Machine Learning in R

Lecture 1 – Introduction

Jean Feng – Epidemiology and Biostatistics

Course objectives

- Generate intuition and mechanistic understanding for a set of important machine learning methods

- Gain insight as to which methods are most appropriate in which situations

- Learn best practices in machine learning

- Discover potential pitfalls to avoid in machine learning

- Enable practical implementation of machine learning methods covered in R

Course topics

- What is machine learning
- Model choice, evaluation and comparison
- Supervised machine learning methods
- Unsupervised machine learning methods
- Applications in biomedical problems

Logistics

- Lectures and labs every Wednesday. Lectures will be recorded and posted online.

- Grades based on:

    - 5 homework assignments (70%)

    - Data analysis project: final project (30%)

- Turn in assignments and project thru CLE.

- All labs and homework assignments will be done in Rmarkdown. (If you would like to do homework in Python, please let me know.)

- Check materials online at CLE:

    - Detailed syllabus

    - Course forum

Data analysis project

- You will need to perform a data analysis project as part of the course (30% contribution to grade)

- You can either:

    - Analyze a dataset we provide to you
    - Or analyze your own dataset

- Turn in a formal writeup of your data analysis and findings

- You should start thinking about which dataset you want to use.

- Examples on CLE

R

- You need to have R installed on your laptop: https://cran.r-project.org/. (Current version is R 4.1.1.)

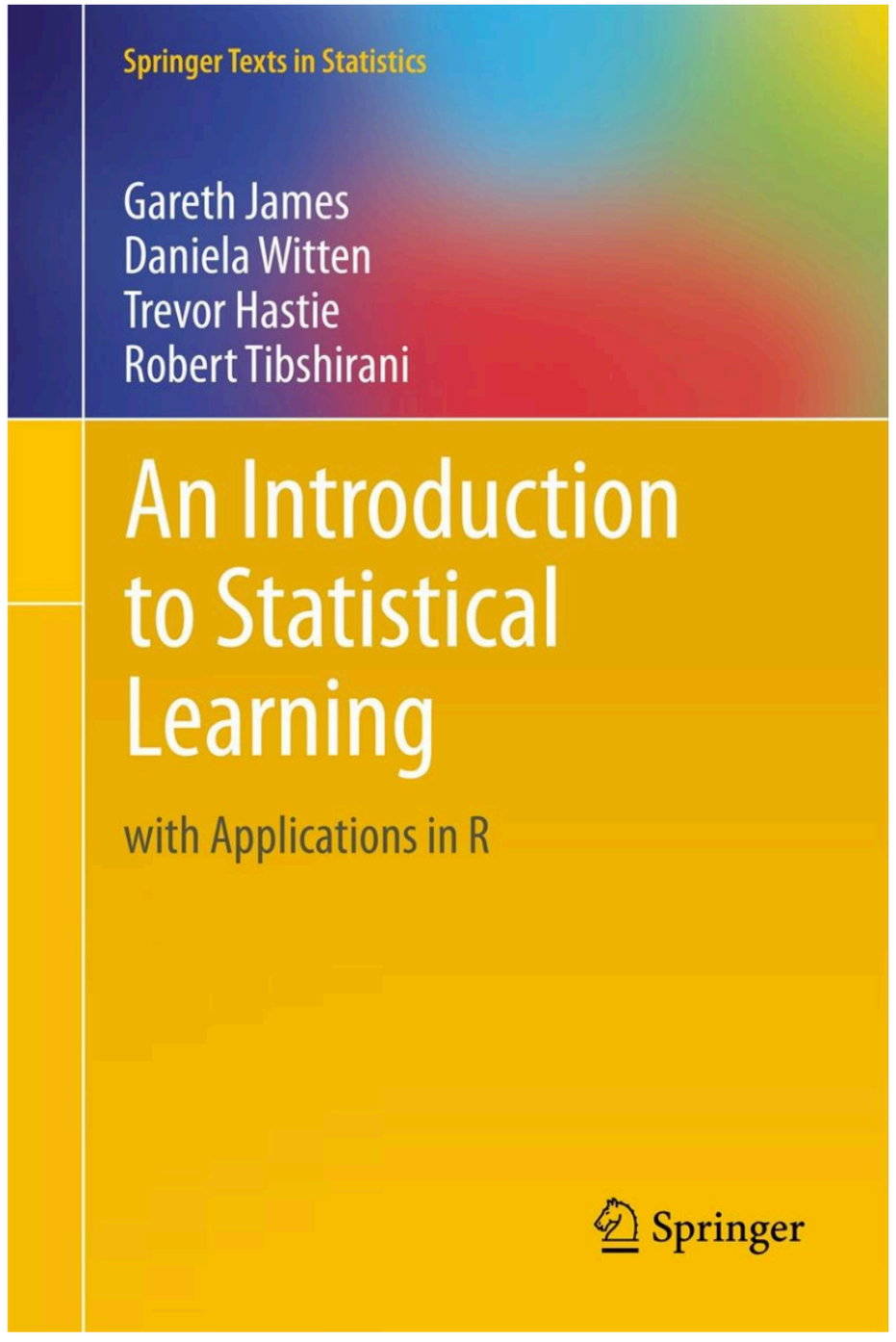
- Install RStudio Desktop: https://www.rstudio.com/products/rstudio/

- Installing packages in R:
  - `install.packages(“package_name”)`
  - `install.packages(“package_name1”, “package_name2”)`
  - Try an example to make sure you don’t run into problems: e.g. `install.packages("lme4")`

Texts

- Parts of this course borrow from *An Introduction to Statistical Learning*, and associated lectures. James et al., available for free at: https://statlearning.com/

- Also useful is the more technical *Elements of Statistical Learning*: http://web.stanford.edu/~hastie/ElemStatLearn/ (again, available free)



TICR Professional Conduct Statement

- I will maintain the highest standards of academic honesty
- I will not use answer keys from prior years
- Problem Sets and Projects: I am permitted to consult with other classmates, but I will write final answers in my own words away from other classmates

Today's lecture

1. **What is Machine Learning?**
    1. **Supervised vs Unsupervised Learning**
2. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. Model selection
    5. Bias-Variance Trade-off
    6. Multivariate models

What is machine learning?

- Machine learning (ML) is the study of computer algorithms that learn to make predictions about outcomes of interest or identify relationships between variables given a training dataset.

- A lot of ML is built on statistics. We'll see these connections through the course.

- How does ML differ from traditional statistical modeling? We do not necessarily use a *probabilistic* model to characterize relationships. We are allowed to use any procedure that does the job well.

Examples

- Predict disease-free survival times of breast cancer patients based on diagnostic measures, demographic variables and genetic information

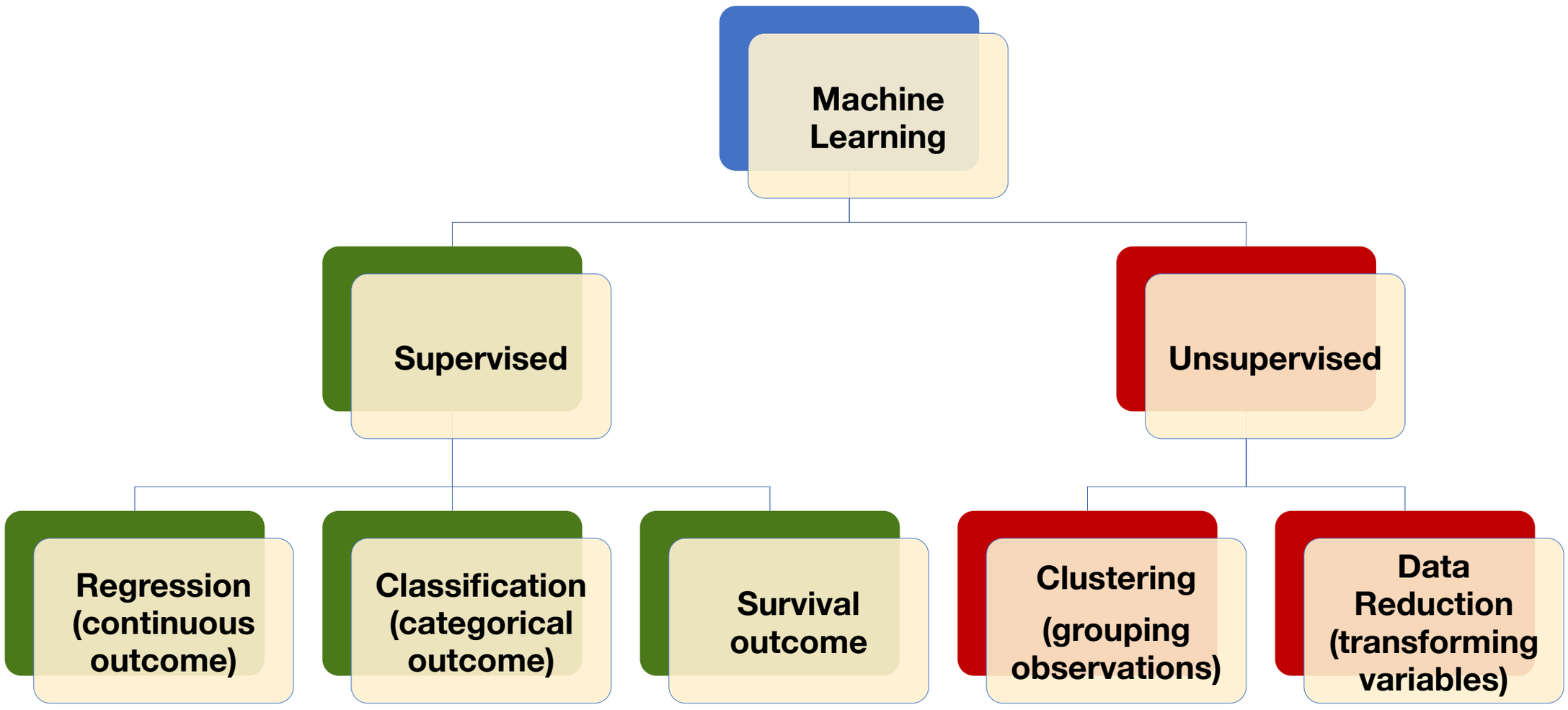
- Determine whether there are distinct subgroups of patients with anxiety and depression based on different patterns of progression

- Automatically label patients as different types of dementia based on neuropsychological testing scores

Supervised vs Unsupervised

Supervised Learning	Unsupervised Learning
• Dataset contains predictors X and outcomes Y	• Dataset contains predictors X but does not contain outcomes Y
• Goal: Train a model to predict future outcomes from sets of predictors	• Goal: Discover structures and patterns in the data
• Techniques: Regression (continuous outcomes) and classification (categorical outcomes)	• Techniques: Clustering and dimension reduction

Supervised vs Unsupervised



Are the following questions **supervised** or **unsupervised**?

- Predict disease-free survival times of breast cancer patients based on diagnostic measures, demographic variables and genetic information

- Determine whether there are distinct subgroups of patients with anxiety and depression based on different patterns of progression

- Automatically label patients as different types of dementia based on neuropsychological testing scores

Are the following questions **supervised** or **unsupervised**?

- Predict disease-free survival times of patients breast cancer surgery based on diagnostic measures, demographic variables and genetic information (supervised)

- Determine whether there are distinct subgroups of patients with anxiety and depression that may have different patterns of progression (unsupervised)

- Automatically label patients as different types of dementia based on neuropsychological testing scores (supervised and unsupervised)

Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. **Supervised learning:**
    1. **Notation**
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. Model selection
    5. Bias-Variance Trade-off
    6. Multivariate models

Supervised Learning

- **Response:** what we want to predict, also known as the outcome, target, or dependent variable

$$Y$$

  - In a regression problem, Y is continuous

  - In a classification problem, Y is a categorical variable

$$x_i$$

  - In a survival problem, Y is a survival outcome (survival time and status censor/uncensored)

- **Predictors**: inputs, regressors, covariates, features, or variables

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

  - e.g. X_1(heart rate), X_2 (exercise yes/no), etc.

Supervised Learning

- **Dataset**:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

$$\{(x_i, y_i) : i = 1,...,n\}$$

- **Model**: A function that outputs Y given a point x
  - True model/underlying population model: $f(x)$
  - Fitted model: $\hat{f}(X)$ (the hat notation denotes that it is an estimate of the true model)

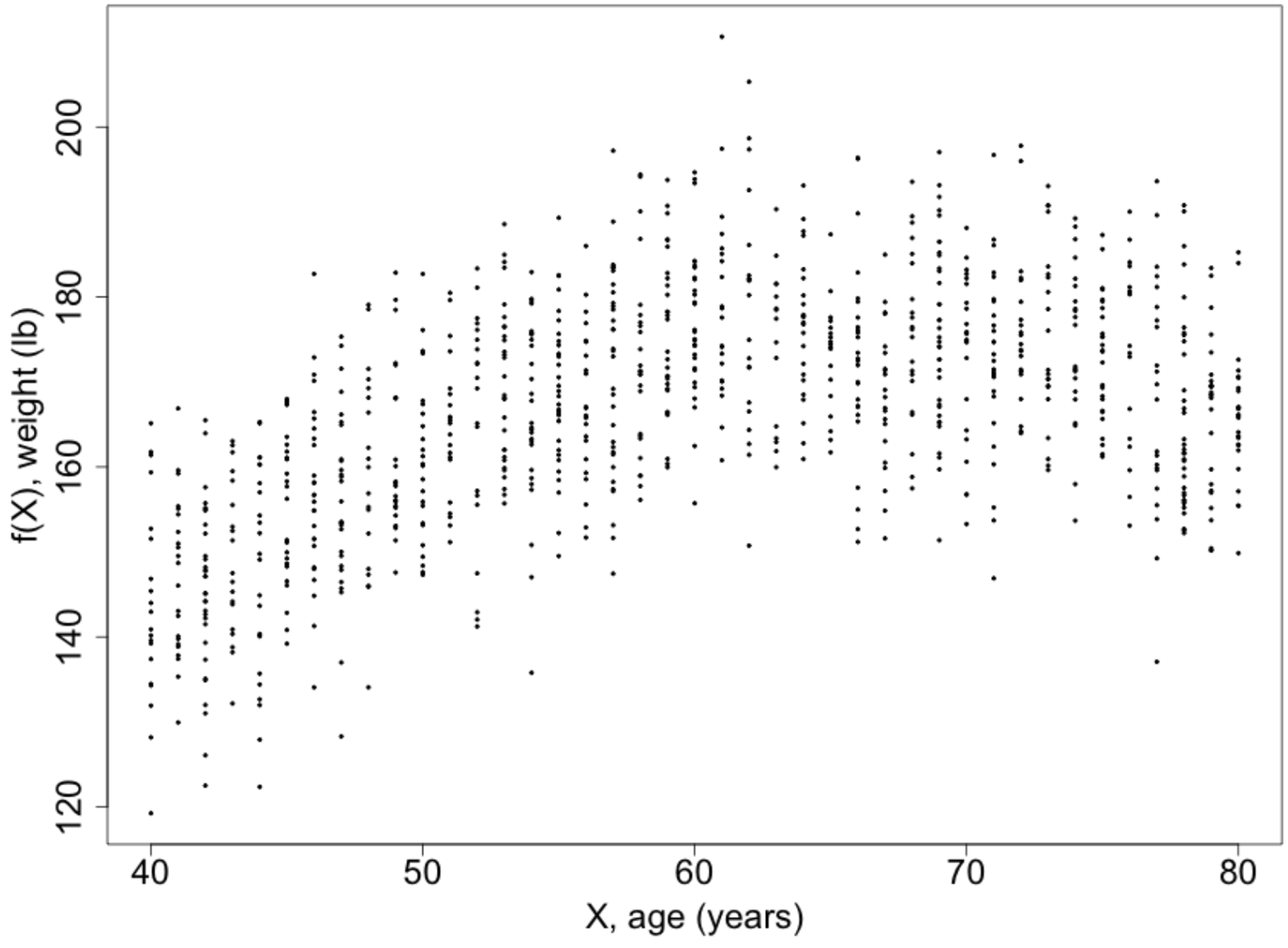
Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. Supervised learning:
    1. Notation
    2. **Evaluating models using a loss function**
    3. Parametric and non-parametric models
    4. Model selection
    5. Bias-Variance Trade-off
    6. Multivariate models

What makes a good prediction model?

