# Assessing Models

UBCO MDS — DATA 570

# Assessing Models


- There is no one-size-fits-all *best* model.
- This module aims to introduce you a small subset of the statistical modelling choices there are to choose from.
- Selecting the best approach *for a particular problem* can be one of the most challenging tasks for a modern data scientist.

# Assessing Models


- A natural way to evaluate the quality of our model is to measure how well it predicts the response (i.e. the variable we are wanting to predict) in our observed data.
- Our response variable will generally be one of two forms: categorical or numeric.
- So how will we assess the performance of our models?

# Mean Squared Error


- With a numeric response, we can look at the mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where $\hat{f}(x_i)$ is the prediction for the $i$th observation, and $y_i$ is the response actually observed.

- MSE is small/large when predicted values are close/far to the true response

- Question: Can anyone see a problem with this definition?

# Mean Squared Error


- Sometimes we have access to test data that we keep separate from the training data we used to fit our model
- In other words, we might find our $\hat{f}$ based on training data $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and see how close $\hat{f}(x_{n+1})$ predicts $y_{n+1}$, where $(x_{n+1}, y_{n+1})$ is a *test* observation not used to train the statistical learning method.

# Mean Squared Error


- When MSE is calculated on our training data $\text{Tr} = \{x_i, y_i\}_1^n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ we call it the *training MSE* or $MSE_{Tr}$
- When MSE is calculated on our test data $\text{Te} = \{x_i, y_i\}_1^m = \{(x_{n+1}, y_{n+1}), (x_{n+2}, y_{n+2}), \ldots, (x_m, y_m)$ we call it the *test MSE* or $MSE_{Te}$.
- We want to choose the method that gives the lowest test MSE, as opposed to the lowest training MSE.

# Mean Squared Error


- In cases where test data is not available, we might be tempted to choose the method that gives us the smallest training MSE instead.
- There is no guarantee, however, that the method with the lowest training MSE will correspond to the method having the lowest test MSE.
- We want to choose the method that gives the lowest test MSE, as opposed to the lowest training MSE.

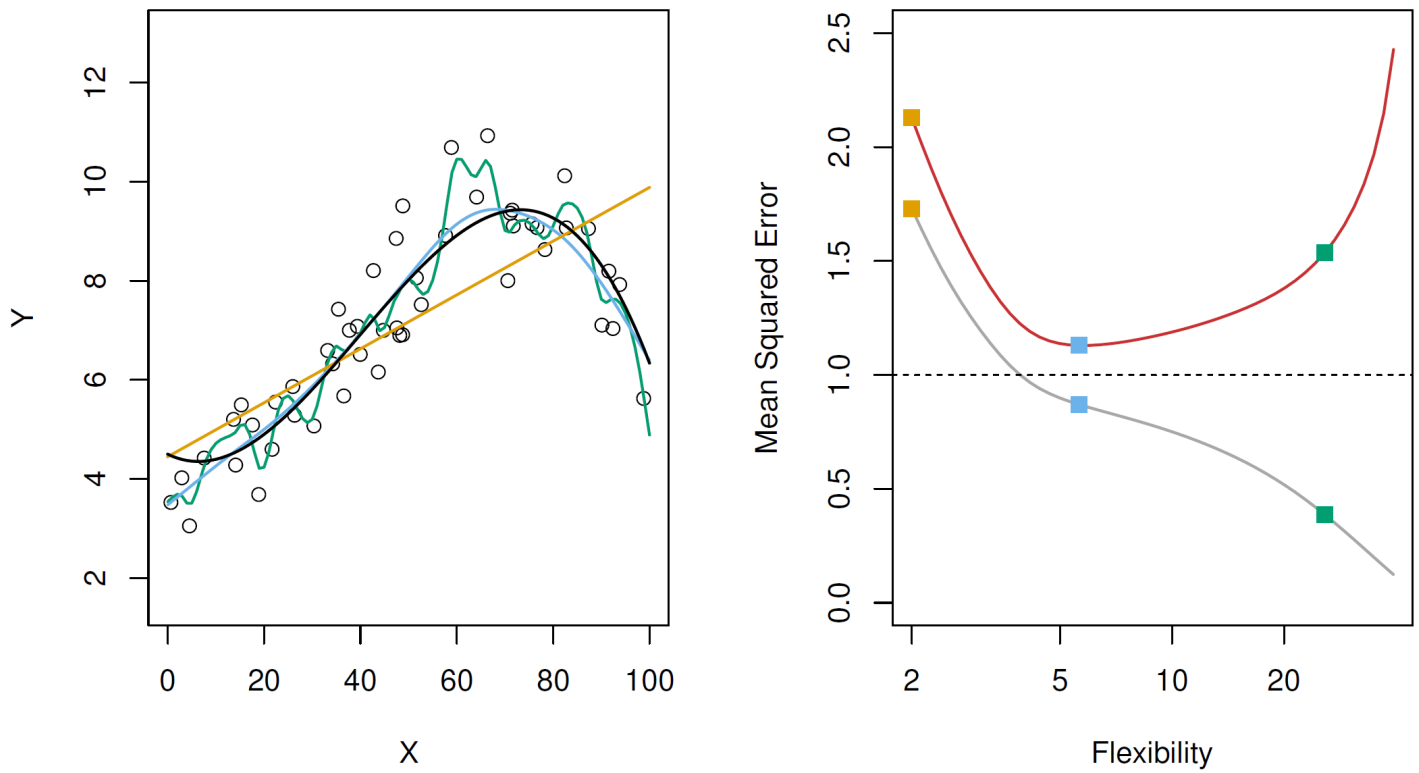



Figure: ISLR Fig 2.9. Left: Data simulated from $f$, shown in black. Three estimates of $f$ are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

