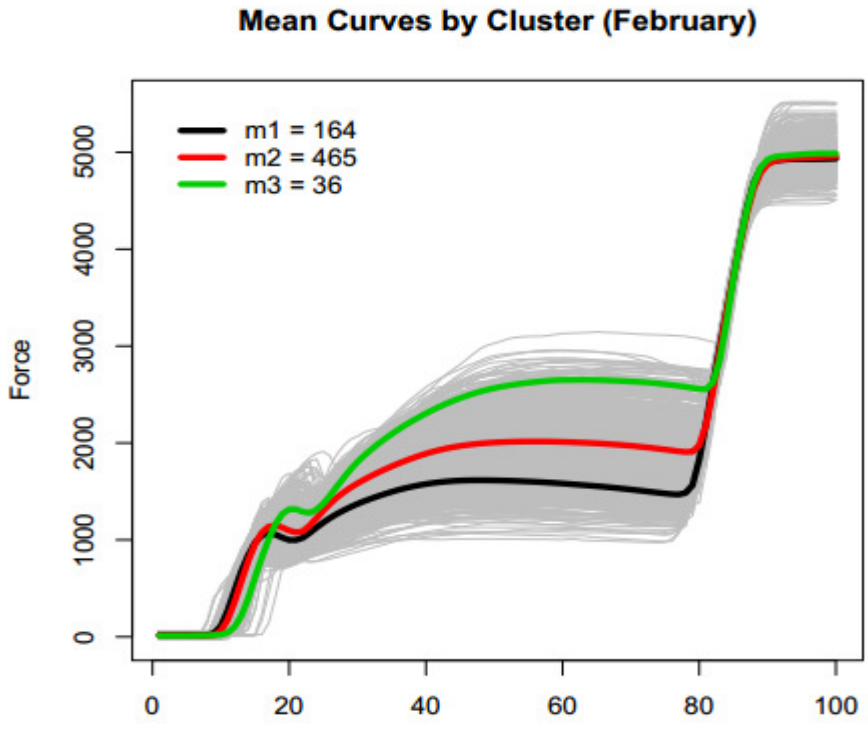




# Example - Valve Seat Insertion

- **Problem:** Some insertions are bad
  - If a part is badly inserted, it will fall out after 10,000 to 20,000 km and cause leakage around the seat.
  - We don't know which insertions are bad.
  - Would have to take the engine apart to know.
- **Goal:** Identify changes in the curves
  - Are any of the curves drastically different from the rest?
  - How does the process change over time?

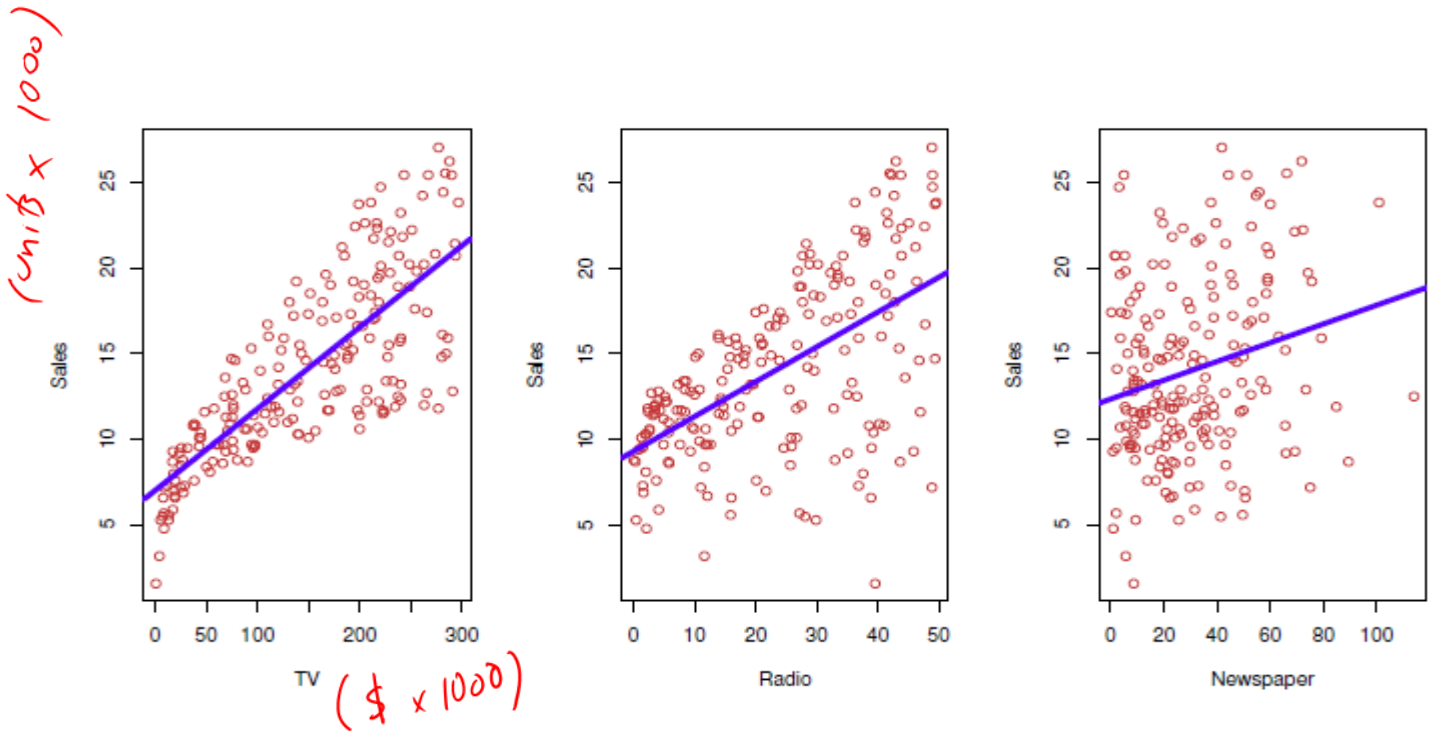
In general, unsupervised tasks include:
- Clustering, which is the identification of groups of similar objects (e.g. insertions)
- Dimension reduction, which is the identification of lower-dimensional representations of the data (here we have 100 measurements per curve, but the shape varies in many fewer important directions).
- We'll see unsupervised methods in week 4




Another unsupervised example is given in ISLR Ch 1, concerning gene expression data (NCI60 data)

**Chapter 2: Overview of supervised learning**

Motivating example from ISLR: Advertising expenditures and corresponding sales of a specific product.