- Consider supervised learning with a continuous outcome and a single predictor

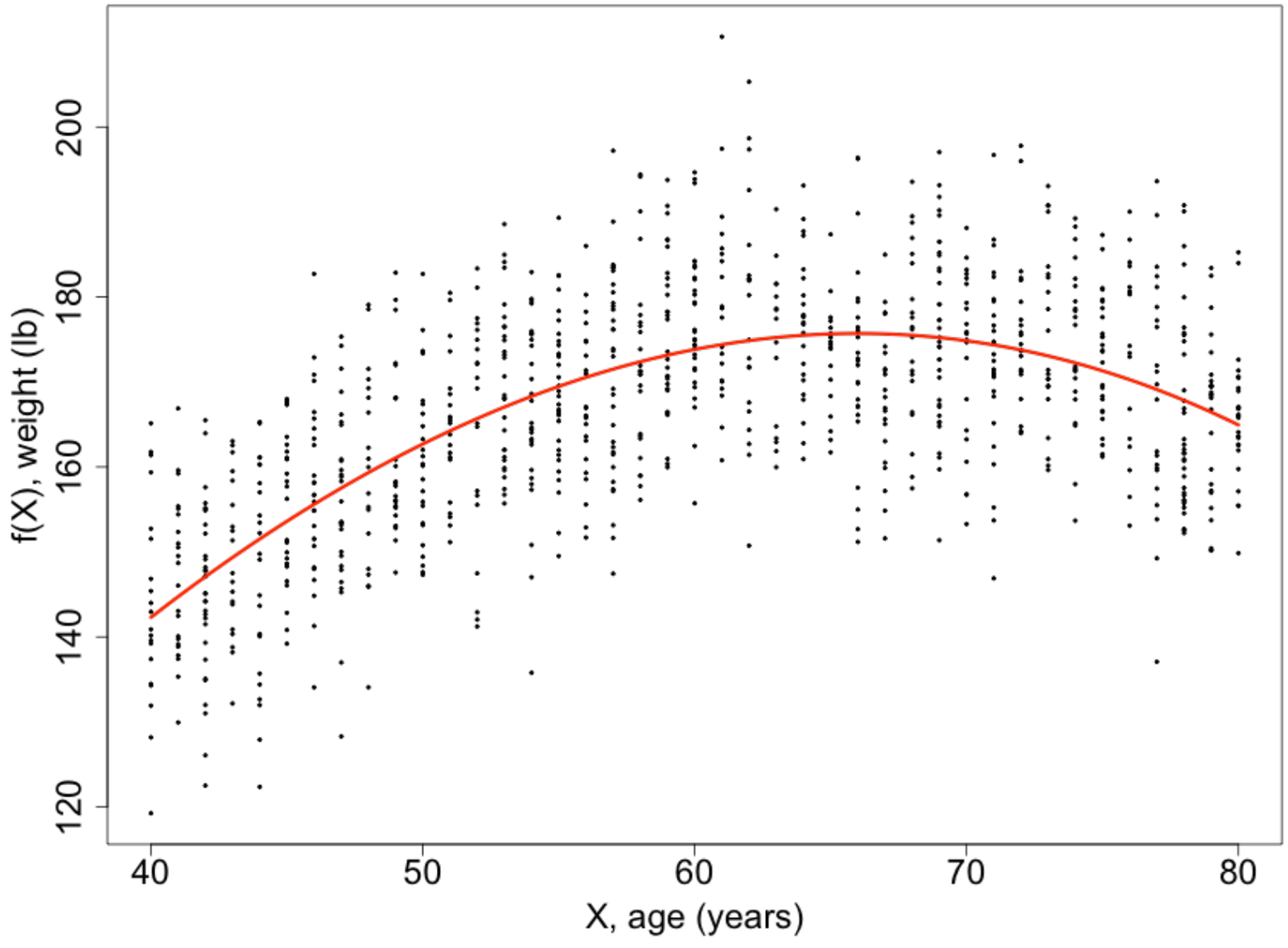
Training data



What makes a good prediction model?

- Consider supervised learning with a continuous outcome and a single predictor

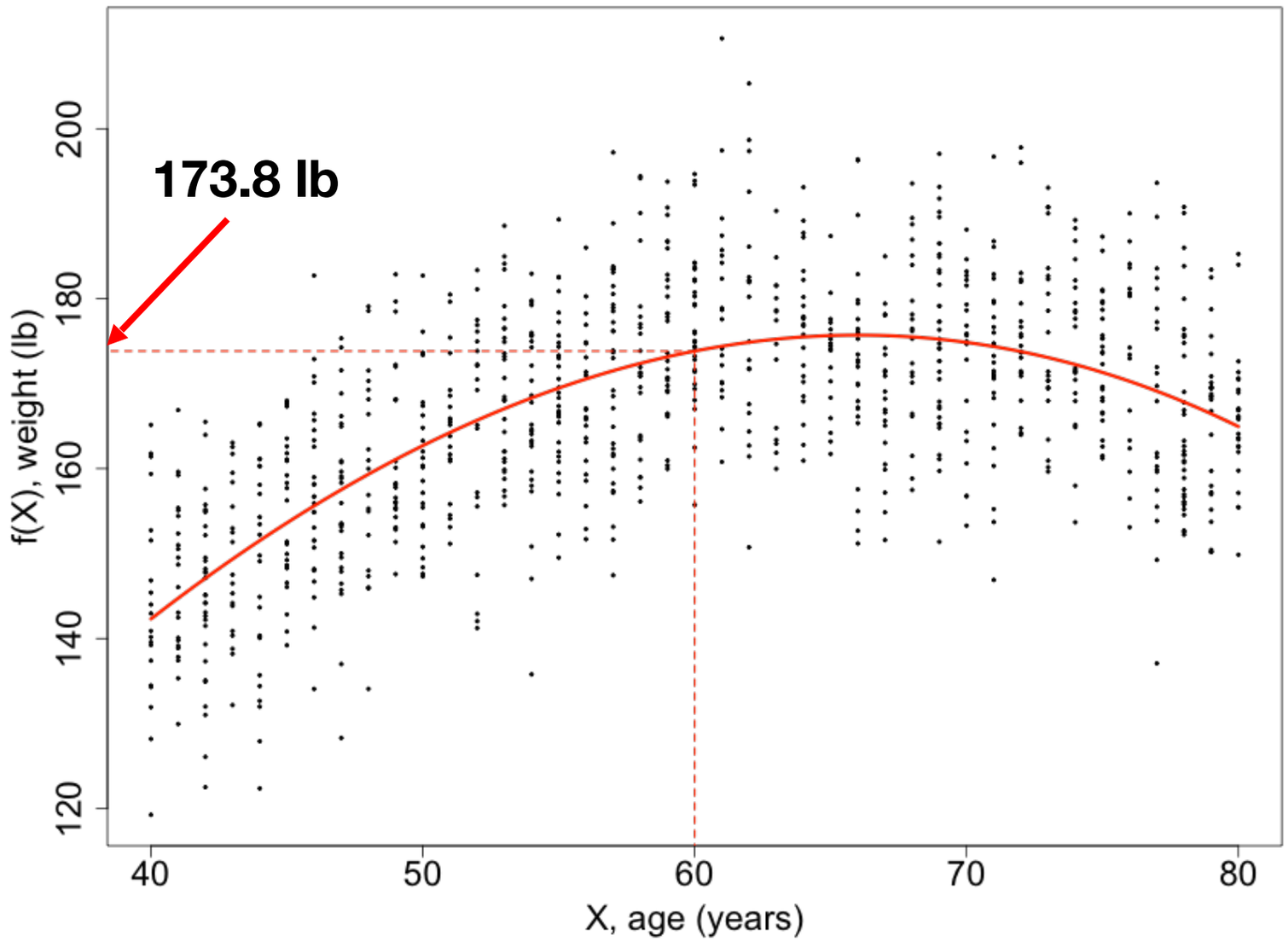
Training data



What makes a good prediction model?

- Consider supervised learning with a continuous outcome and a single predictor

Training data



Loss functions

- Prediction models are usually evaluated using a **loss function** $\ell(y, \hat{y})$.

- For continuous outcomes, we most often use the squared error loss $\ell(y, \hat{y}) = (y - \hat{y})^2$.

- Our goal is to find a model $\hat{f}$ that minimizes the expected squared error loss, i.e.

$$\mathbb{E}\left[(Y - \hat{f}(X))^2\right].$$

- **Q: What model minimizes this expectation?**

The squared error loss

- **Q: What model minimizes the expected squared error loss, i.e.** $\mathbb{E}\left[(Y - \hat{f}(X))^2\right]$**?**

- **A:** The best model predicts the average response $\mathbb{E}[Y \mid X = x]$ for given variables $x = (x_1, \ldots, x_p)$.

To see why, note that:

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

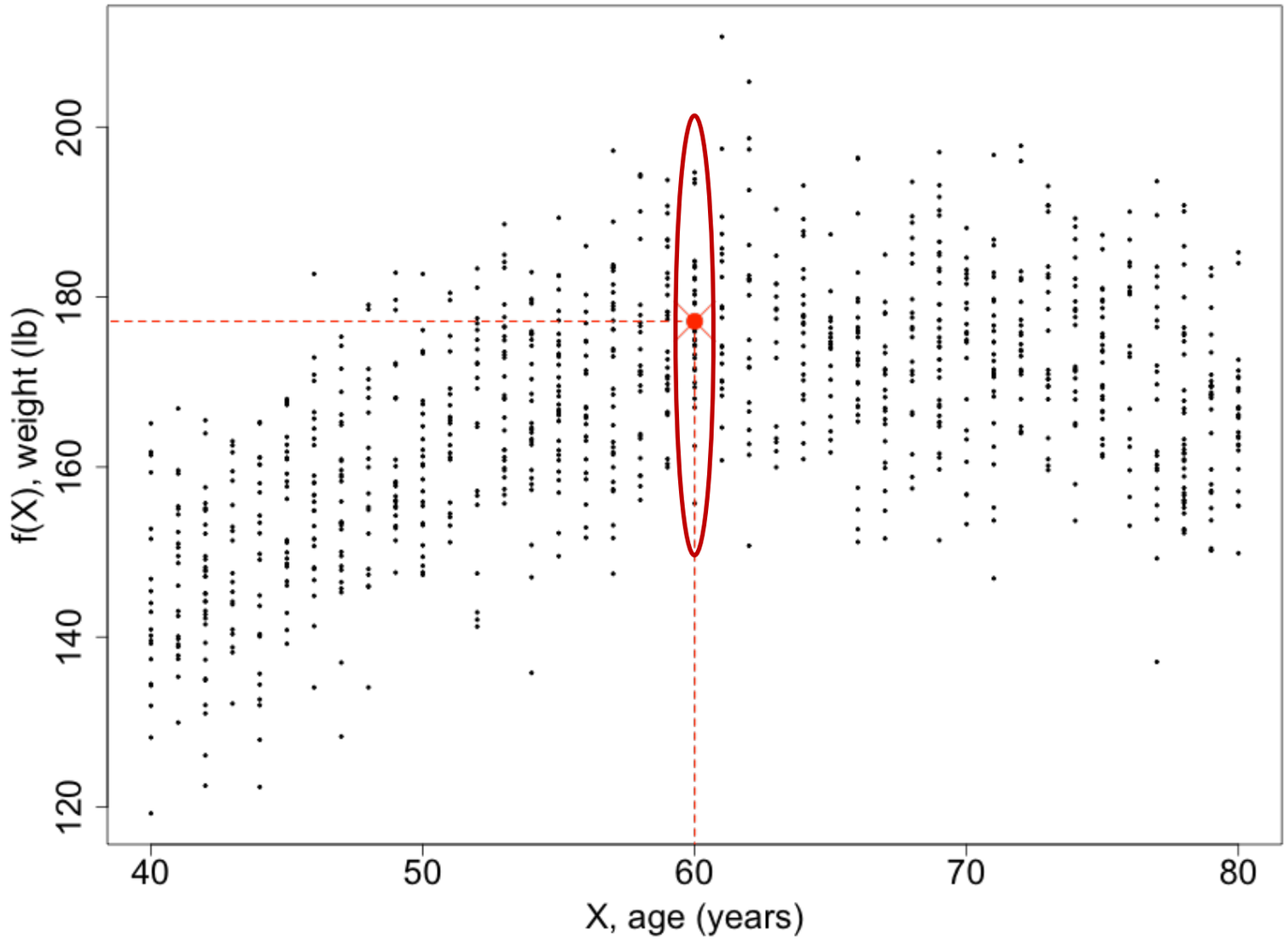
Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. **Parametric and non-parametric models**
    4. Model selection
    5. Bias-Variance Trade-off
    6. Multivariate models

So… can we just predict the mean?

- The previous slide suggests a model that predicts the weight for someone with age x using the mean weight across all individuals of age x in the dataset.

