Supervised Machine Learning and Learning Theory

Lecture 6: Cross-validation, bootstrap, and subset selection

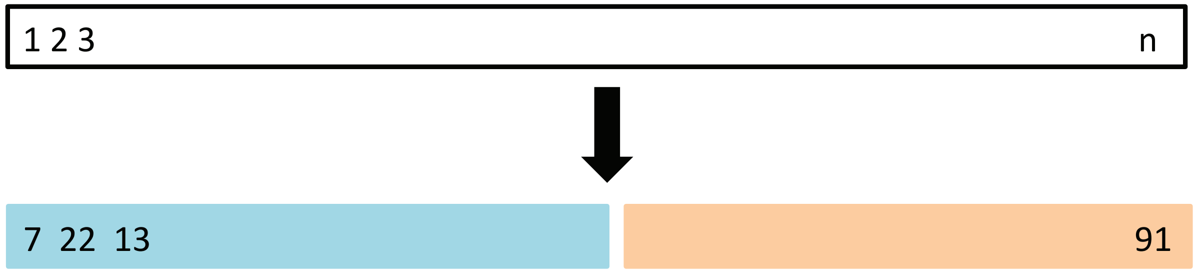
September 24, 2024

Warm-up questions

- What's the difference between the logistic function and logistic loss?
- How could we extend the logit model to multi-class classification?
- What is the mixture of Gaussians model?
- What's the difference between LDA and QDA?
- Why does QDA have a quadratic decision boundary?

Cross validation

- Goal: Using the training dataset alone, find out the test error as closely as possible
- A first attempt: Randomly split the data in two parts; Train the method in the first part, compute the error on the second part



- Issue: loses half the data samples, and the split has a lot of randomness

Example

- Estimate miles per gallon (mpg) from engine horsepower
  - Auto data: horsepower, gas mileage, and other information for 392 vehicles
- **Linear model**

$$\text{mpg} = \beta_0 + \beta_1 \text{horsepower}$$

- **Polynomials**

$$\text{mpg} = \beta_0 + \beta_1 \text{horsepower} + \beta_2 \text{horsepower}^2$$

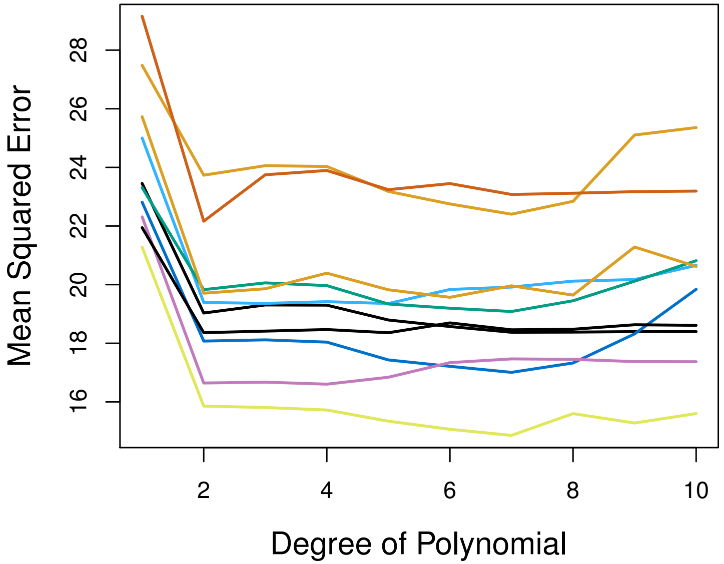
$$\text{mpg} = \beta_0 + \beta_1 \text{horsepower} + \beta_2 \text{horsepower}^2 + \beta_3 \text{horsepower}^3$$

$$\dots$$

- Which polynomial is the right relationship? Partition 392 samples into two sets with equal size; one is the training dataset and the other one is the validation dataset