Clearly increasing spending on advertising will increase sales.

a = intercept (sales @ X = 0)

b = slope (for every $1000 spent on TV, I sell b thousand more units)

The blue lines are 3 linear regressions, such as:

Sales = a + b (TV spending)

Can we fit a better model, using all three variables?

Sales = f(TV, Radio, Newspaper)

We want to learn " f " from available data:

$$\{(Y_i, X_{1i}, X_{2i}, X_{3i})\}_{i=1}^{n}$$

n = 200 observations

That is, we have n objects (here, markets) and for each object we observe response Y and predictors X1, X2, X3 (TV spend, Radio spend, Newspaper spend)

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$ = vector of predictors, (or features, or inputs)

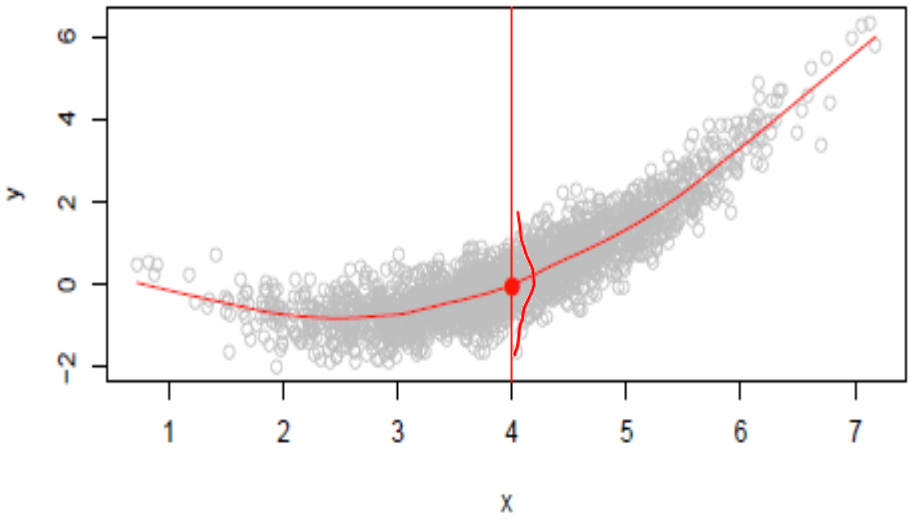
(generally of length p)

$Y = f(X) + \varepsilon$

$\varepsilon$: error or noise

f can be used to:

- make predictions of Y at specific X
- understand relationship between X & Y
  - increasing?
  - linear?
  - what variables are important




What is a good "f(x)"?

For example at x=4, what should f be?

A good value is

f(4) = E(Y | X=4)

E(Y|X=4) means the expected value (average) of Y given X=4

This is also defined for vector X, eg f(x) = f(x1,x2,x3) = E(Y|X1=x1,X2=x2, X3=x3)

This minimizes mean squared prediction error E[(Y - g(X))^2|X=x] over all functions g at all points X=x.

There's still irreducible error $Y = f(X) + \varepsilon$

For any estimate $\hat{f}(x)$ of $f(x)$ we have

Expected squared prediction error

$$E\left[(Y-\hat{f}(X))^2 \mid X=x\right] = \underbrace{\left[f(x)-\hat{f}(x)\right]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\varepsilon)}_{\text{Irreducible}}$$

(if we assume that $\hat{f}$ and $X$ are fixed)

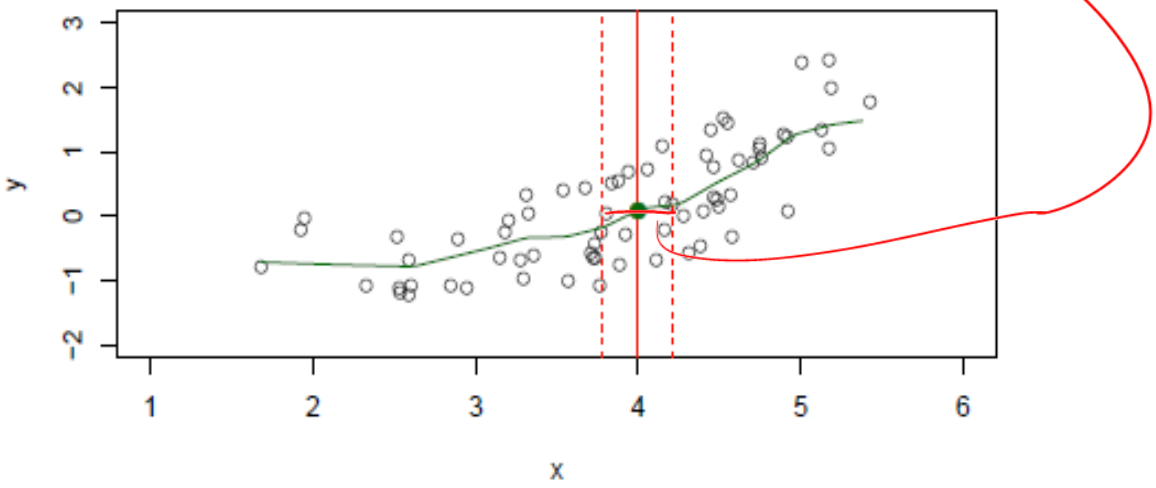
How to estimate f?

With a sample, there will only be a few points with X=4 exactly (or none).

So an exact estimate of E(Y|X=x) is not possible.