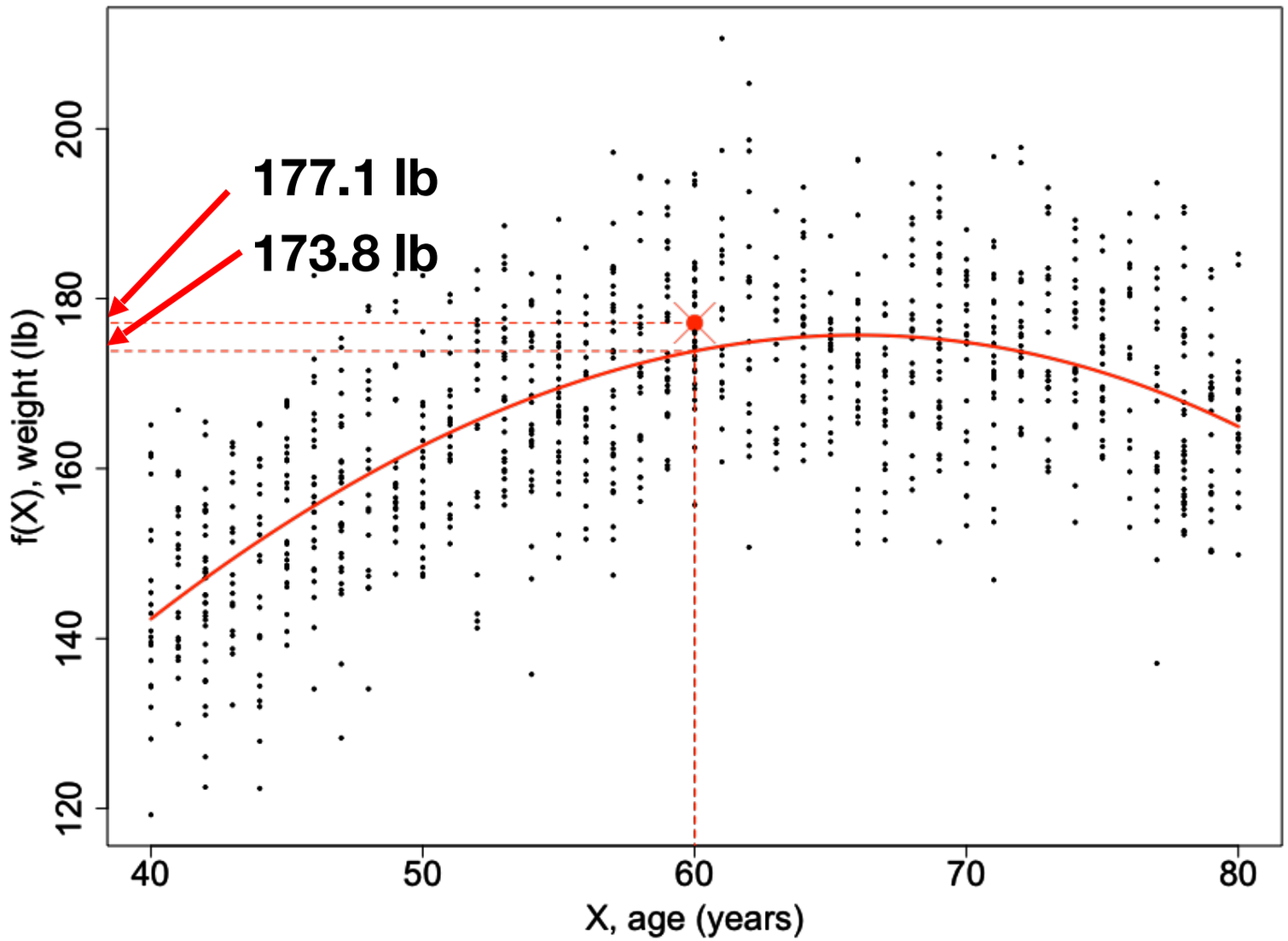
Training data



So… can we just predict the mean?

- It's not a bad idea. The estimate will be unbiased, though it can be very noisy.

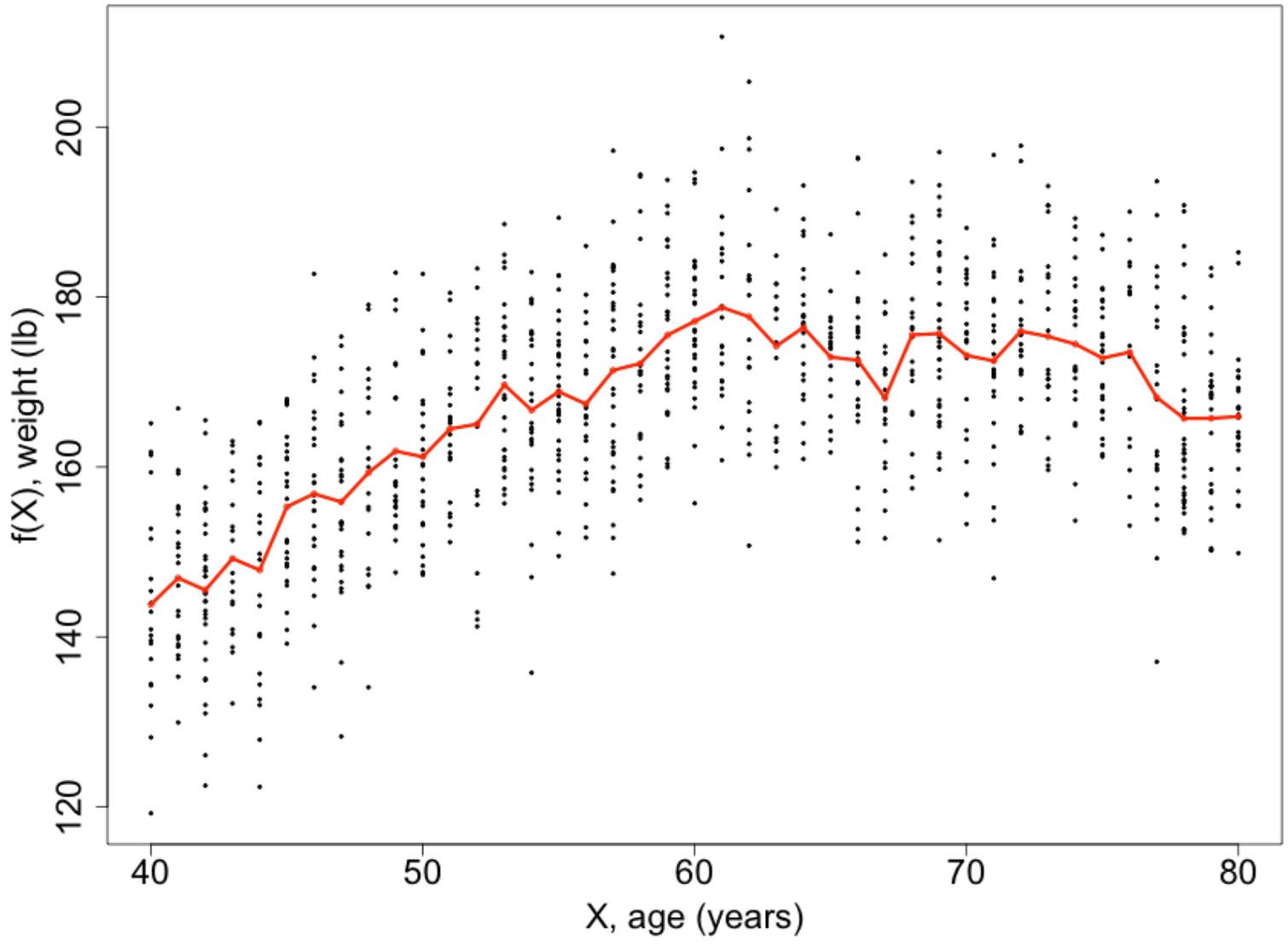
Training data



So… can we just predict the mean?

- Plotting the prediction for all ages, we see that the prediction can indeed be quite noisy.

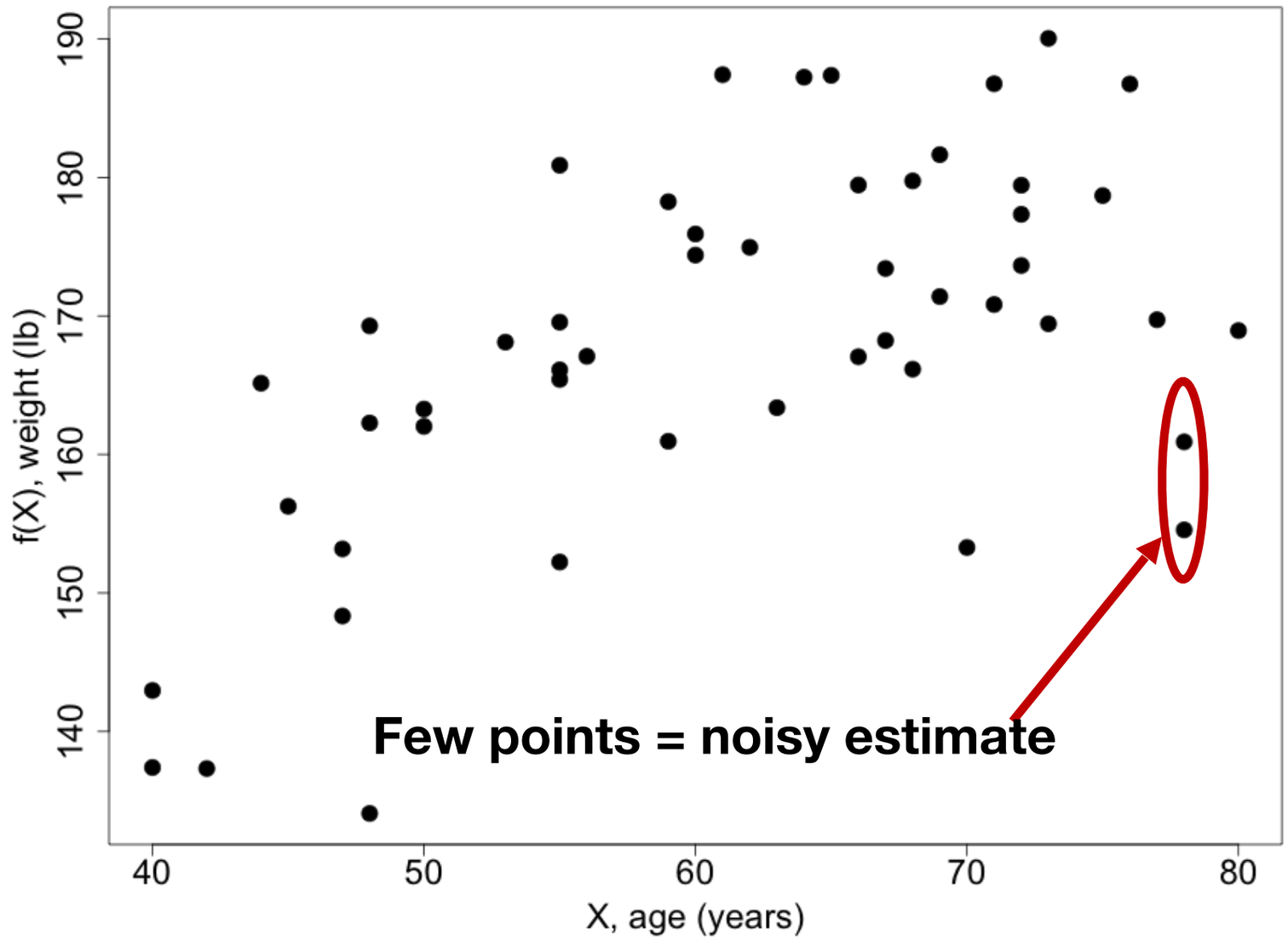
Training data



What if we have less data?

- The bigger problem in most cases is that we don't have enough data and the mean is too noisy.

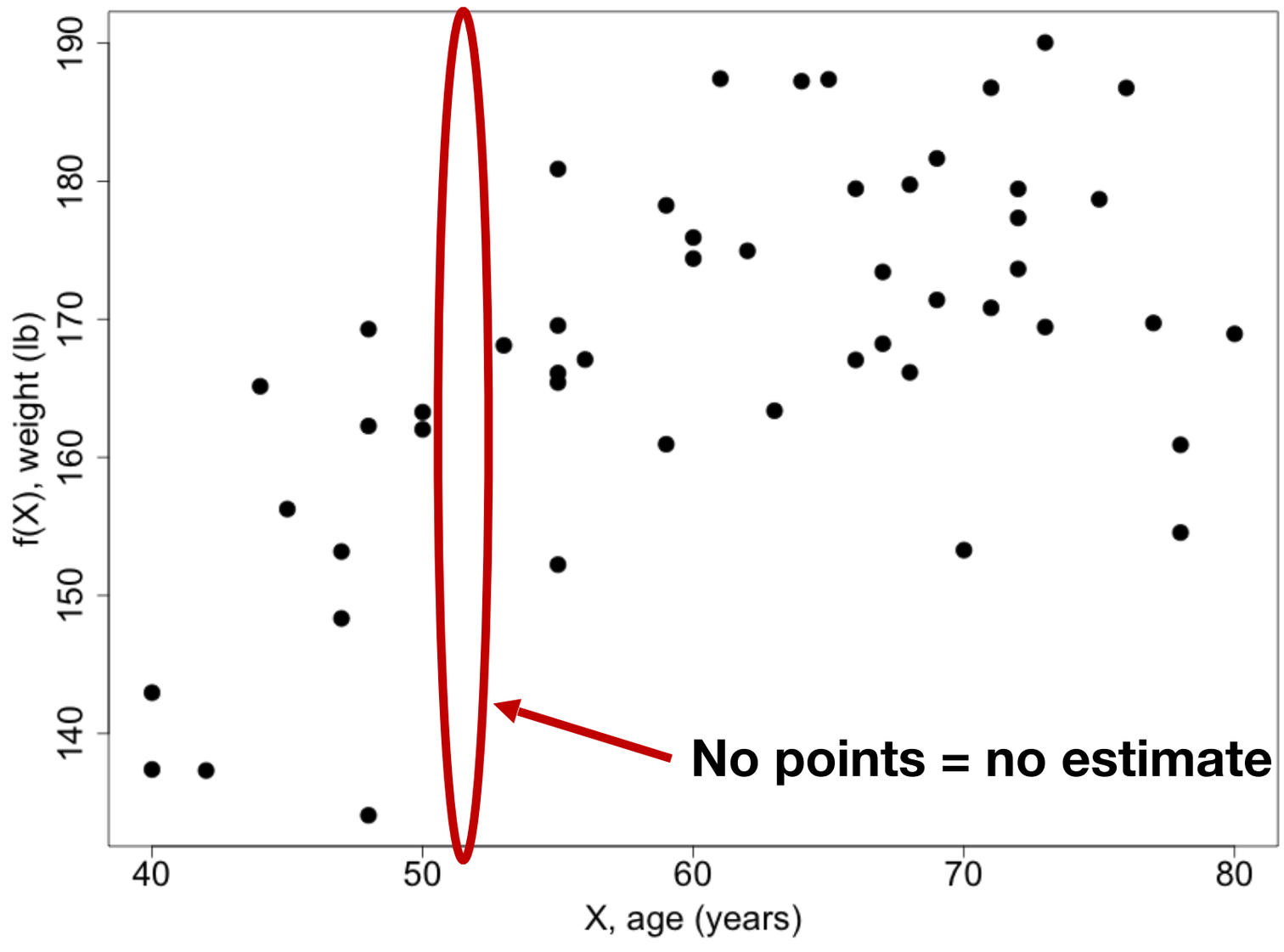
Training data



What if we have less data?

- In some cases, we have no individuals for a particular age, which means we can't even compute the mean.

Training data

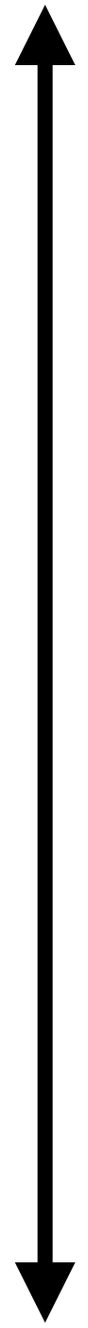


What if we have less data?