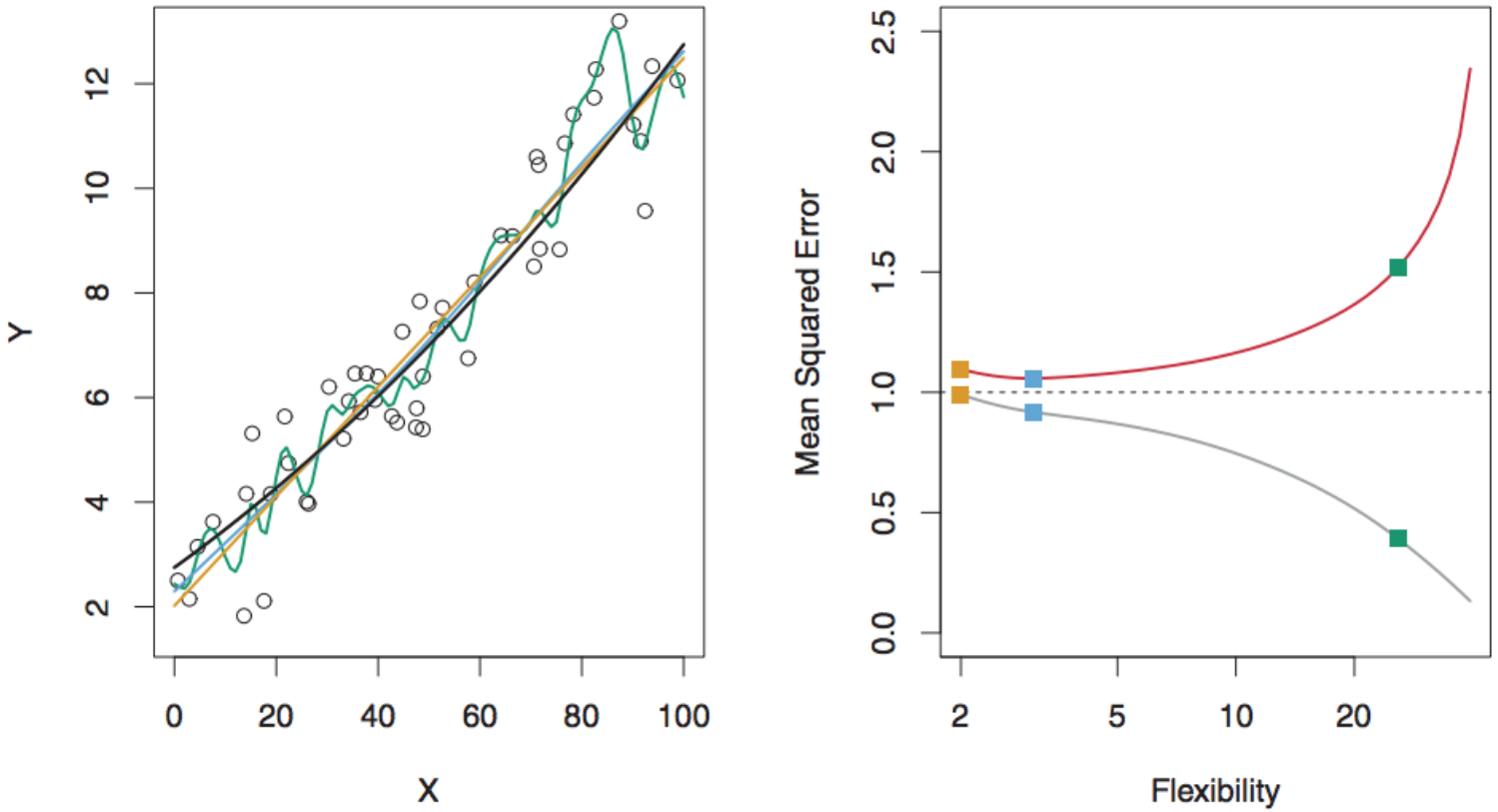



Figure: ISLR Figure 2.10

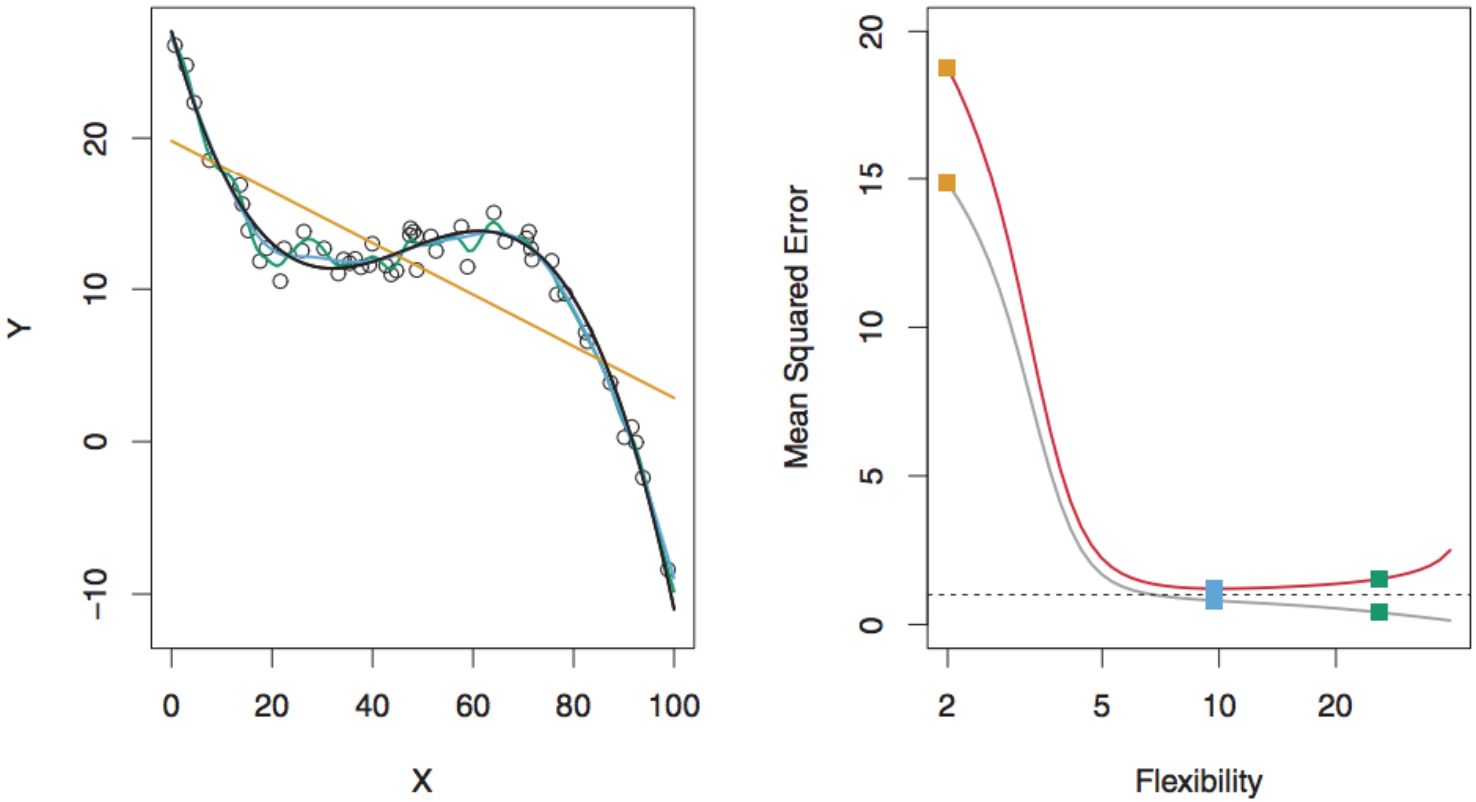



Figure: ISLR Figure 2.11

# Mean Squared Error


- Adding flexibility will always lead to a decrease in the training *MSE*...but not necessarily the testing *MSE*.
- Method that yeild a small training MSE but a large test MSE as said to be overfitting, and is a serious concern in statistical learning.
- Q: What's a simple relationship that we can usually expect between the training and testing *MSE*s, regardless of overfit?