Relax the definition and let

$$\hat{f}(x) = \text{Ave}\left(Y \mid X \in \underbrace{N(x)}_{\text{neighbourhood of } x}\right)$$

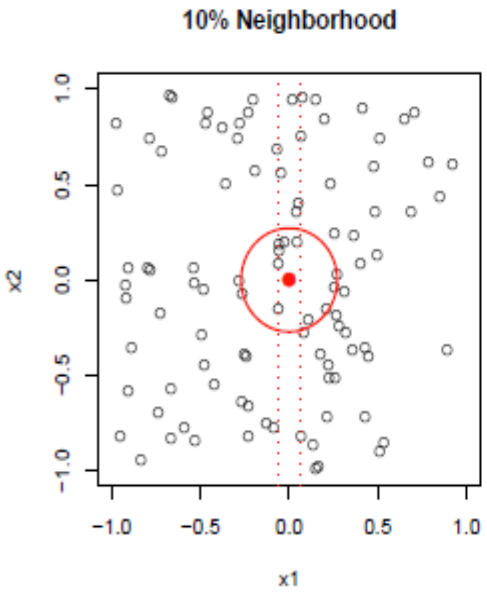


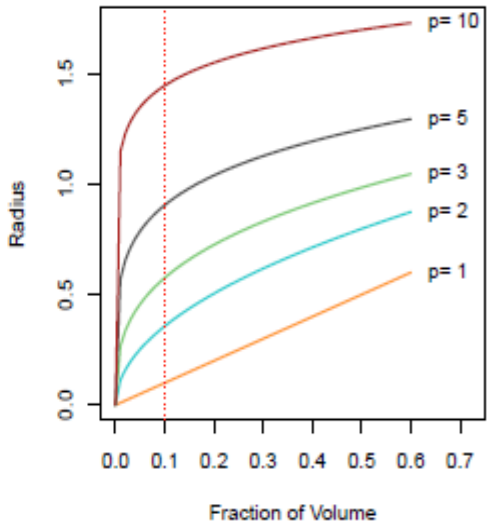

Nearest neighbour averaging:

k-nearest neighbours calculates the distance from x (here x=4) to all training points, and chooses the k nearest ones. The prediction at x=4 is the average of the Y values of these k neighbours.

KNN works well in low dimensions, p=1, 2, 3, 4 and large N. But in high dimensions it suffers the CURSE OF DIMENSIONALITY.

Curse of dimensionality: for data that is uniform on (-1,1) intervals along axes in p dimensions, what is the radius needed to contain 10% of the data values?




Parametric and structured models

This shortcoming of KNN leads us to consider parametric and structured models, which make much stronger assumptions about the relationship between X and Y.

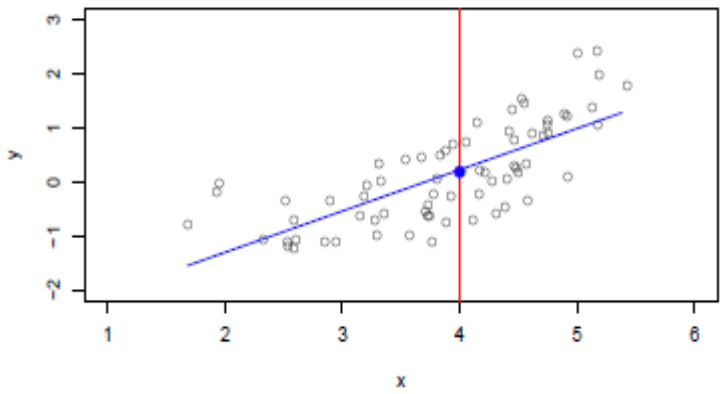
The linear model is an important case:

$$Y = \underbrace{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}_{f(x)} + \varepsilon$$

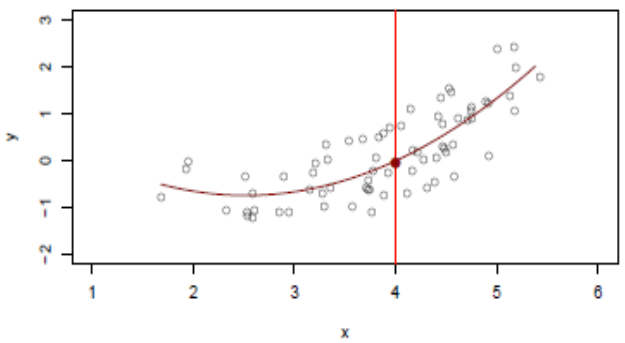
Coefficients estimated with training data (e.g. by least squares, see Ch 3)

Powerful and (sometimes) good approximation to the unknown true f(X).

linear $Y = \beta_0 + \beta_1 x$




quadratic $Y = \beta_0 + \beta_1 x + \beta_2 x^2$

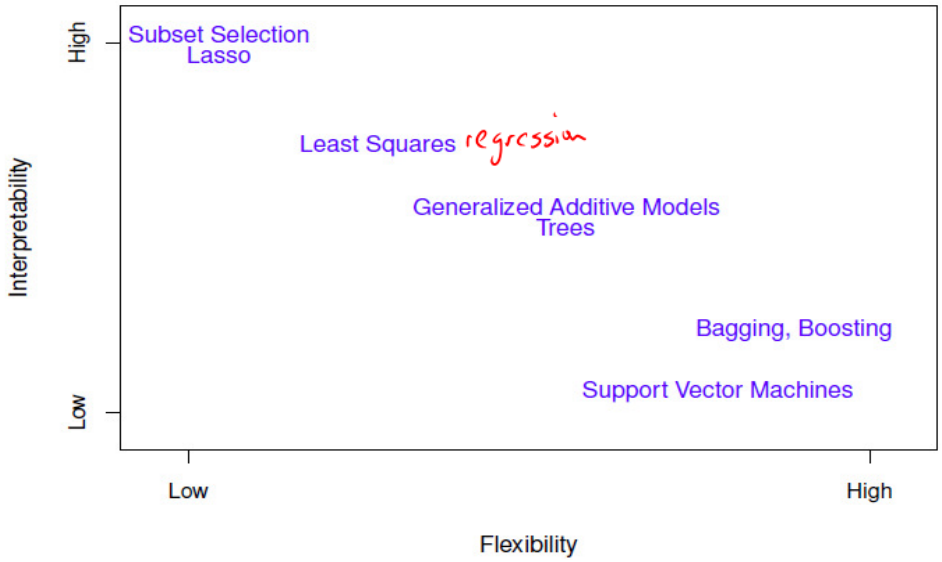


**Trade-off between simple and complex models:**

Many of our models will have an adjustable amount of flexibility.