- **A:** Make some assumptions. For example:
  - The age vs weight relationship is linear.
  - The age vs weight relationship is quadratic.
  - The age vs weight relationship is smooth.
  - …

Parametric vs Non-parametric

- **Parametric methods** make an assumption about the functional form of the true model and aims to estimate the parameters in the true model.
    - For example, a common assumption is that the model is linear: $E[Y \mid X = x] = \beta_0 + \beta_1 x$. Linear regression aims to estimate β_0 and β_1.
- **Non-parametric methods** try to estimate the true model without making explicit assumptions on its functional form.
    - An example of this is the procedure we just described, where we predict the mean. The problem with this approach is that without any assumptions, the predictions can be very noisy.

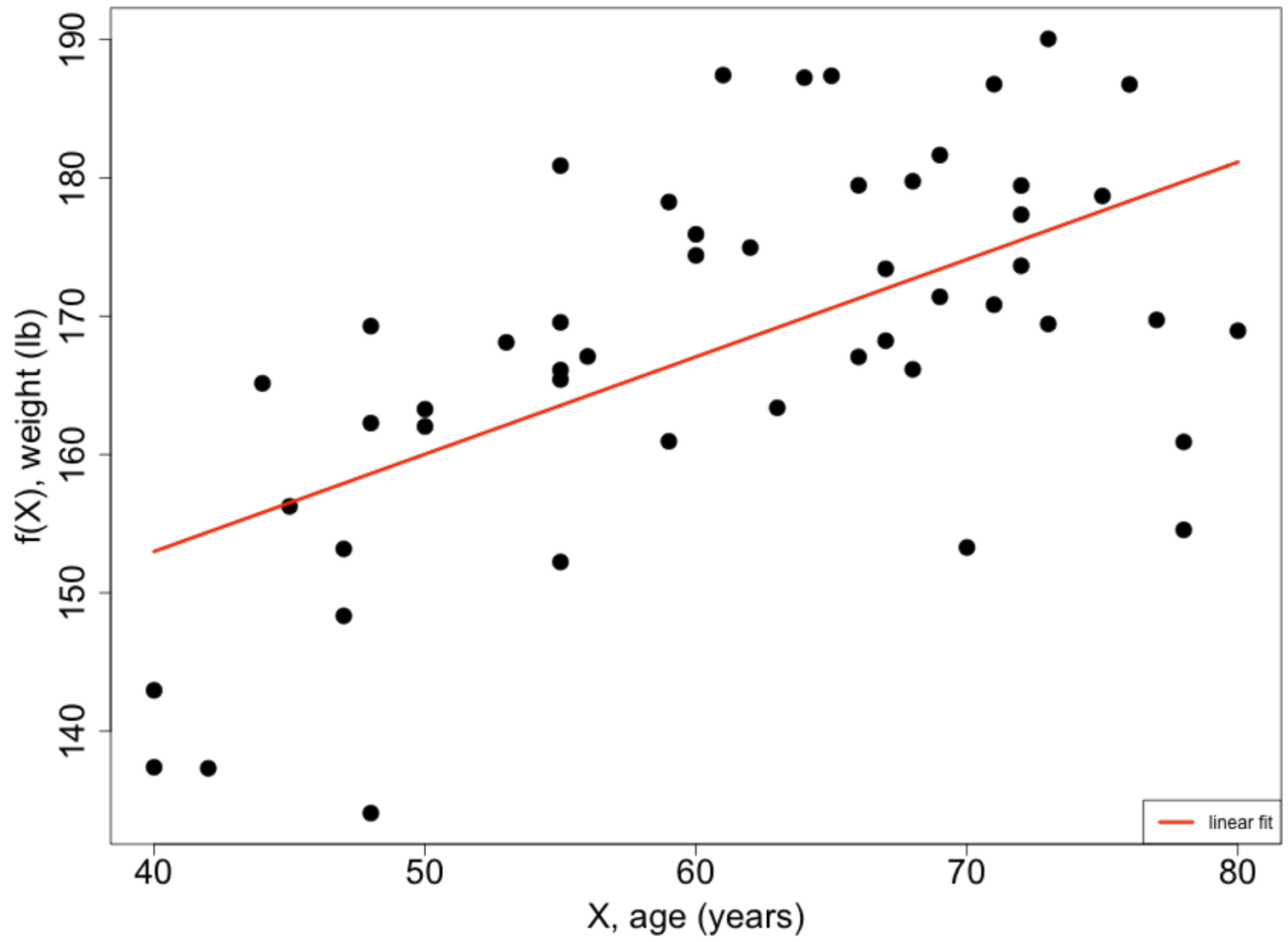


Very strong assumptions

Very weak assumptions

Solution 1: Linear regression (Parametric)

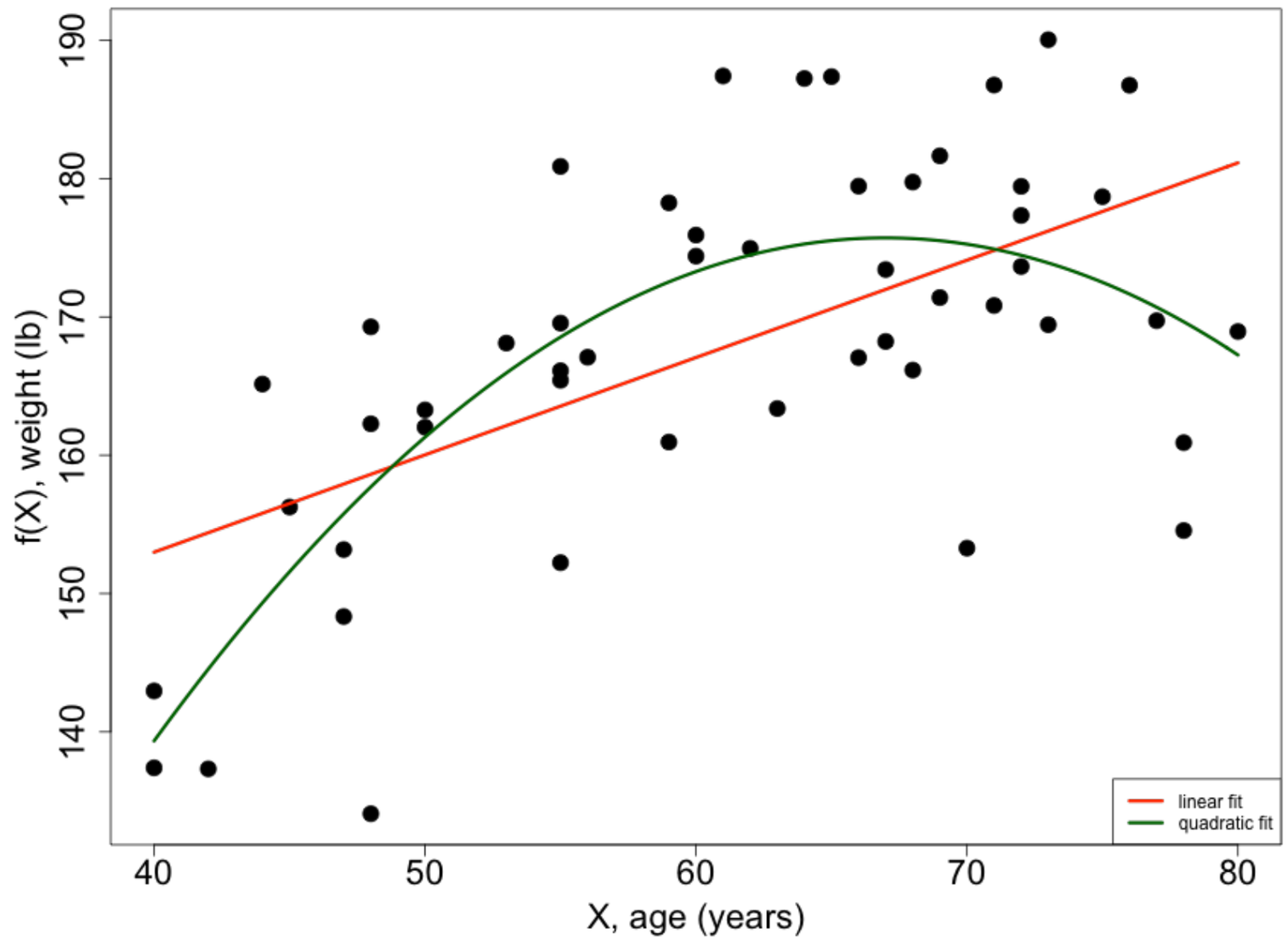
- Using linear regression, we estimate the model
$\hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X_1$



Solution 2: Polynomial regression

(Parametric)

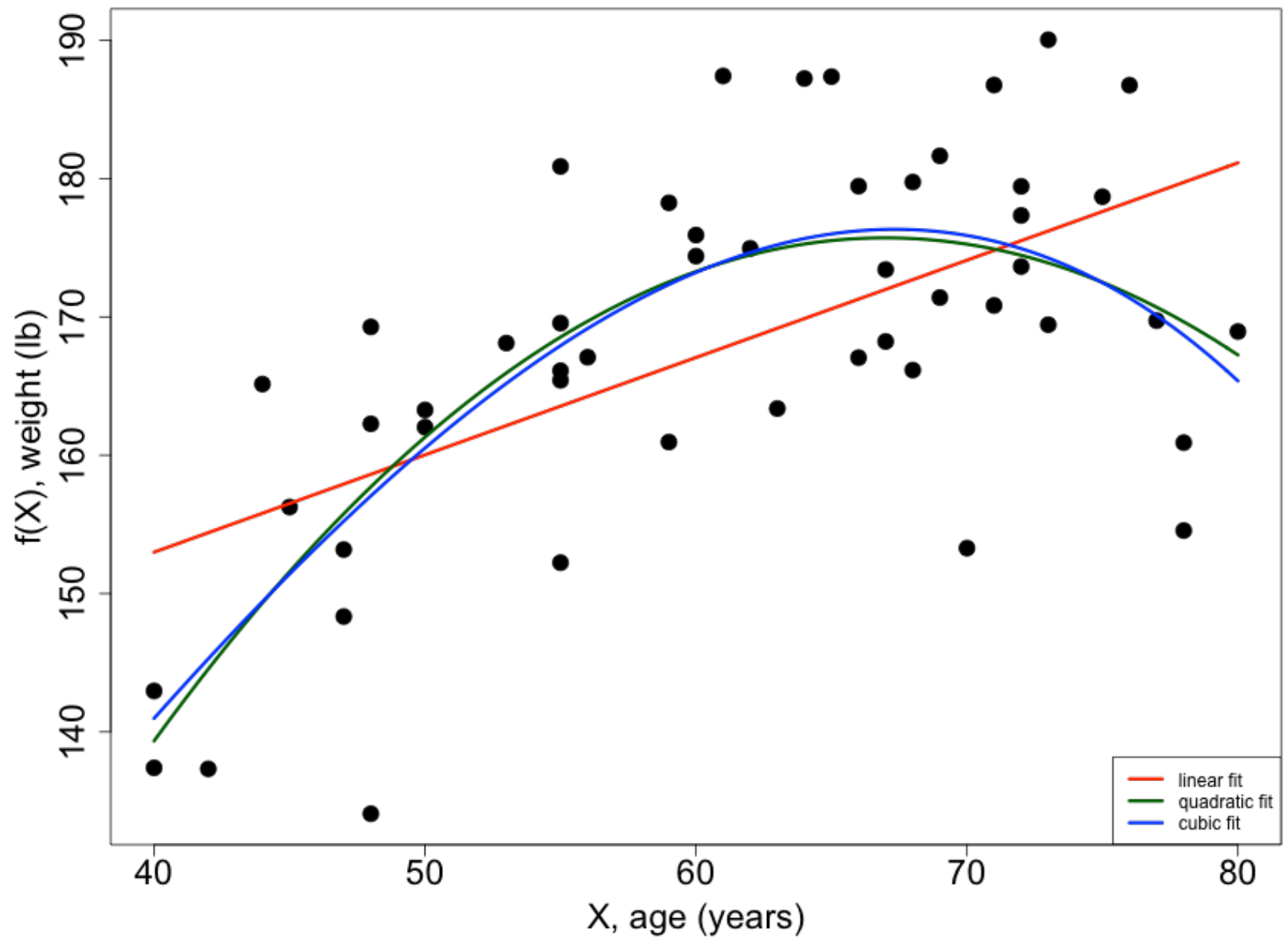
- We can make this model more flexible by doing a quadratic fit: $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_1^2$



Solution 2: Polynomial regression

(Parametric)

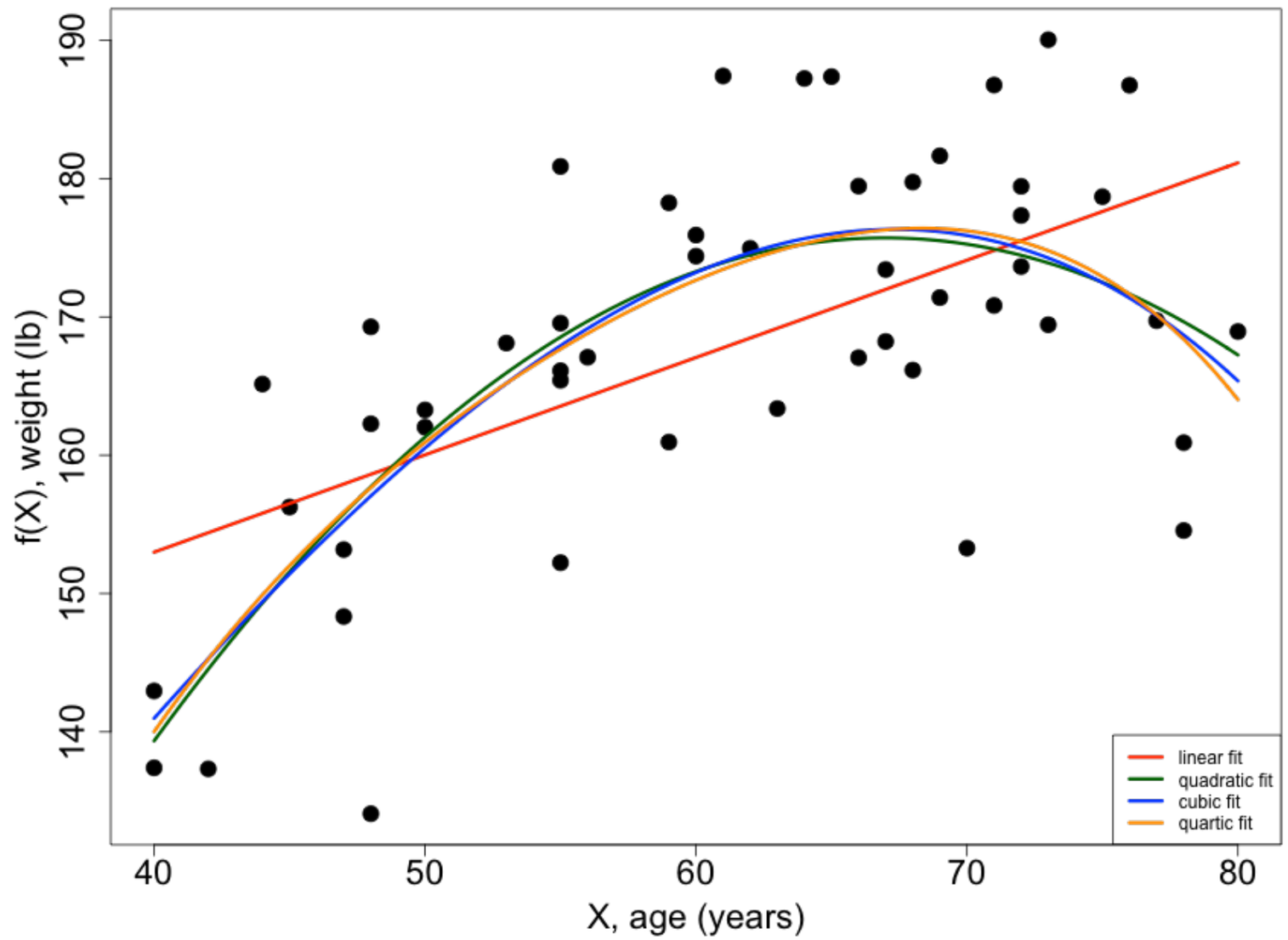
- A cubic fit is even more flexible:
  $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_1^2 + \hat{\beta}_3 x_1^3$