# Bias and Variance


- Note that for a given test value $x_0$ the expected test *MSE* can be written as

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + Bias(\hat{f}(x_0))^2 + Var(\epsilon)$$

- In words, the expected test MSE is equal to the sum of the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$, and the variance of the error terms ($\epsilon$)

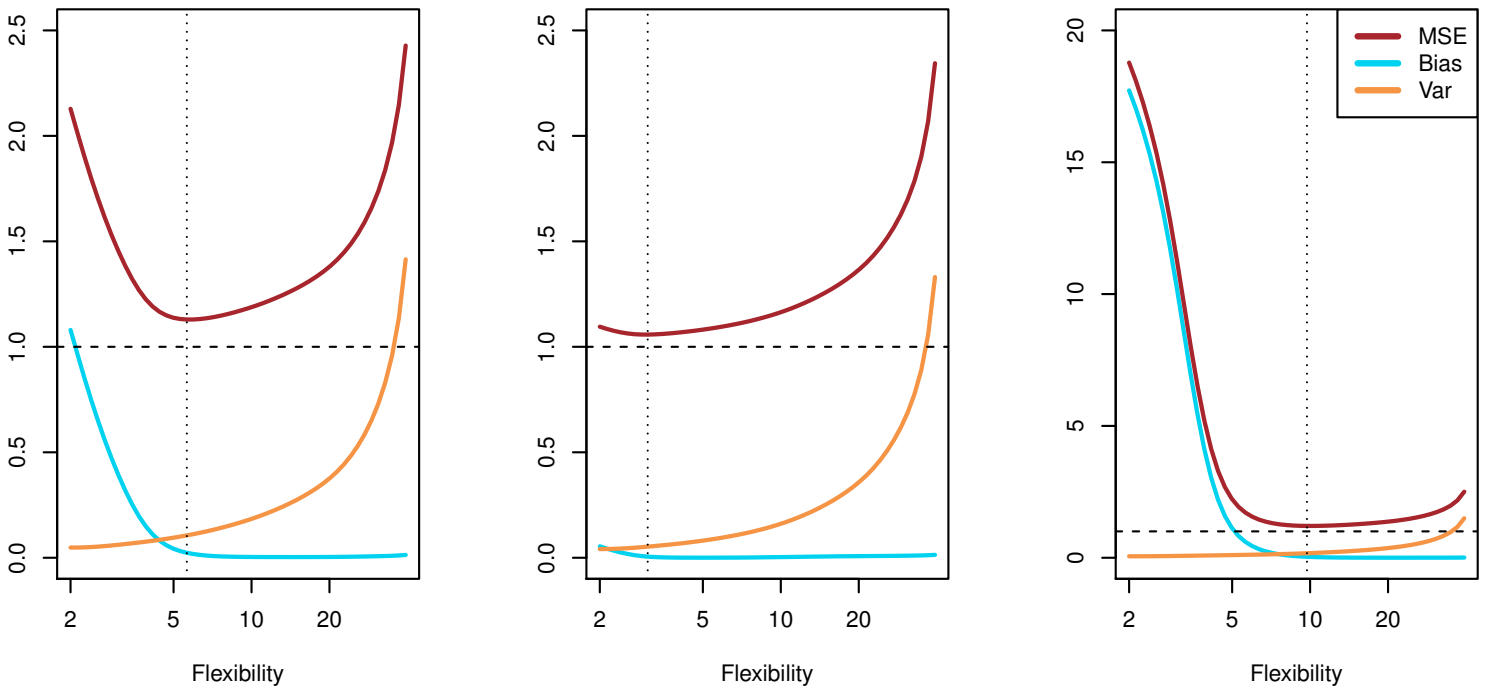
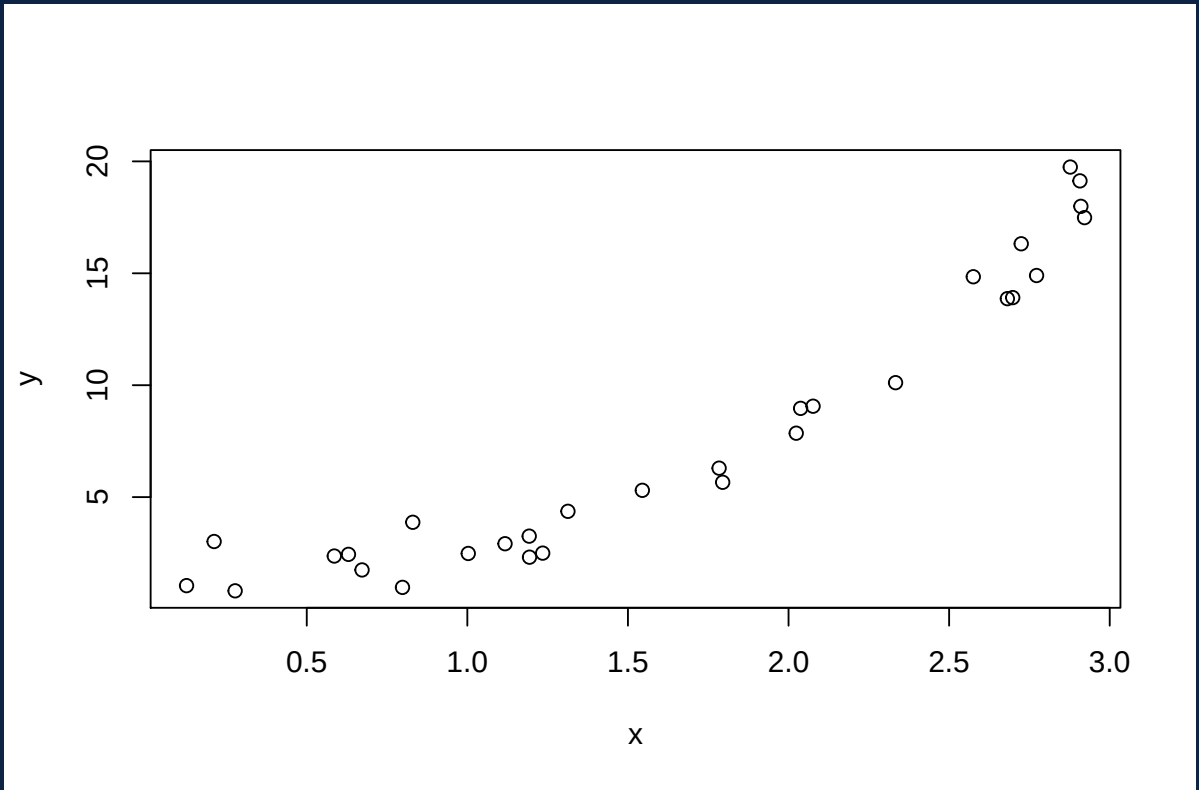
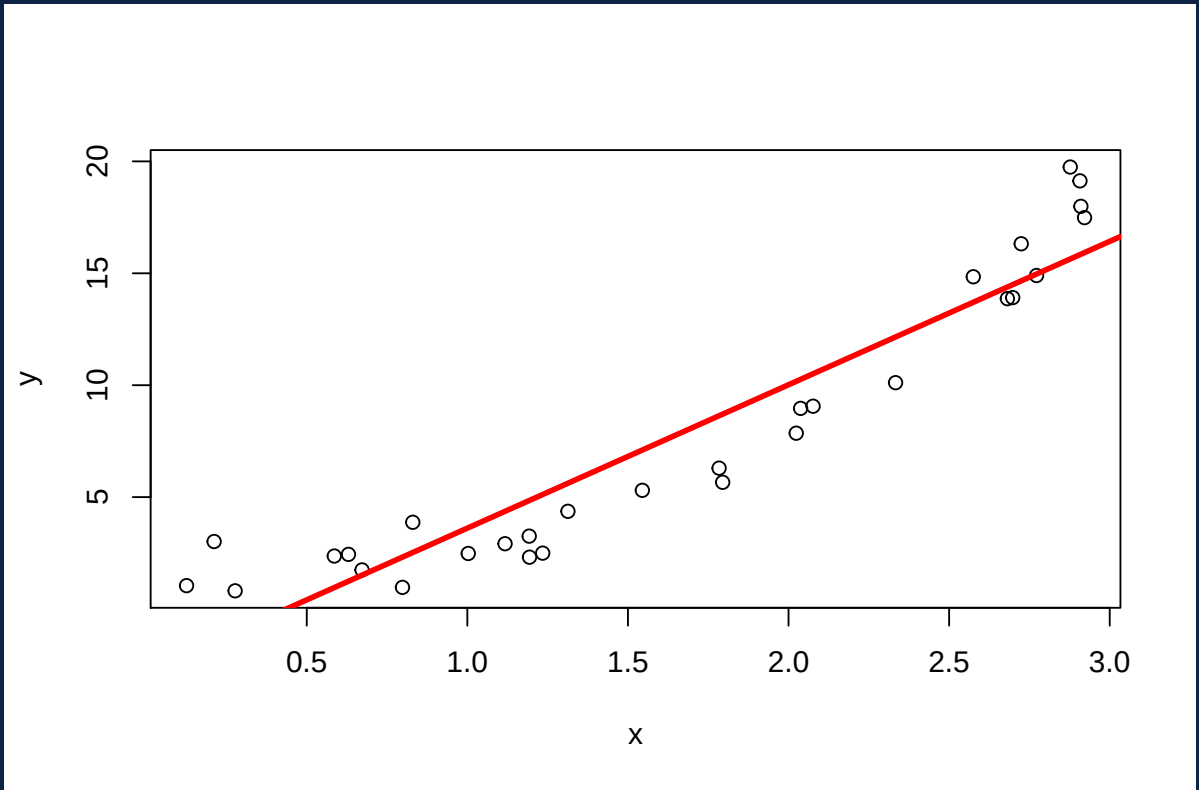