Adjusting this flexibility will be a key trade-off we must make. It affects:
* Prediction accuracy
* Interpretability of the model

In general, the more flexible a method is, the more accurate and less interpretable it is
- This is true for different families of models (shown below) as well as within a family of models.




How do we choose how flexible to make our model?

Example (on separate pdf file: "02testexample.pdf"):

* Regression with a single X and a nonlinear function, with 20 "training" observations.
* We use a test set to measure accuracy of various fitted models.
* In this example our model is a polynomial regression with various different maximum powers:

$$y = \underbrace{\beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_p X^p}_{f(x)} + \varepsilon \quad \text{for } p = 1, 2, \ldots, 10$$

< separate pdf "02testexample-annotated.pdf" >

Suppose we fit a model $f(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

- We could compute the average squared prediction error over Tr:

$$\mathrm{MSE}_{\mathsf{Tr}} = \mathrm{Ave}_{i \in \mathsf{Tr}}[y_i - \hat{f}(x_i)]^2$$

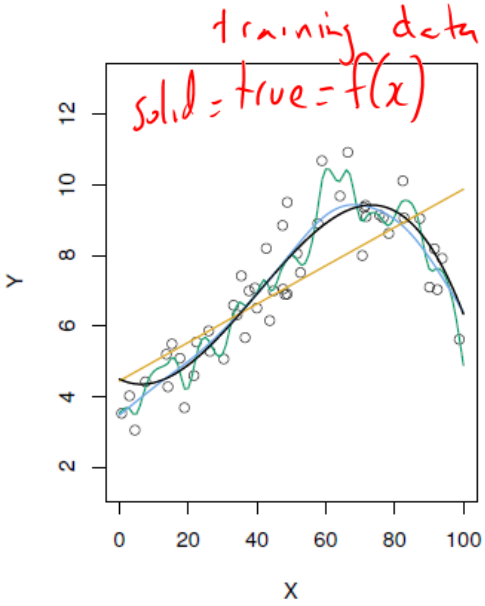
This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\mathsf{Te} = \{x_i, y_i\}_1^M$: ← new data, not same as training set.

$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}}[y_i - \hat{f}(x_i)]^2$$

MSE for $\hat{f}(x) = f(x)$ (the best possible MSE)

Ch 3 gives three more train / test examples, with varying
- complexity (i.e. nonlinearity of function)
- noise level



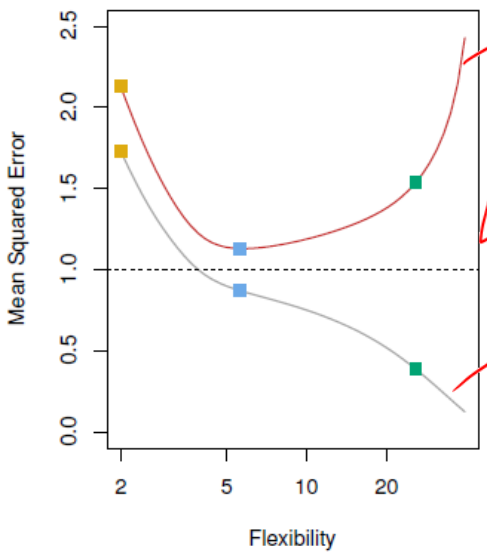
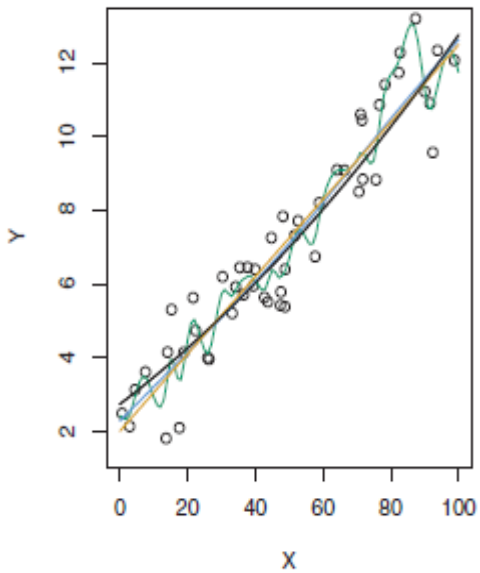




Fig 2.9:
- Nonlinear function
- Medium noise level



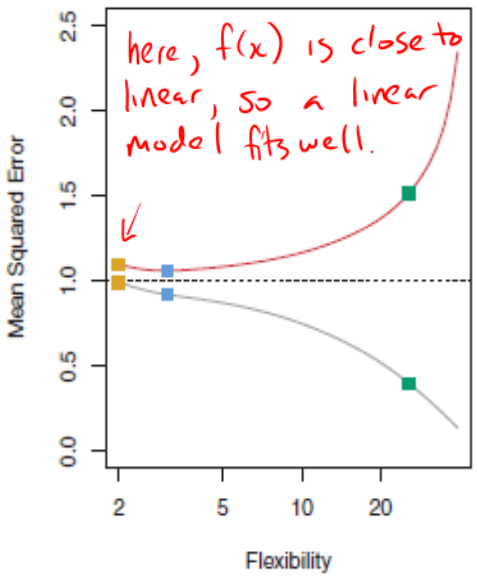
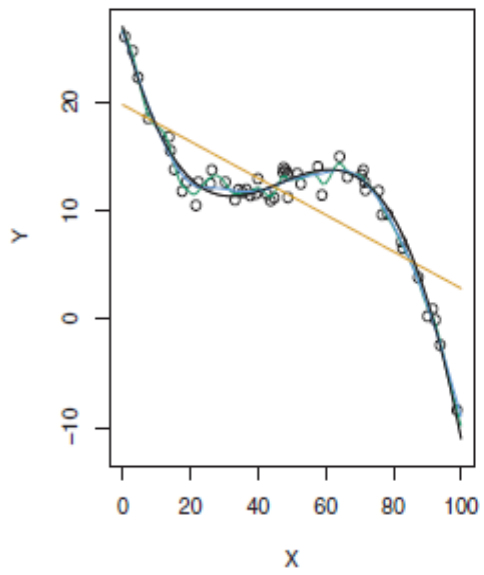