Solution 2: Polynomial regression

(Parametric)

- A quartic fit? $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_1^2 + \hat{\beta}_3 x_1^3 + \hat{\beta}_4 x_1^4$



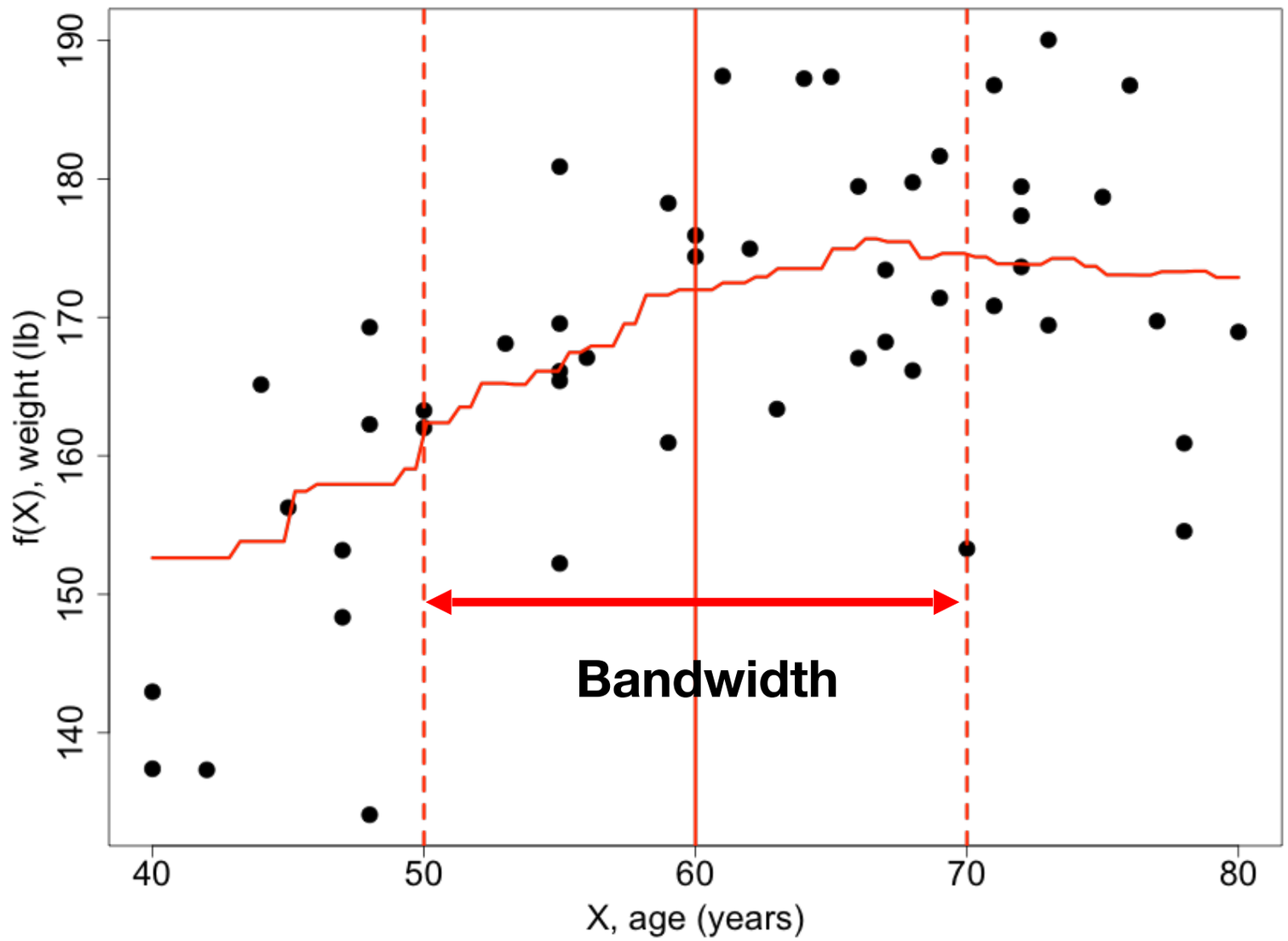
Solution 3: Kernel smoothing

(Non-parametric)

- A common non-parametric solution is to "average" over a *neighborhood* of X using kernel smoothing:

```
ksmooth(x, y, kernel = "box", bandwidth = 20)
```

Training data



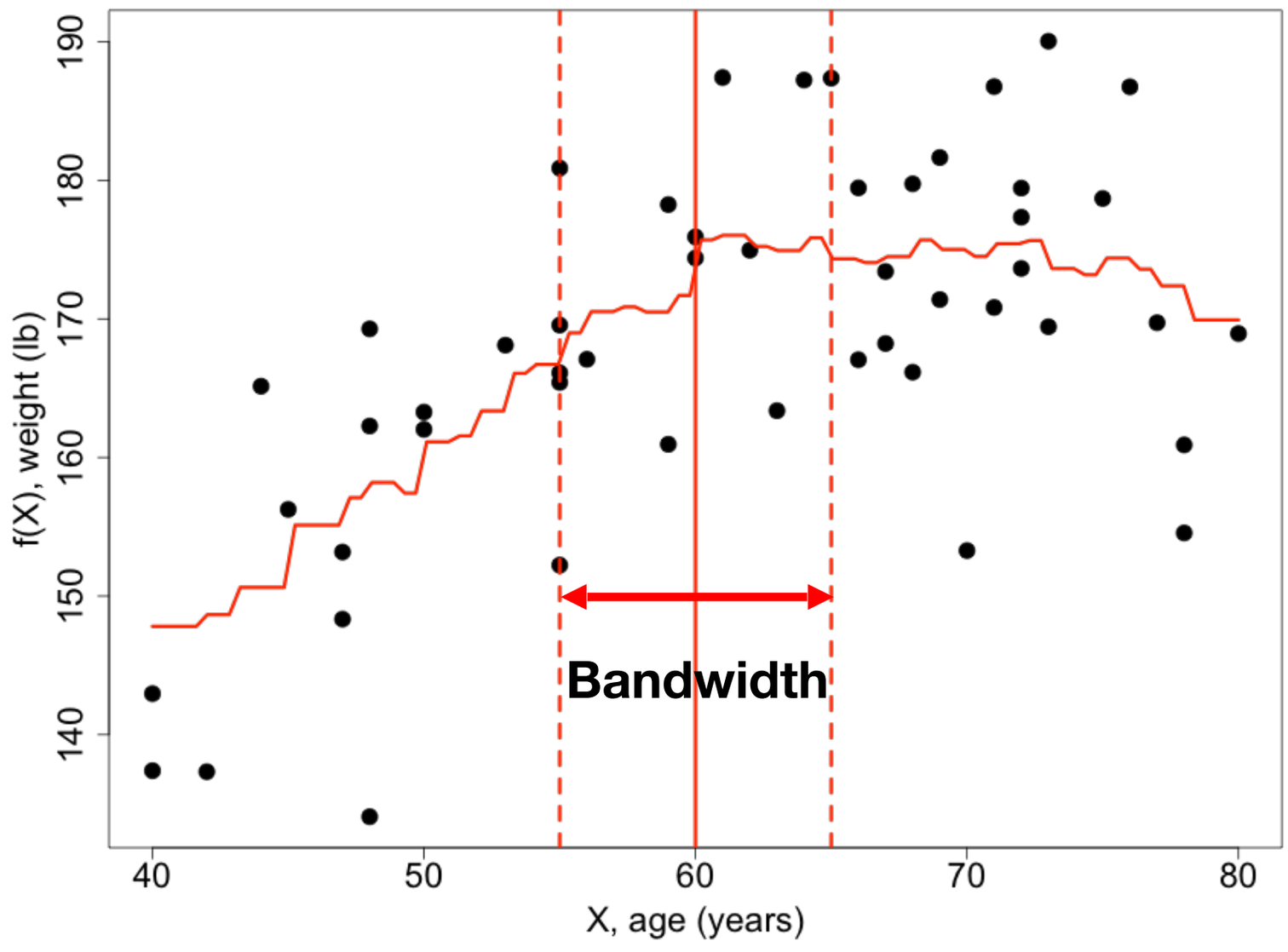
Solution 3: Kernel smoothing

(Non-parametric)

- We can fit more wiggly models by decreasing the kernel bandwidth:

```
ksmooth(x, y, kernel = "box", bandwidth = 10)
```

Training data



Parametric vs Non-parametric

- Which one should I use?
    - General rule of thumb: Non-parametric models typically require more data to be accurate. So use a parametric model for a small dataset and a non-parametric method for a large dataset.

Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. **Model selection**
    5. Bias-Variance Trade-off
    6. Multivariate models

Model selection

- We just showed many different ways for fitting a prediction model:
    - Linear regression
    - Polynomial regression with different degrees
    - Kernel smoothing with different bandwidths
- **Q: How do we pick among all these models and all these hyperparameters?**

Model selection

- We just showed many different ways for fitting a prediction model:
    - Linear regression
    - Polynomial regression with different degrees
    - Kernel smoothing with different bandwidths
- **Q: How do we pick among all these models and all these hyperparameters?**
- **A:** Estimate the expected loss of the fitted models and pick the one with the lowest value.

Estimating the expected loss

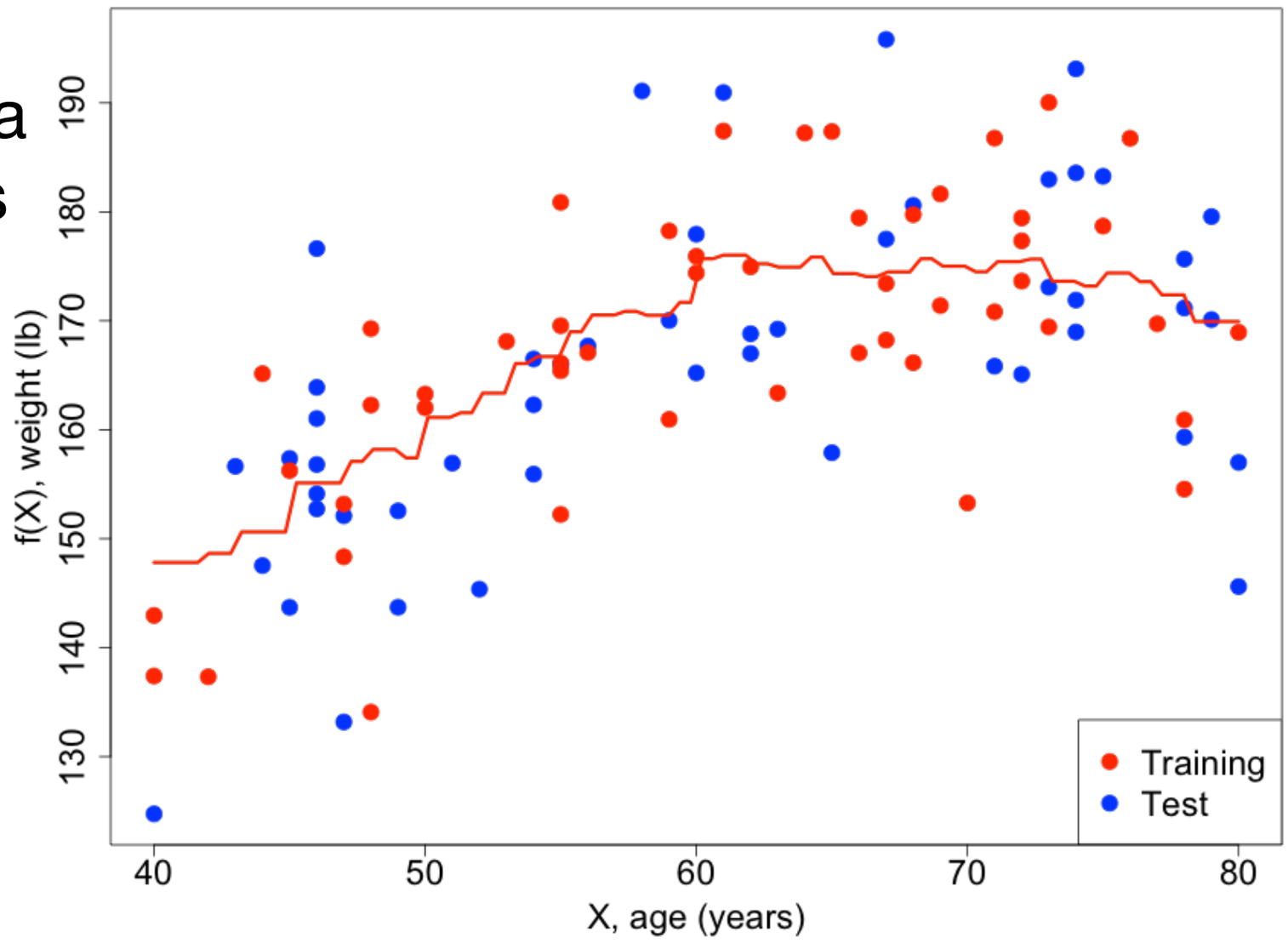
- **Key point**: You cannot estimate the expected loss of a machine learning model using the data you just used to fit the model.

- Machine learning models are very flexible, unlike traditional statistical models. They can easily **overfit** to the training data.

- **Overfitting** is when the prediction method achieves a very small loss on the data it trained on, but its expected loss in the target population is much larger.