Figure: ISLR Fig 2.12. Squared bias (blue curve), variance (orange curve), Var($\epsilon$) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.

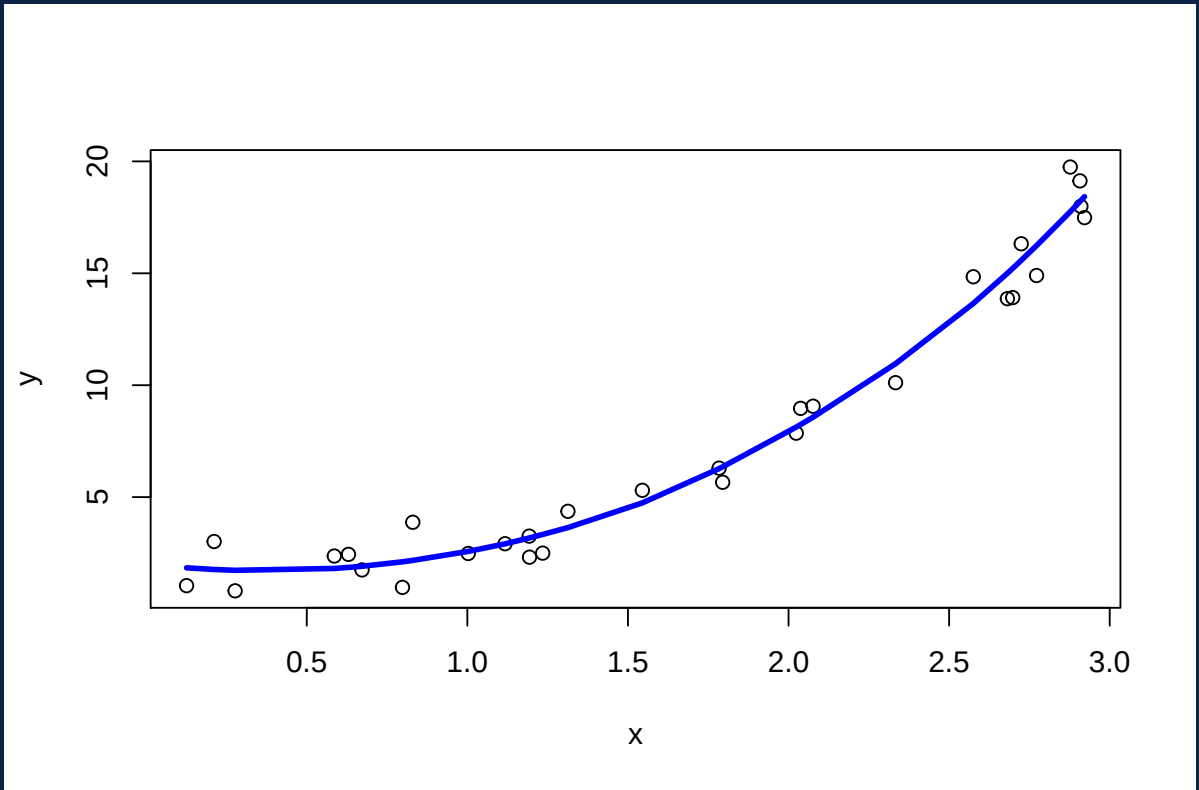
# Bias and Variance: Example



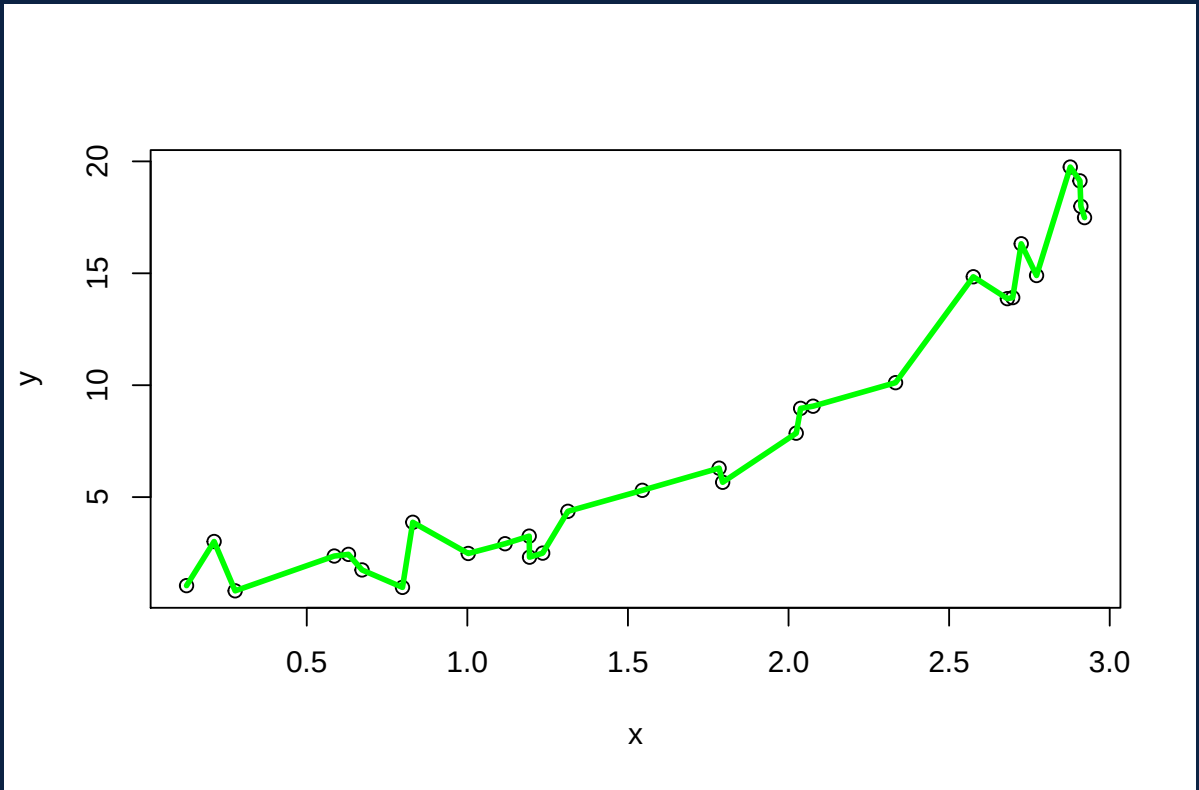
# Bias and Variance: Example



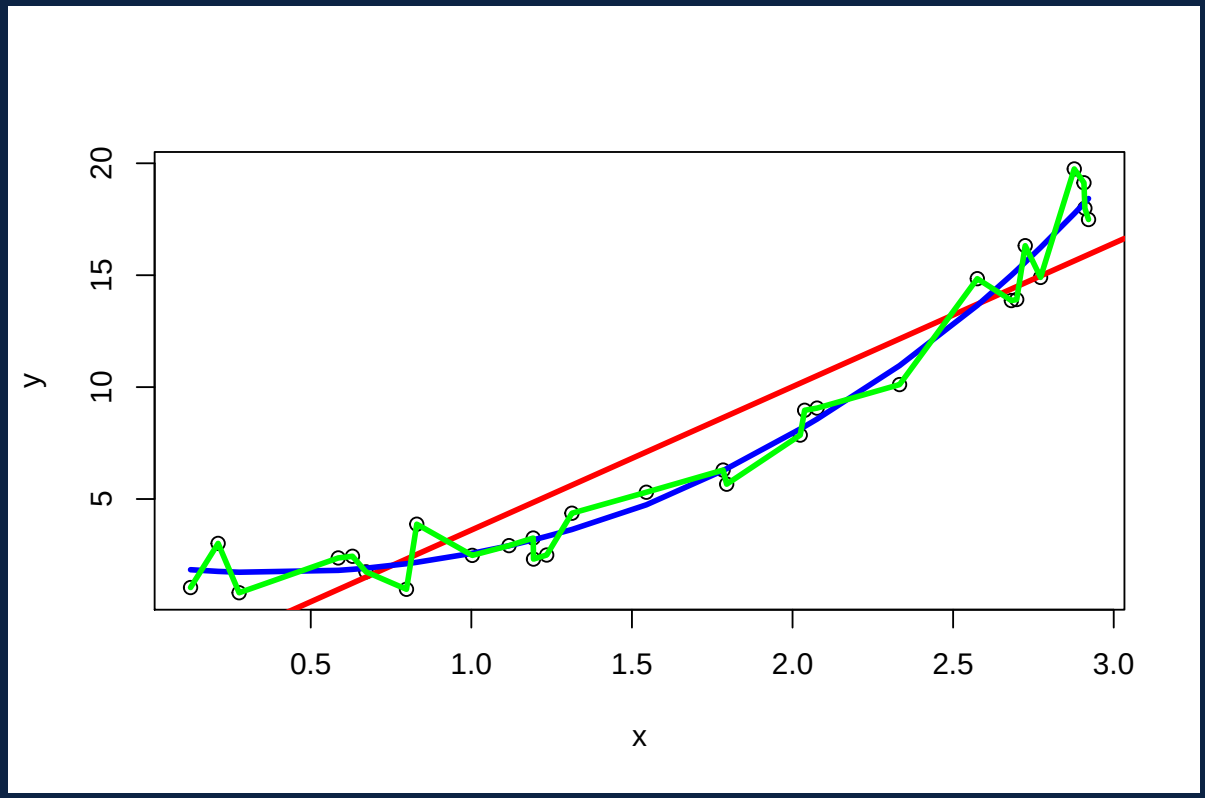
# Bias and Variance: Example



# Bias and Variance: Example

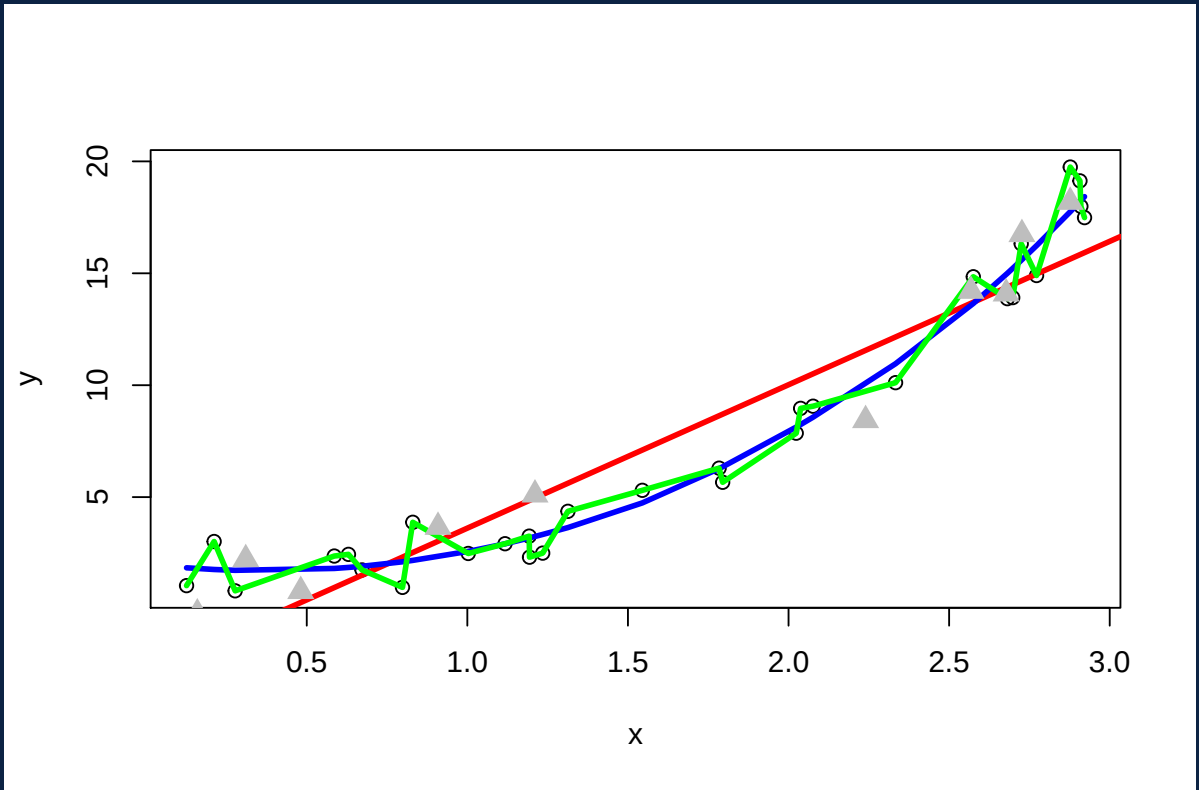


# Bias and Variance: Example

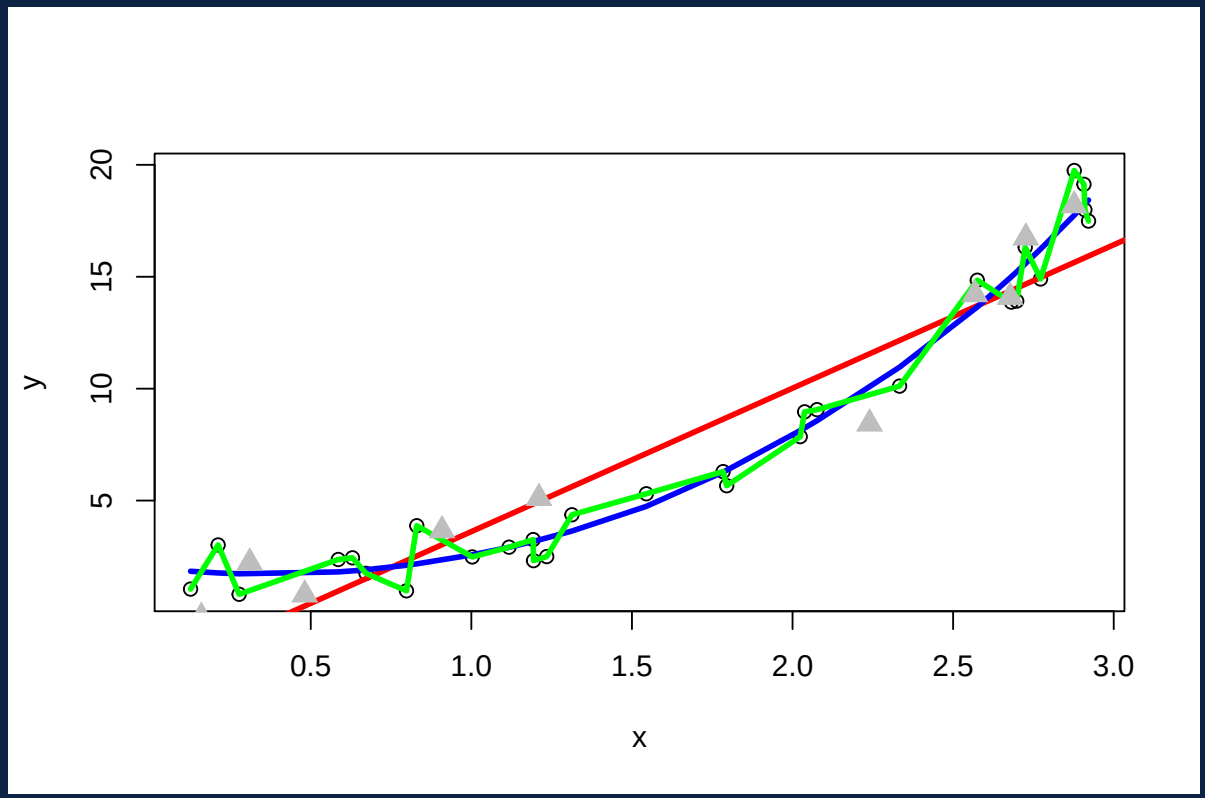