Fig 2.10:
- close to linear function
- medium noise level



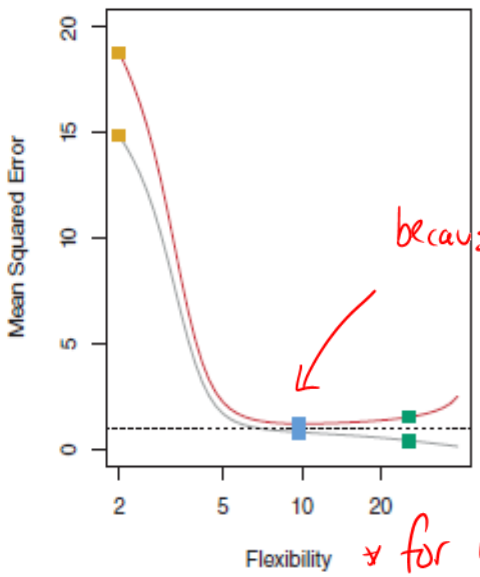




Fig 2.11:
- very nonlinear function
- low noise level

because of a complicated model such as poly with 10th degree fits well.

* for large training sets, we can fit more complex models

We can see in the different examples that the the appropriate level of flexibiliy will change for different problems.

In Sec 2.2.2, the expected test MSE is decomposed (without proof) as

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

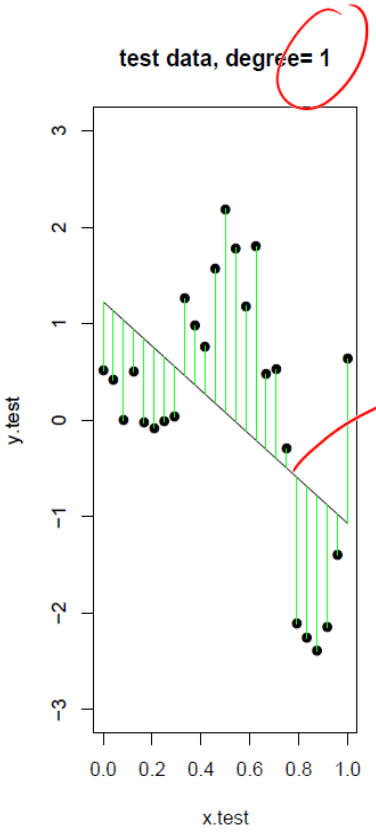
$$Bias\left(\hat{f}(x_0)\right) = E\left(\hat{f}(x_0)\right) - f(x_0)$$

$y_0$: random y at $x_0$

$x_0$: test point

$\hat{f}$: estimated f for random training set

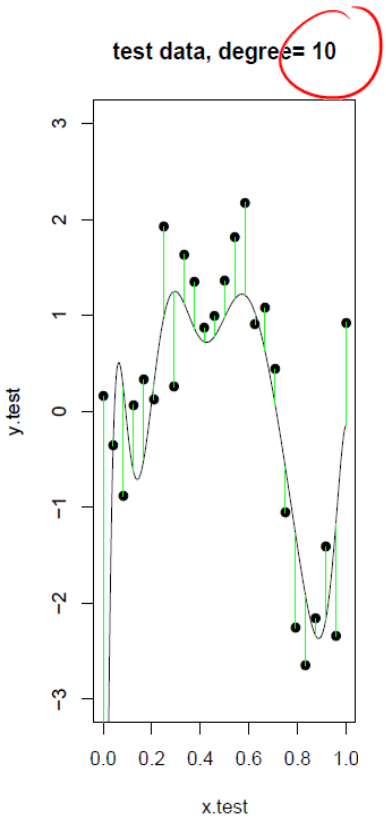
Here, we're taking E() & Var() over all possible y, and over all poss. training sets.




Linear model has high bias - systematic difference between linear prediction and nonlinear function. It has low variance because it will give the same (biased) answer for different training sets.

$\hat{f}(x)$

10th degree polynomial has low bias, but high variability ("wiggle"). You can think of variability as sensitivity to perturbations in training data.

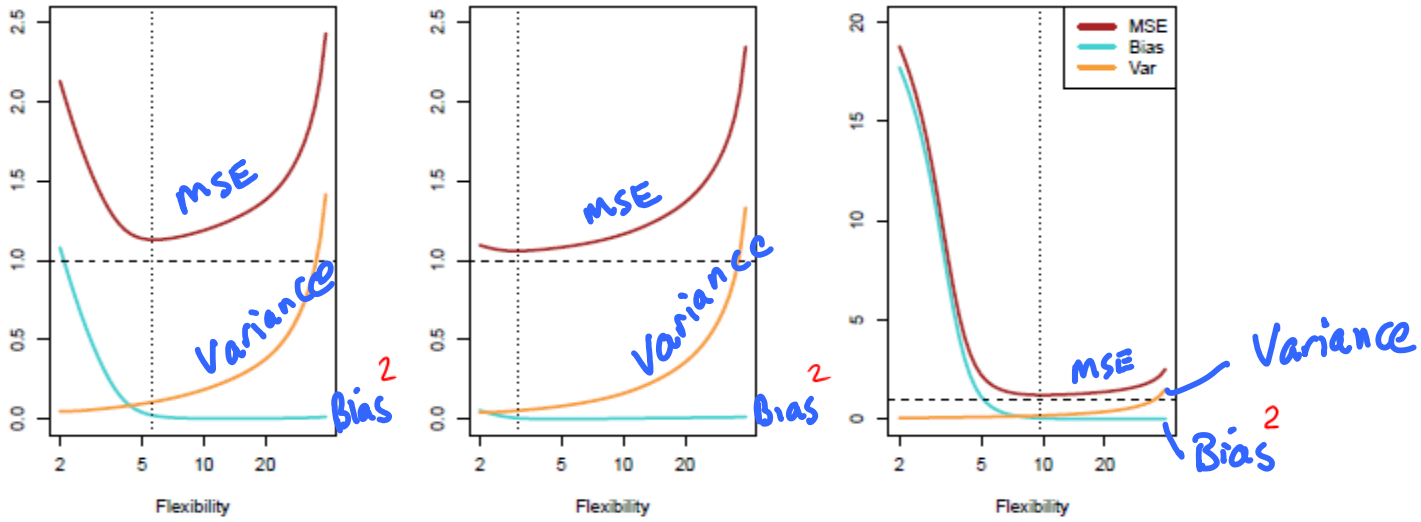



## Bias-variance trade-off for the three examples

- nonlinear
- medium noise

- linear
- medium noise

- very nonlinear
- low noise



**Classification problems**
So far, we've only discussed a numeric response ("regression").

Many of the ideas above are similar for classification, in which we want to predict a categorical response.