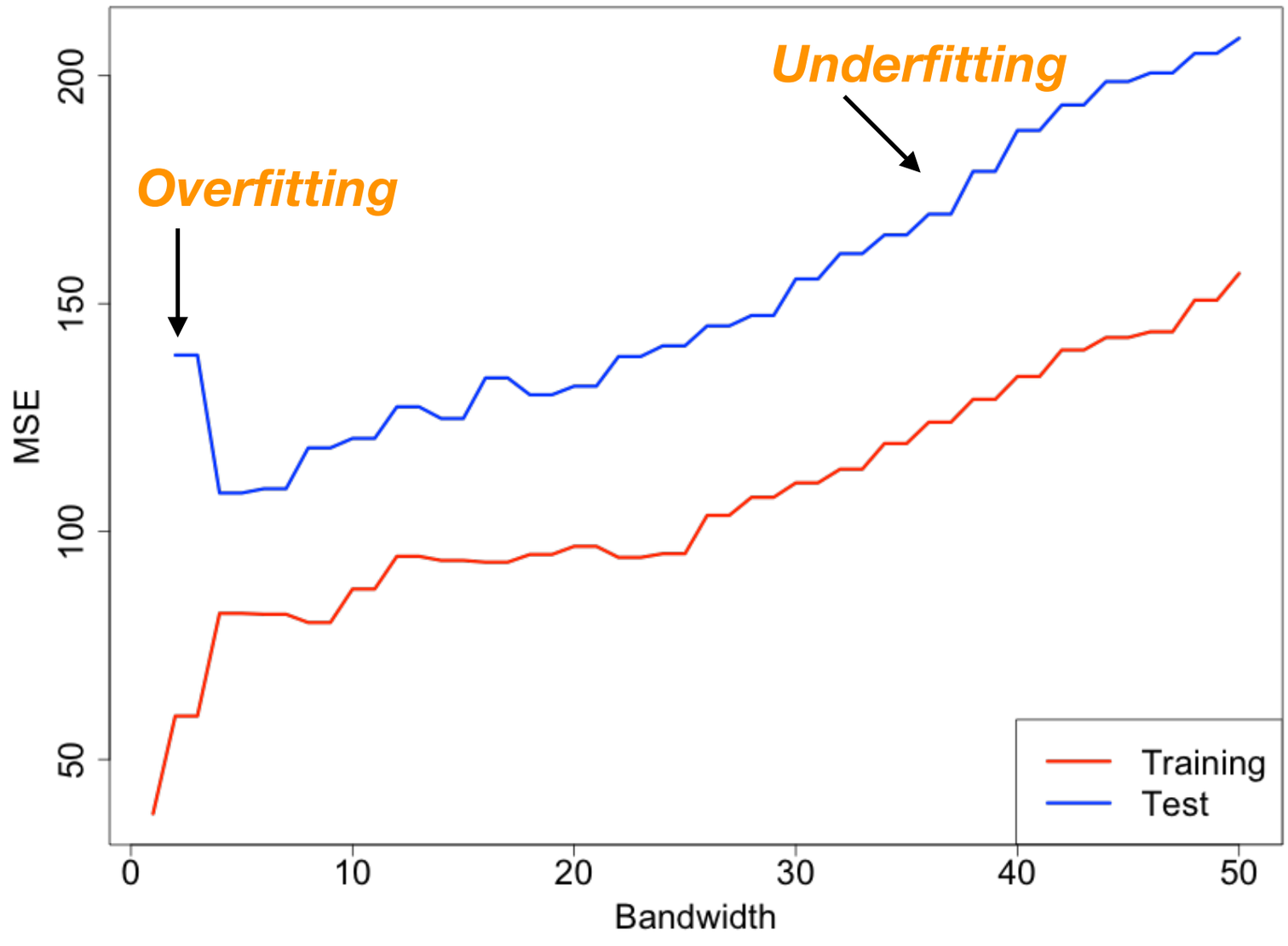
A test dataset

- We should instead estimate the expected loss using a test dataset, which is disjoint from the training data.



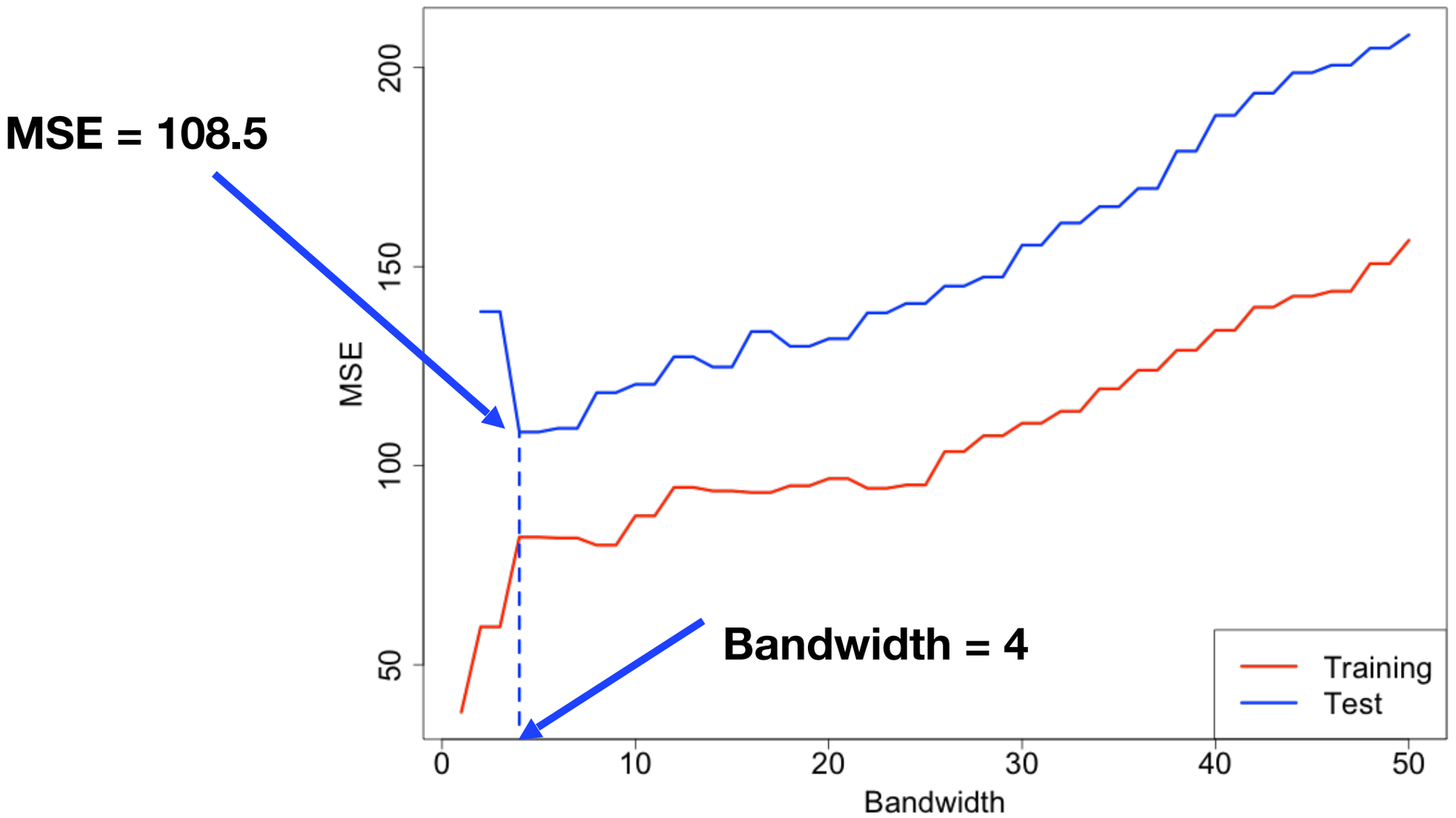
Example: Choosing the bandwidth

- Let's pick the best bandwidth for a kernel smoothing approach.



Example: Choosing the bandwidth

- The bandwidth that achieved the minimum mean squared error on the test data is 4.



What is a valid test dataset?

- **Q:** Suppose the data used to train our ML model (predicting weight given age) was EHR data from UCSF from 2012-2019. Which of the following are valid test datasets you can use to evaluate the performance of the ML algorithm?

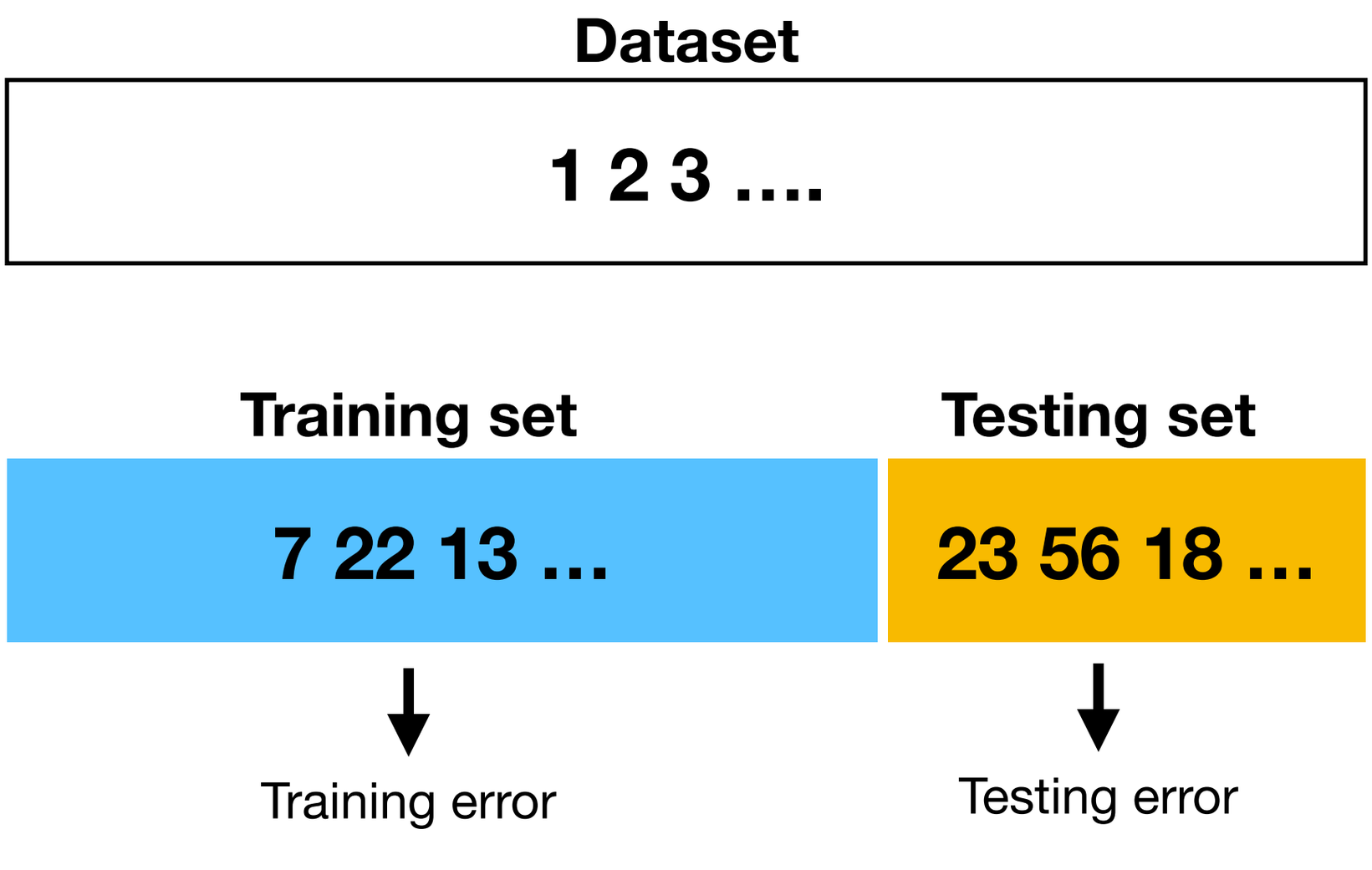
  - **a.** UCSF's EHR data from 2020-2021 ✓
  - **b.** UCSF's EHR data from 2012-2019

    This is okay if you properly split your data.
  - **c.** Stanford's EHR data from 2012-2019

    Be careful about overlapping patients between the two hospital systems!
  - **d.** JHU's EHR data from 2012-2019 ✓ *Probably ok, assuming no overlapping patients*
  - **e.** Prospectively-collected data after the model has been finalized ✓

Training-test split

1. Decide the randomization ratio for the training-test split.
2. Split the data randomly between training and testing.
3. Fit the model on the training set.
4. Evaluate the model on the testing set.



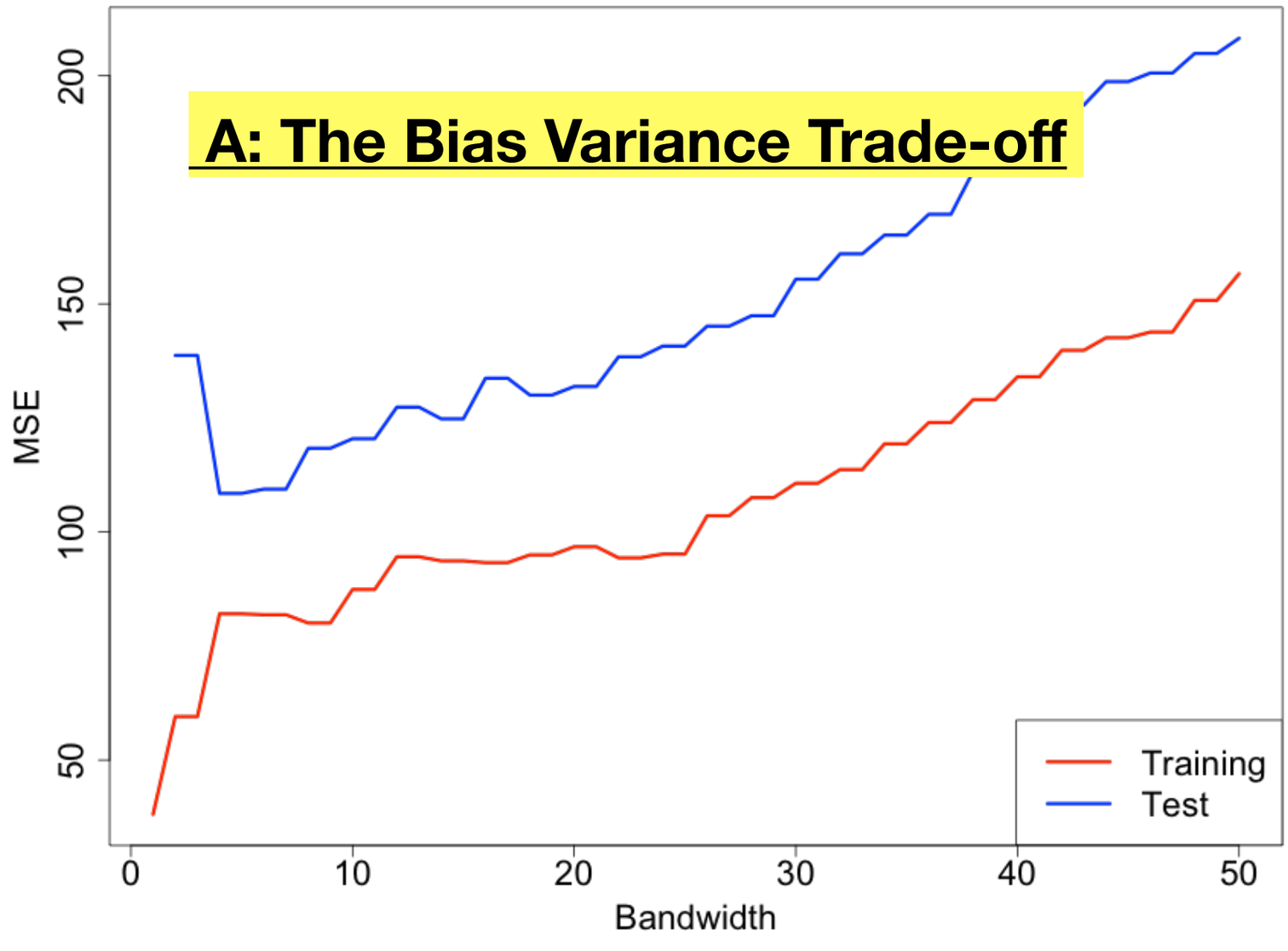
Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. Model selection
    5. **Bias-Variance Trade-off**
    6. Multivariate models

Example: Choosing the bandwidth

Q: Why does the MSE on the test set go down and then up as we decrease the bandwidth?