- Training MSEs:
  - Red: 4.37
  - Blue: 0.83
  - Green: 0.00

# Bias and Variance: Example



# Bias and Variance: Example




- Testing MSEs: (vs. Training MSE)
  - Red: 3.32 (vs 4.37)
  - Blue: 1.56 (vs 0.83)
  - Green: 4.29 (vs 0.00)

# Classification

- When $Y$ is a categorical variable, $MSE$ is not appropriate
- Instead, we consider the error rate

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

- Akin to the last topic, we are generally interested with the testing error rate rather than the training error rate.

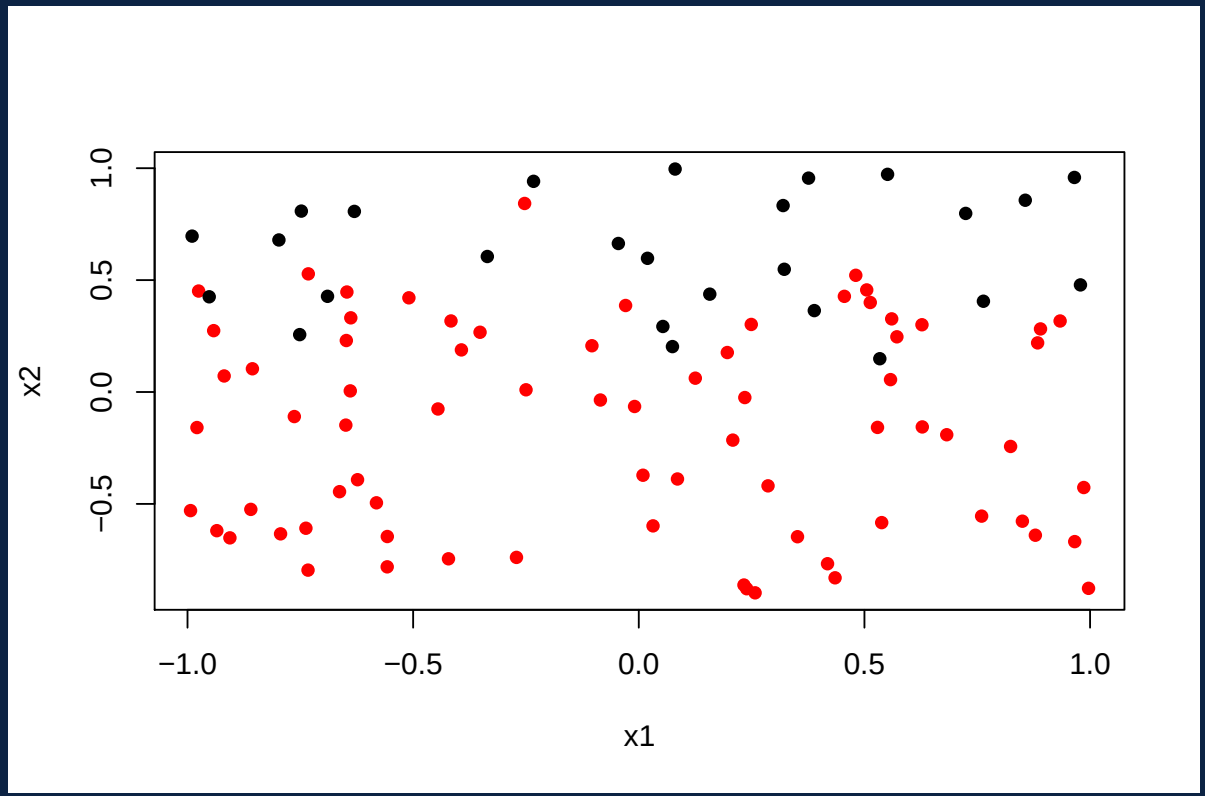
# Bayes Classifier


- The Bayes Classifier minimizes the testing error.
- It assigns each observation to the most likely class, given its predictor values.
- That is, for some predictor vector $x_0 = (x_{01}, x_{02}, \ldots, x_{0p})$ we should assign the class $j$ where the conditional probability

$$P(Y = j \mid X = x_0)$$

is maximized.