Examples: Drug discovery - classify compounds as active / inactive.

$\mathcal{C}$ = set of possible classes (eg $\mathcal{C}$ = {active, inactive})

or $\mathcal{C}$ = {0, 1, 2, ..., 9} in handwritten digit recognition.

like f(x) in regression

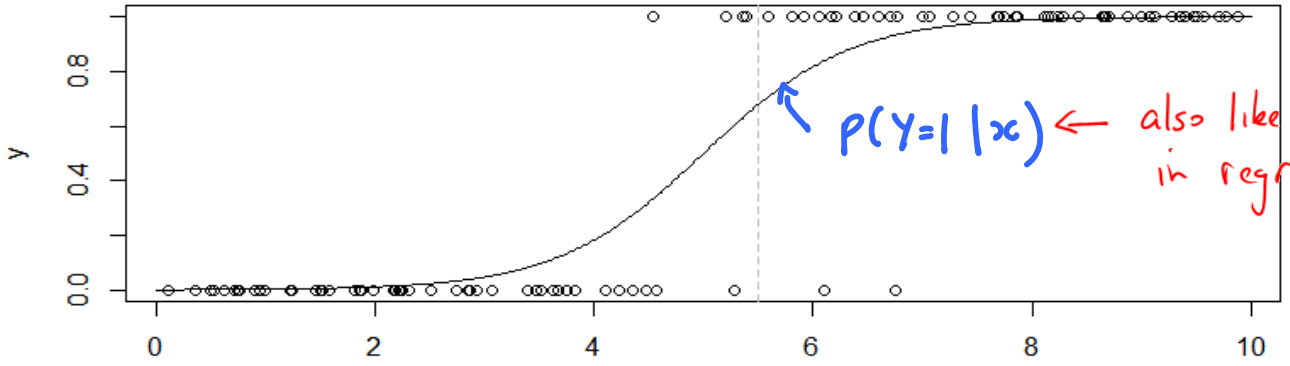
We want to learn a function C(X) that maps X=(X1, ..., Xp) to class labels.

We may also want to:
- asssess uncertainty in our classification
- discover the way in which different X's affect C(X).

Example (not in book): Dose-response experiment

X = amount of toxin $\mathcal{C}$ = {alive (0), dead (1)}

C(5.5) = ? How would we estimate C(5.5)?

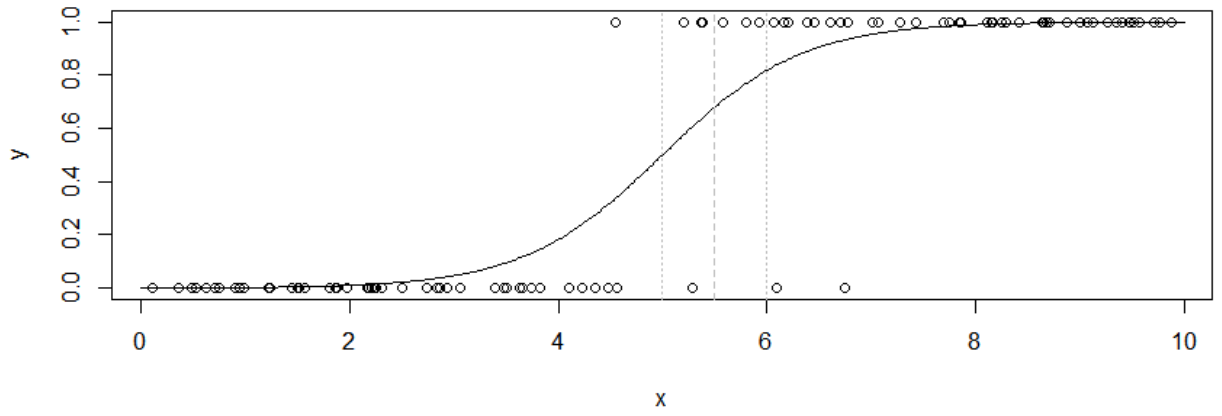



$P(Y=1|x)$ ← also like f(x) in regression

7 nearest neighbours, $\frac{6}{7}$ are dead

$\hat{C}(5.5)$ = "dead"

$\hat{P}(5.5) = \frac{6}{7}$



In the (unrealistic) case where we know P(Y=1 | X=x), the best ("Bayes optimal") classifier is to choose class 1 ("dead") whenever P(Y=1 | X=x) is > 0.5

More generally, for K classes, let

$$p_k(x) = Pr(Y = k \mid X = x), \quad k = 1, 2, \ldots, K$$

$$C(x) = j \text{ if } p_j(x) = \max(p_1(x), p_2(x), \ldots, p_K(x))$$

"pick the class with highest prob. $p_j(x)$"

Instead of MSE, we typically measure performance with the misclassification erorr rate (ideally on at test set "Te") :