A: The Bias Variance Trade-off



The **Bias** **Variance** Trade-off

To achieve a small **test MSE**, you want to pick a model that both has low **variance** and a small **bias**.

$$\mathbb{E}\left[(Y - \hat{f}(X))^2 \,|\, X = x\right]$$

$$= Var\left(\hat{f}(x)\right)$$

$$+ \left(\mathbb{E}[Y \,|\, X = x] - \mathbb{E}[\hat{f}(x)]\right)^2$$

$$+ Var\left(Y \,|\, X = x\right)$$

Average test MSE
(if we run our algorithm over different training datasets)

=

Variance of the predicted value

+

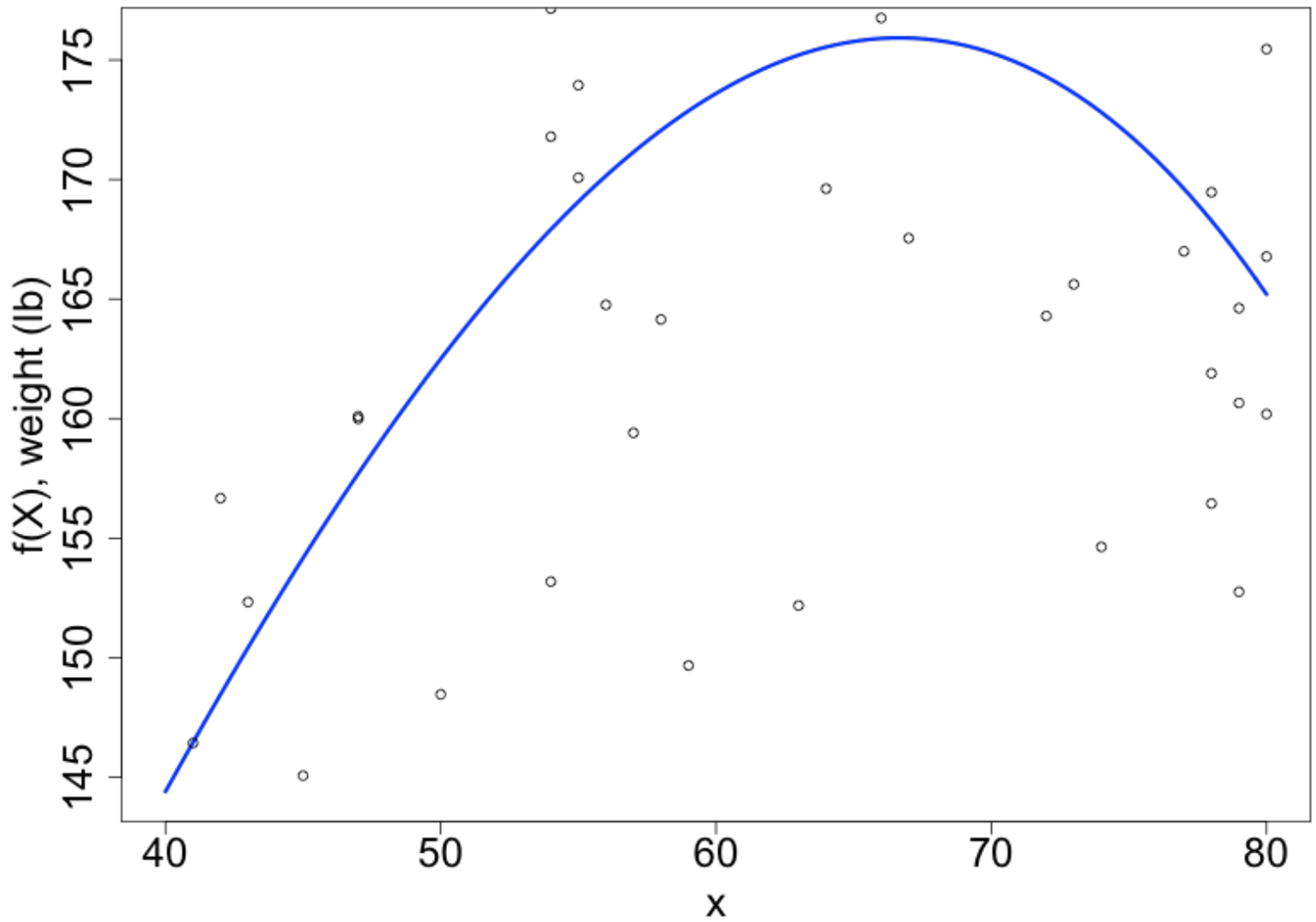
Squared **bias** of the predicted value

+

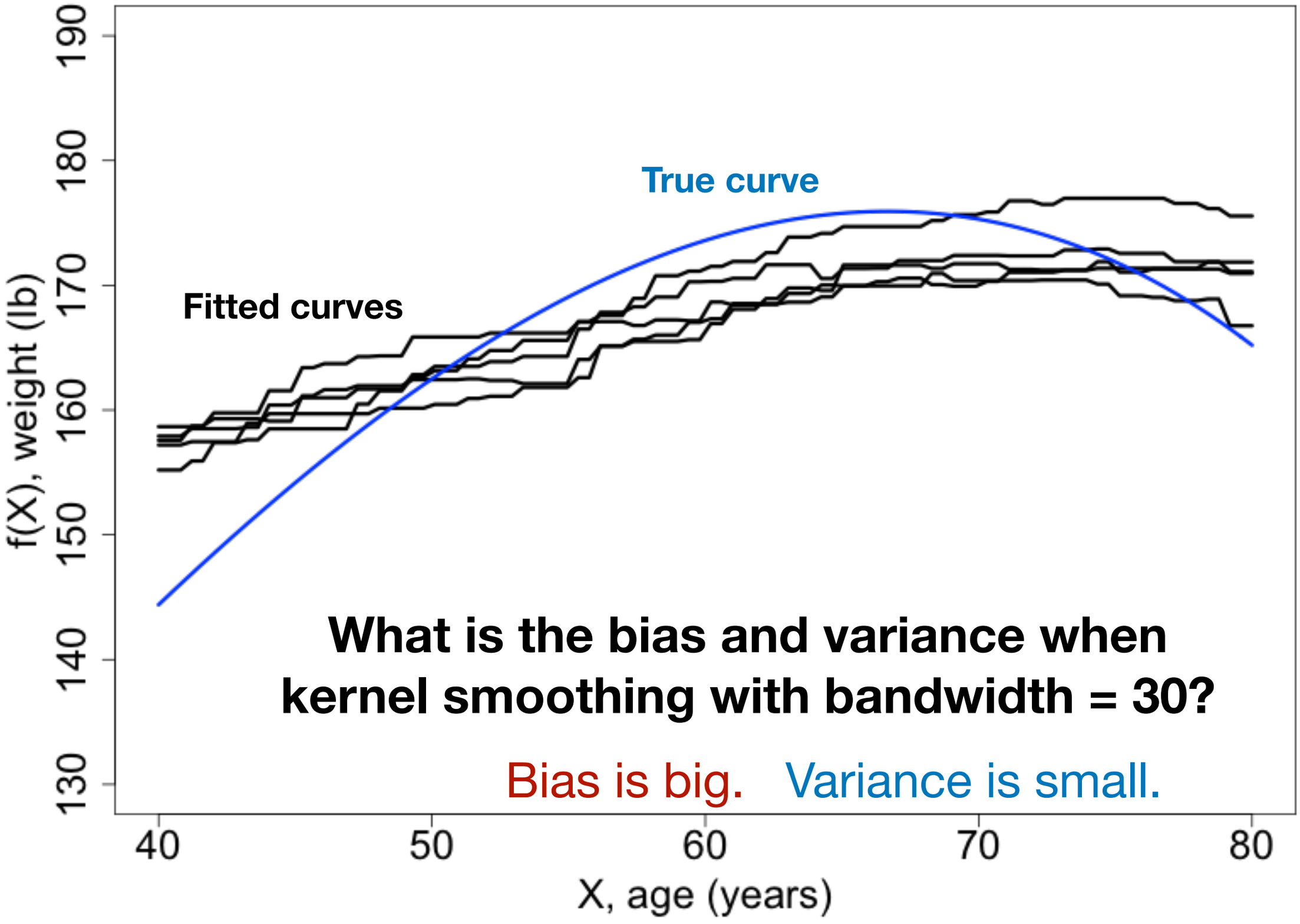
Variance of the outcome at $X = x$
(irreducible error)

The Bias Variance Trade-off: a simulation

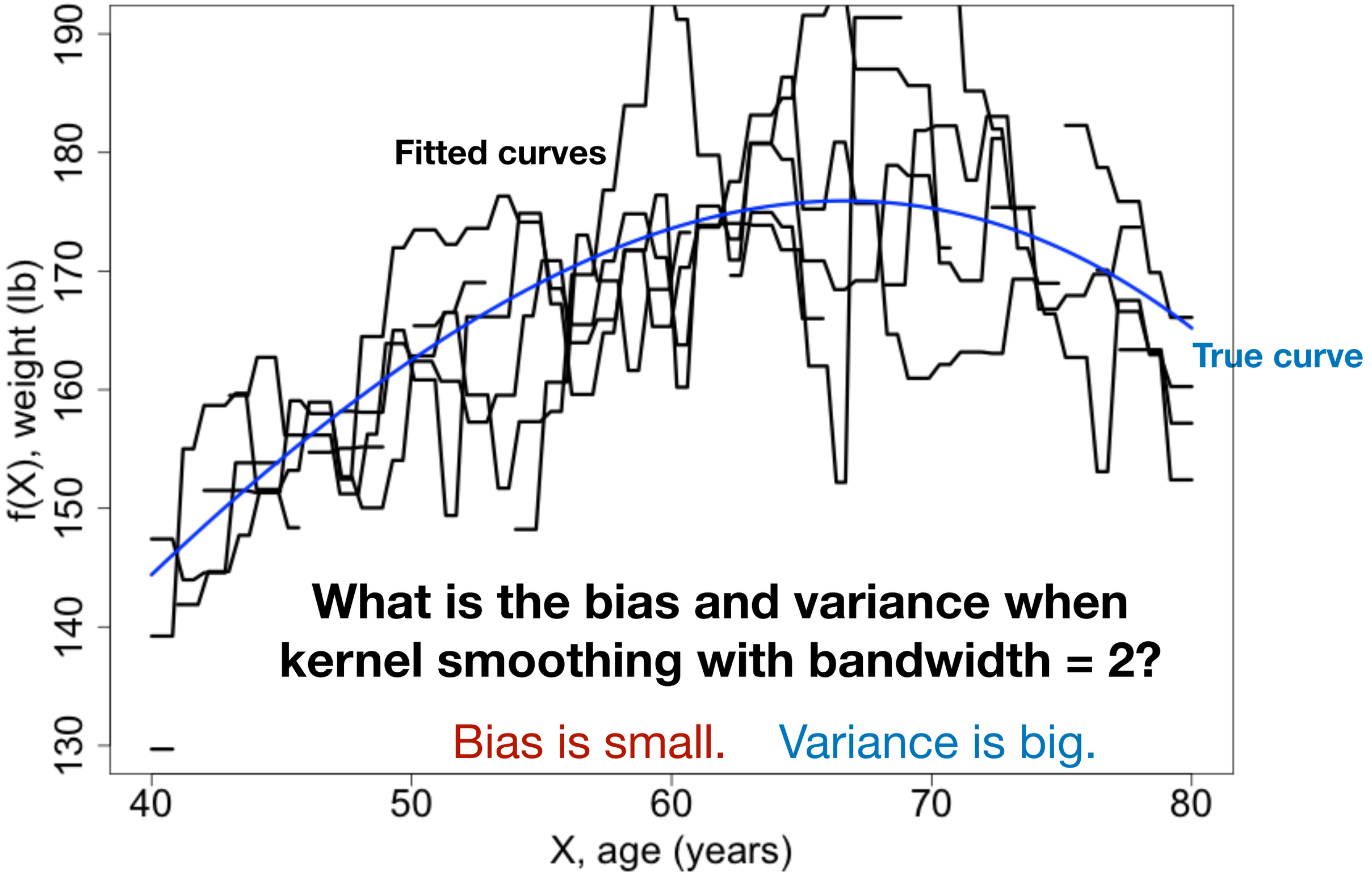
- Let's randomly draw n=50 points from the same model to create m multiple datasets.
- For each dataset, we fit a curve using:
  - Kernel smoother with bandwidth=30
  - Kernel smoother with bandwidth=2



The Bias Variance Trade-off: a simulation

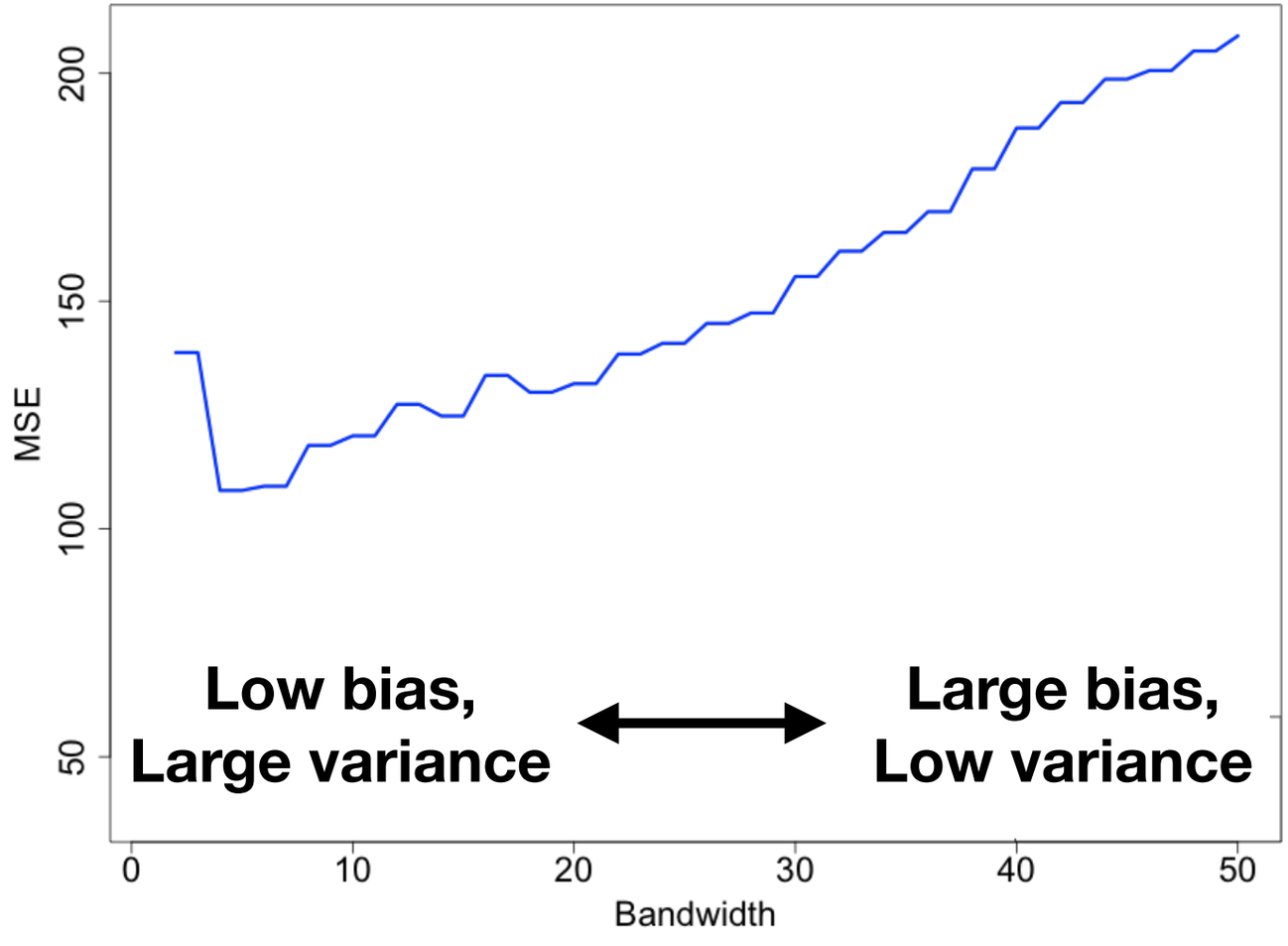


The Bias Variance Trade-off: a simulation



The Bias Variance Trade-off

- Simpler models generally have lower variance but larger bias.
- More complex models generally have larger variance but lower bias.
- To get the smallest test MSE, you need to find the sweet spot.