# Bayes Classifier: Example



# Bayes Classifier: Example


- For a two-group problem, this corresponds to predicting class one if $P(Y = 1 \mid X = x_0) > 0.5$.
- Are in which the probability is exactly 50% is called the Bayes decision boundary.
- Since this data was simulated, we know that the $P(Y = 1 \mid X = 0.5) > 0.5$.
- Decisions are made based on the Bayes decision boundary.

# Bayes Classifier


- The Bayes classifier gives the lowest possible test error rate (Bayes error rate), which is

$$1 - E\left(\max_j P(Y = j \mid X)\right)$$

- In the context of our notation, the Bayes classifier is the $f$ that we are attempting to estimate, and the error rate of that classifier is irreducible error.

- Question: what is the issue with this classifier for non-simulated data?

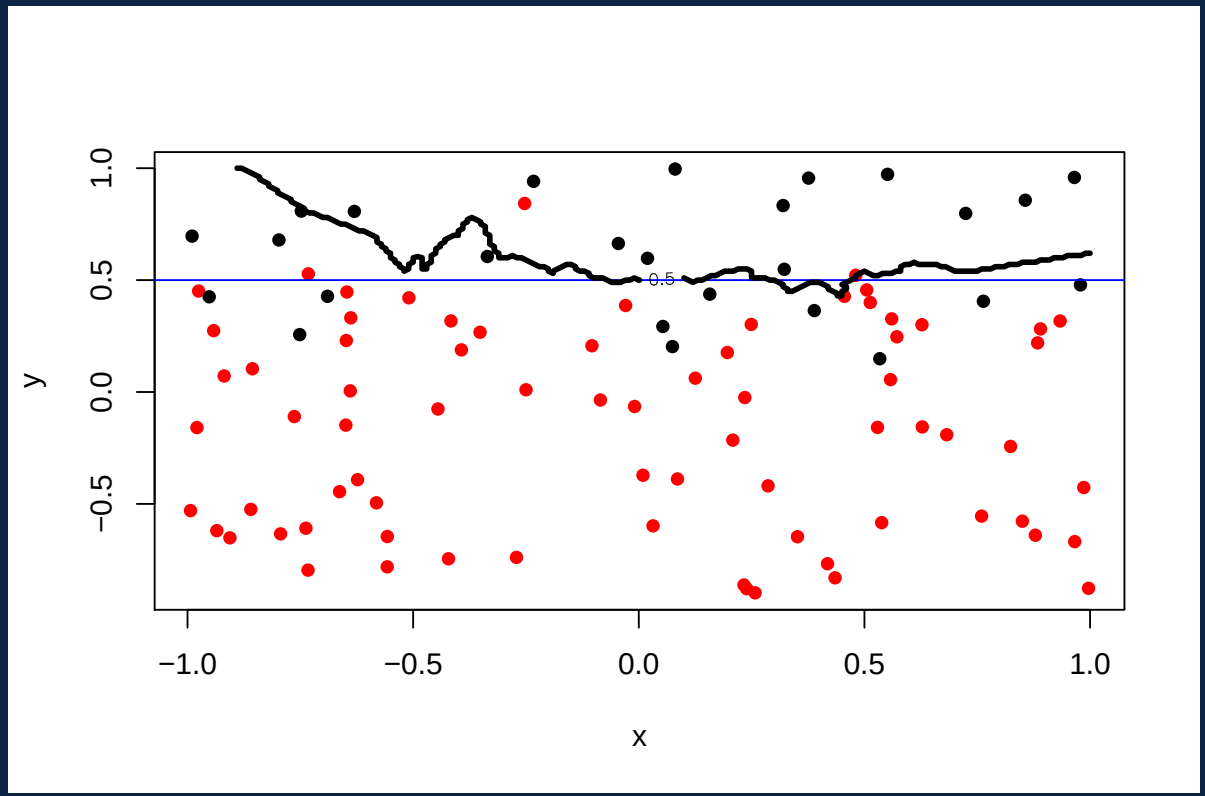
# K-Nearest Neighbours

- Let's look at a classifier which attempts to approximate the Bayes classifier: K-nearest neighbours (KNN).
- Given positive integer $K$ and test observation $x_0$:
  1. Identify $K$ closest points to $x_0$ in training data. Call this set $N_0$
  2. For each class $j$, find

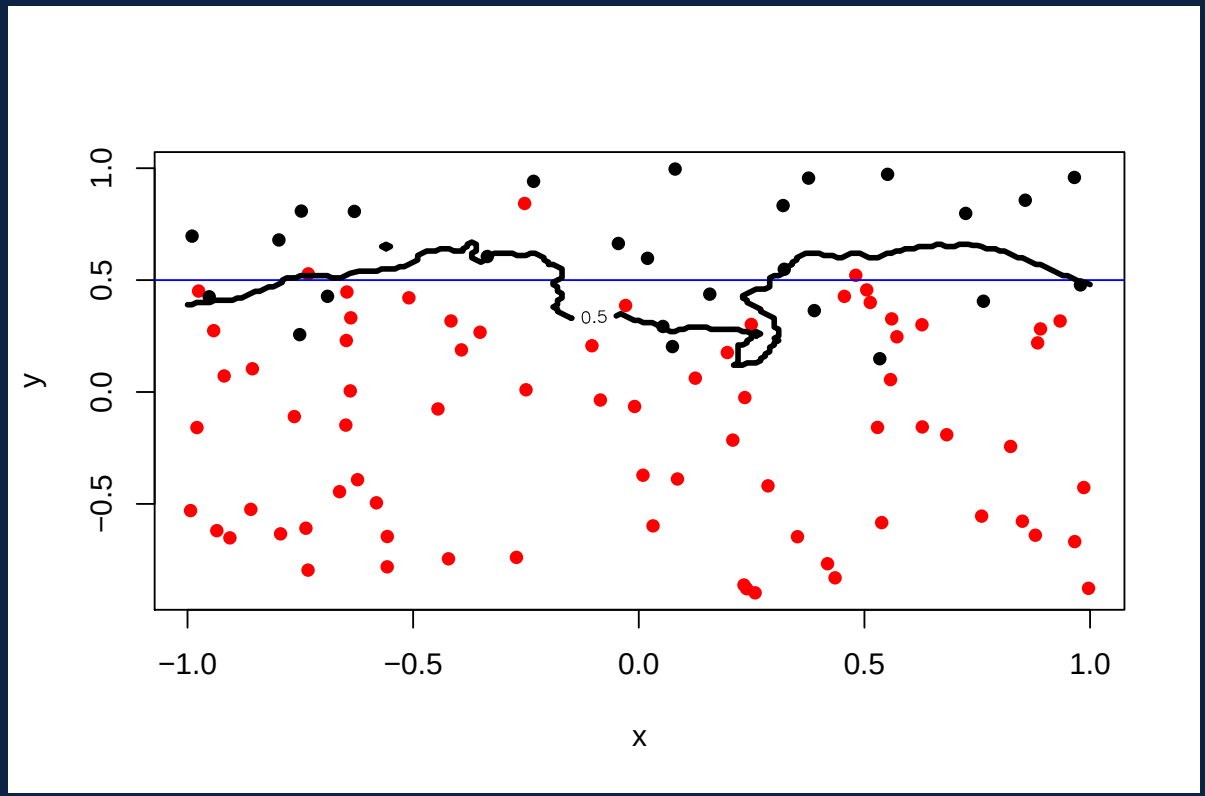
$$P(Y = j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