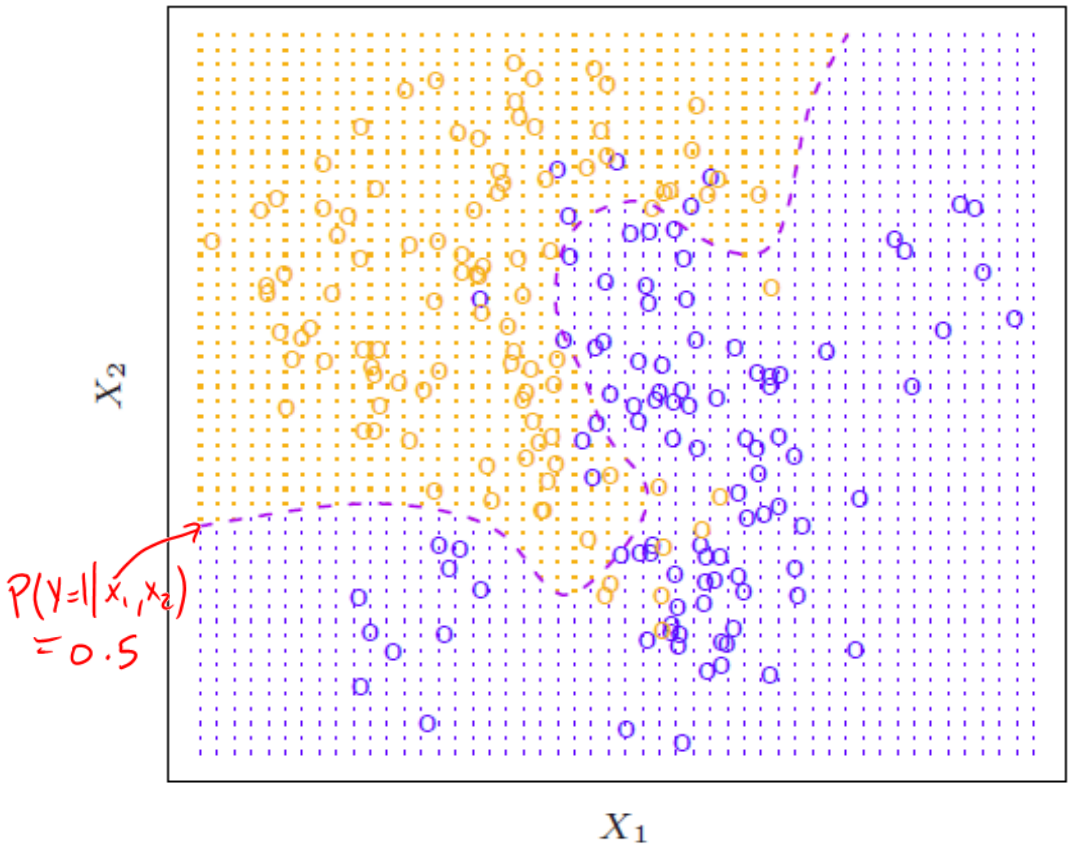
$$\text{Err}_{\text{Te}} = \text{Ave}_{i \in \text{Te}} I[y_i \neq \hat{C}(x_i)]$$

$I[\cdot] = \begin{cases} 1 & \text{if TRUE} \\ 0 & \text{otherwise} \end{cases}$

observed $y$ (ie class) is predicted wrong by $\hat{C}$.

We'll see (in Ch 4) both parametric models (logistic regression) and flexible methods (e.g. k-nearest neighbours) for clasification.

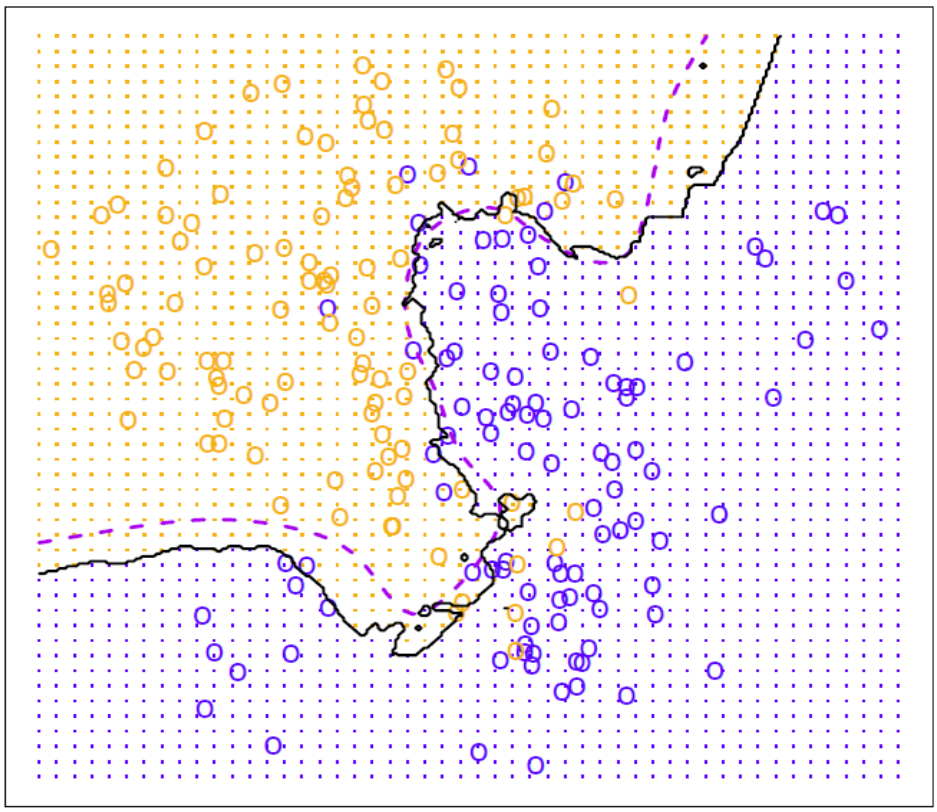
Here we show an example in two dimensions with a complex decision boundary and noisy data.