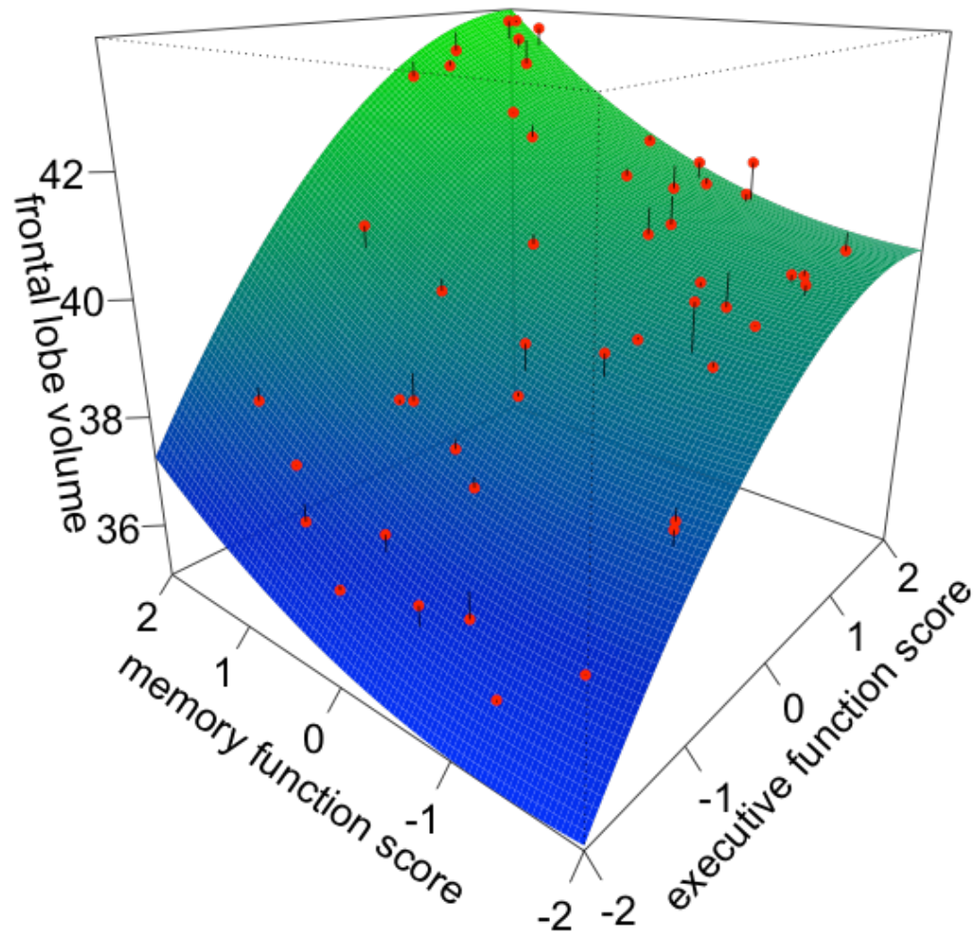


Today's lecture

1. What is Machine Learning?
    1. Supervised vs Unsupervised Learning
2. Supervised learning:
    1. Notation
    2. Evaluating models using a loss function
    3. Parametric and non-parametric models
    4. Model selection
    5. Bias-Variance Trade-off
    6. **Multivariate models**

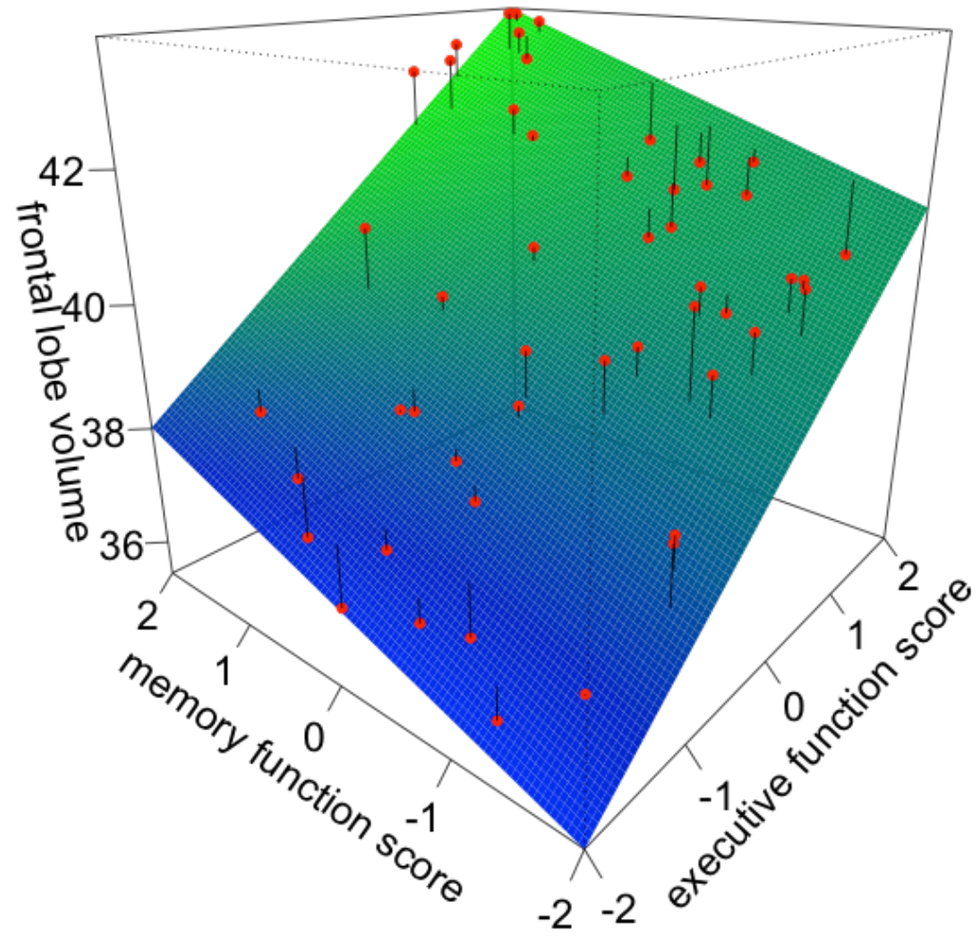
Extending to multiple predictors

- Suppose we have data relating cognitive scores to frontal region brain volume loss:
  - x_1 = memory function score
  - x_2 = executive function score



A multivariate parametric model

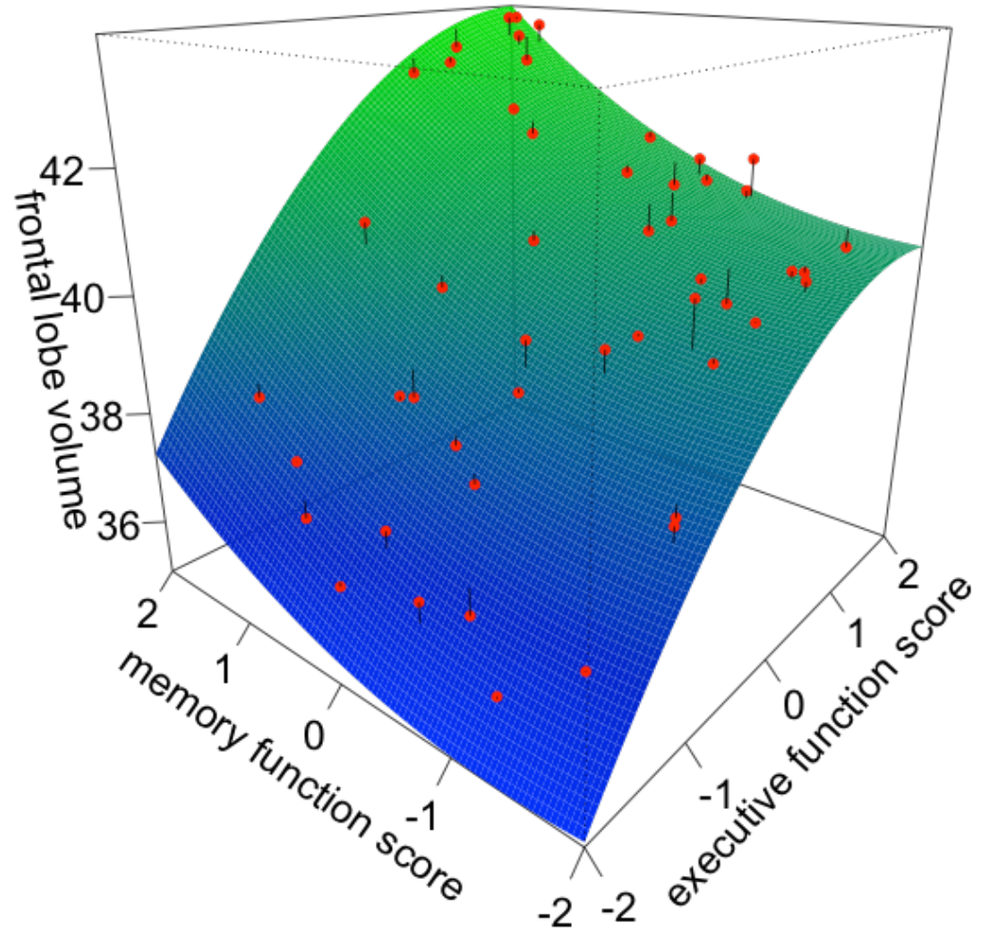
- We can fit a linear model: $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$



A multivariate parametric model

- We can also fit a quadratic model:

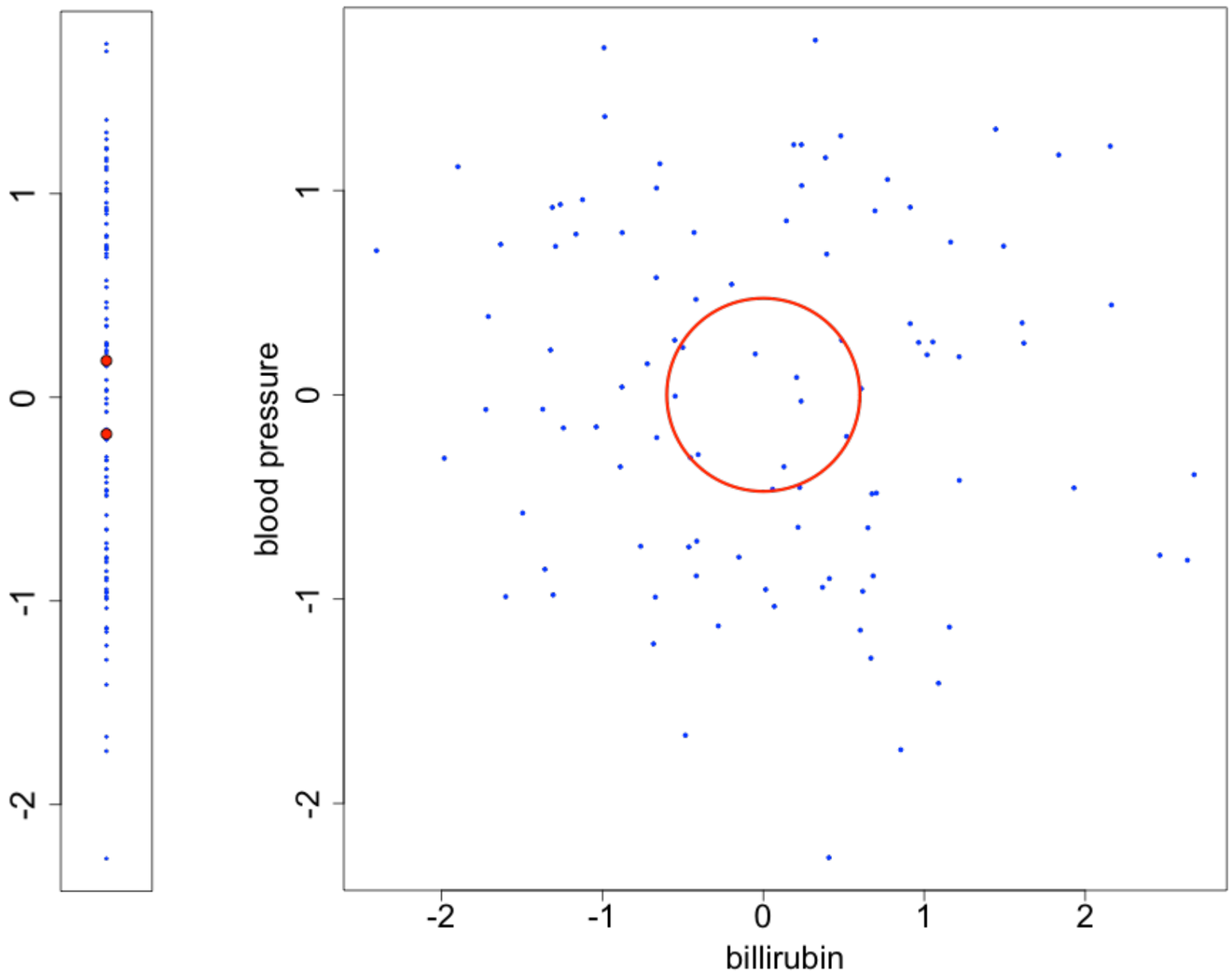
$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2 + \hat{\beta}_4 x_1^2 + \hat{\beta}_5 x_2^2$$



k-Nearest Neighbors

Non-parametric

- A common non-parametric approach is to use k-nearest neighbors (kNN):
- Predict the average response for the K closest points in the training data



Extra materials

- Linear algebra tutorials: Watch Lectures 3.1 to 3.6 of Andrew Ng's Stanford Machine Learning Course on YouTube

    - https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN

- Pages 9-11 of ISLR