  3. Assign observation $x_0$ to $j$ according to the maximum probability.

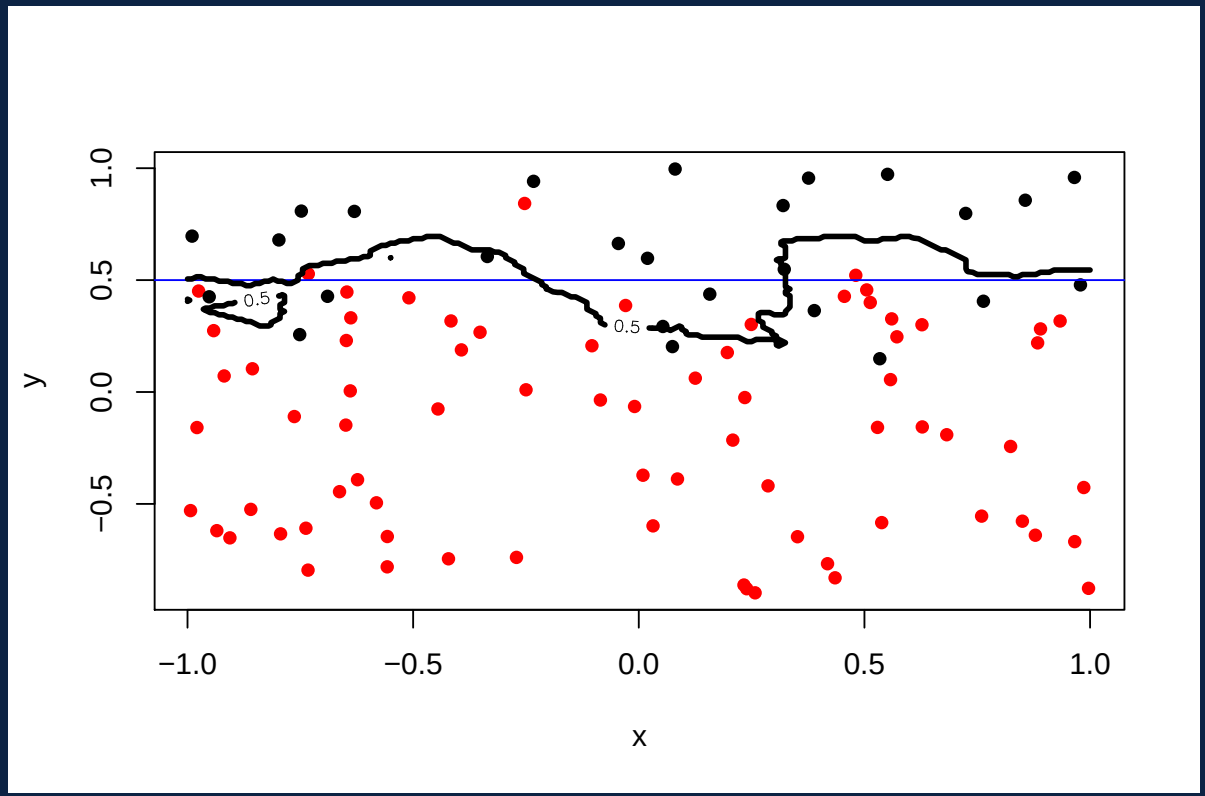
# KNN: Example ($k = 20$)



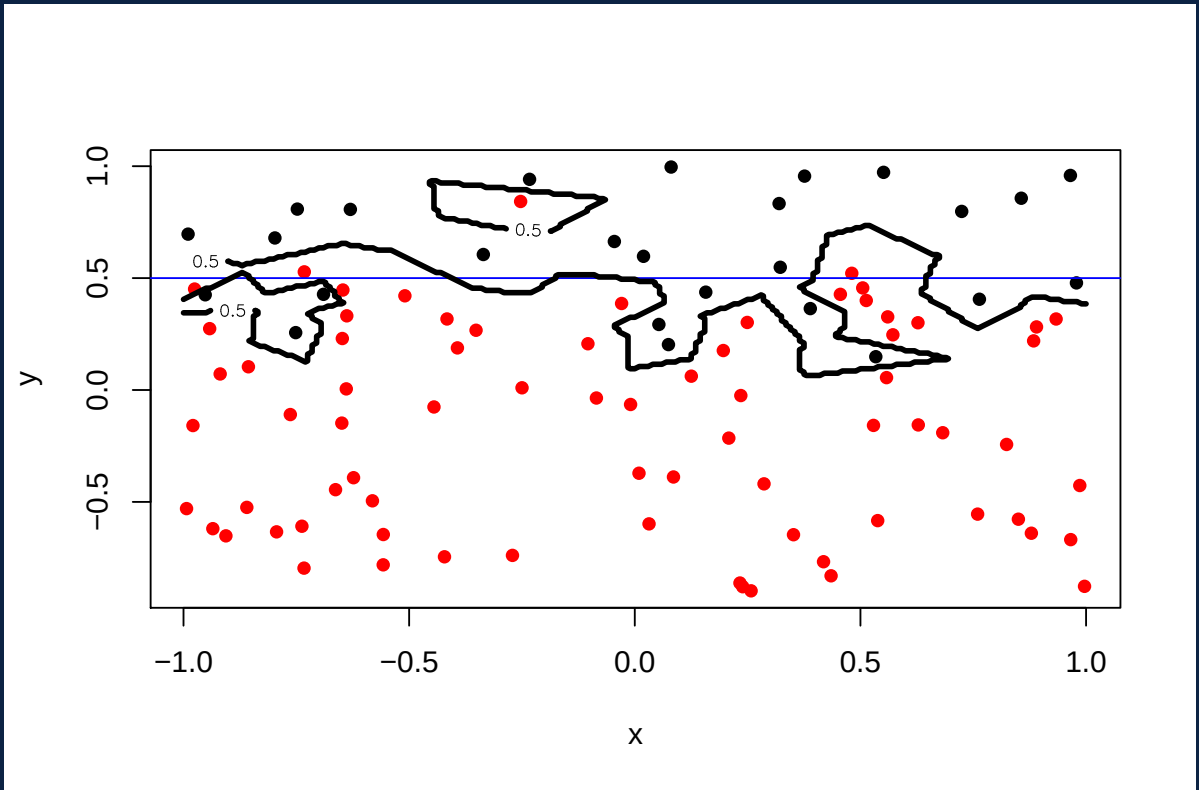
# KNN: Example ($k = 10$)



# KNN: Example ($k = 5$)



# KNN: Example ($k = 1$)



# KNN Discussion


- What if $k = n$?
- Some general rules of thumb for small/large $k$?


THE UNIVERSITY OF BRITISH COLUMBIA