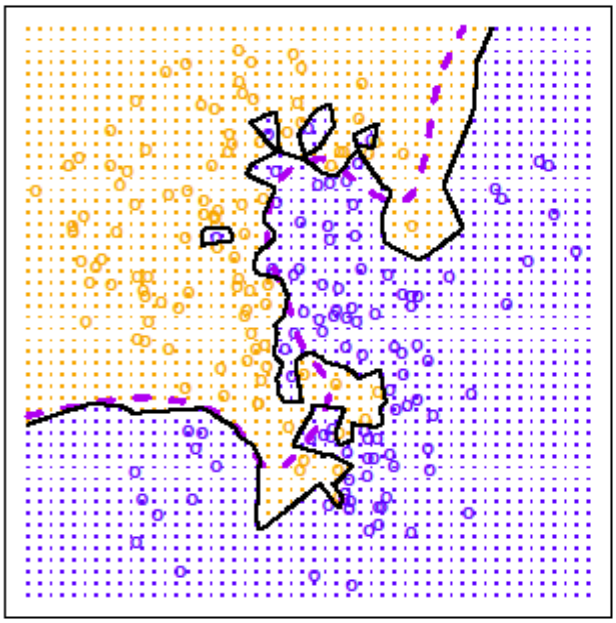

Example:

* 2 classes (orange / blue)
* true (and unknown) decision boundary is dashed line
* Labels are observed with error (or randomness)

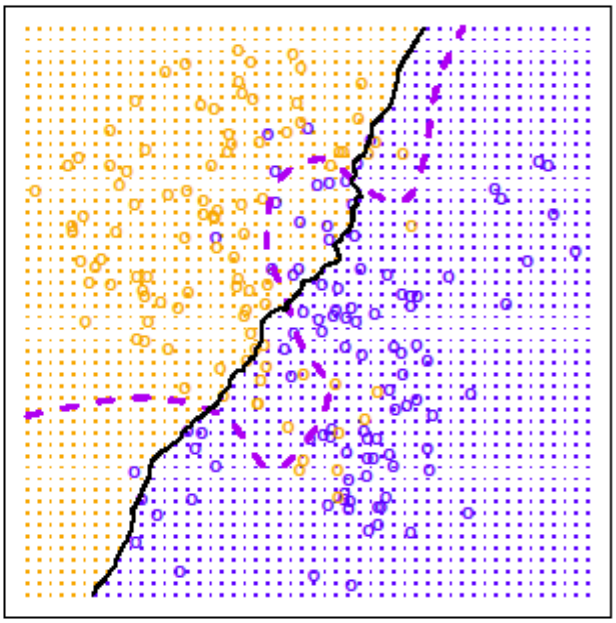
KNN: K=10




KNN: K=1

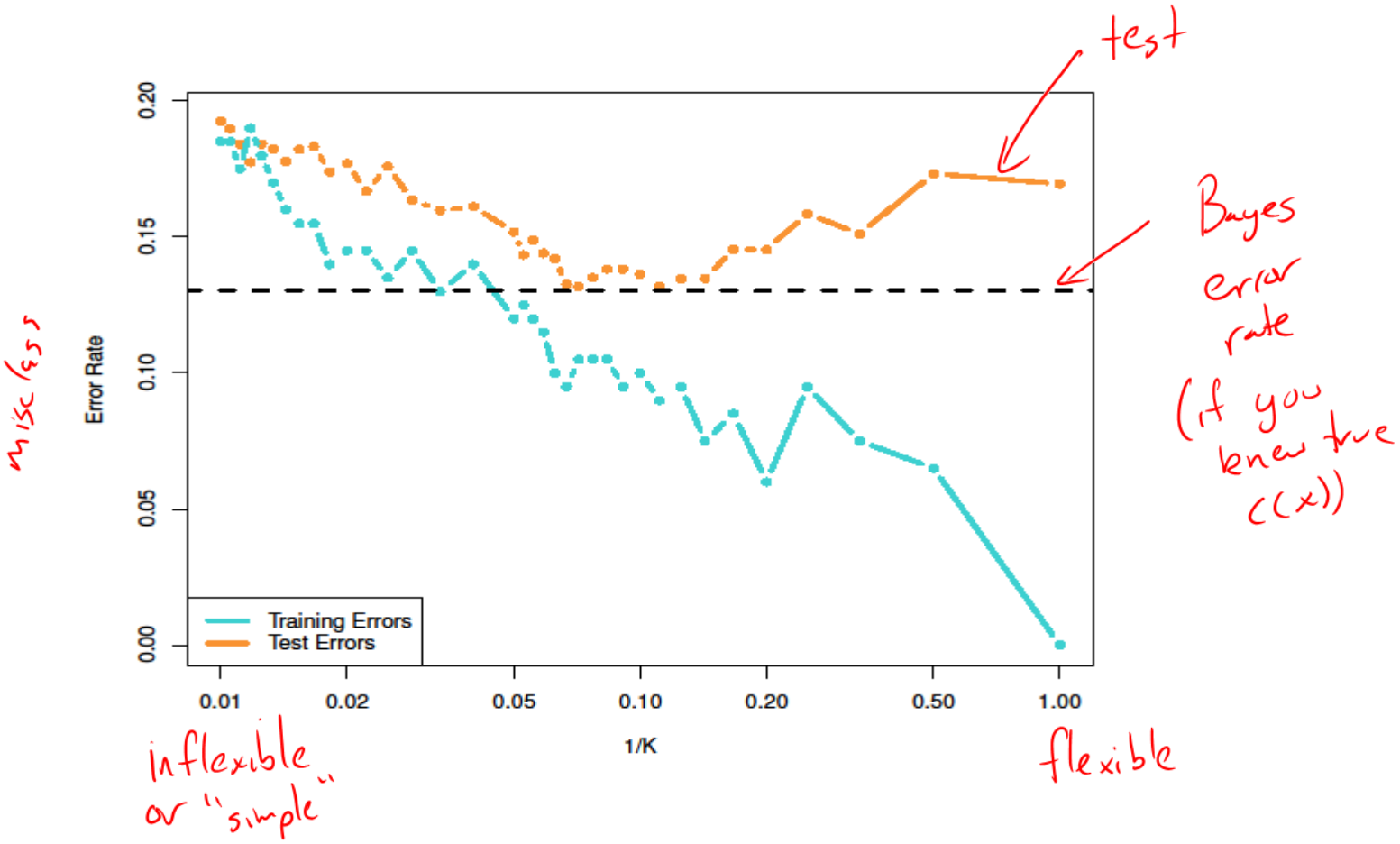


KNN: K=100



high variance
low bias

low variance (insensitive to small perturb.)
high bias ("wrong" boundary)




As with regression problems, we see that the flexibility of the estimator can be varied to control the bias/variance trade-off.

Note the horizontal axis of the above graph. Why is 1/k (k = number of neighbours) plotted instead of k?