Example

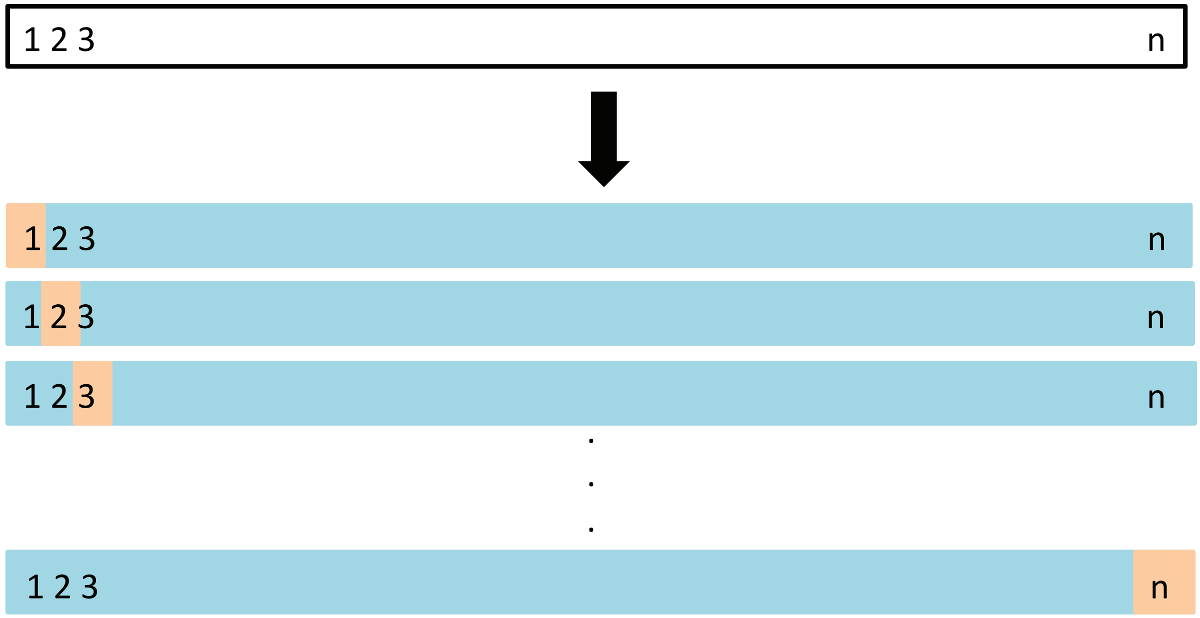
- Estimate miles per gallon (mpg) from engine horsepower



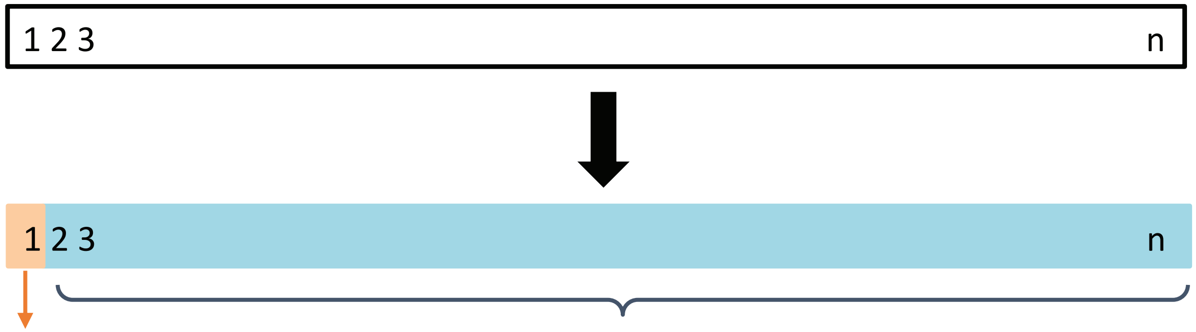
- Each line is the result with a different random split of the data into two parts
- Every split yields a **different** estimate

Leave one out cross-validation

- For every $i = 1, \cdots, n$,
  - Train the model on every point except i
  - Compute the test error on the hold-out point
  - Average over all n points



Leave one out cross-validation

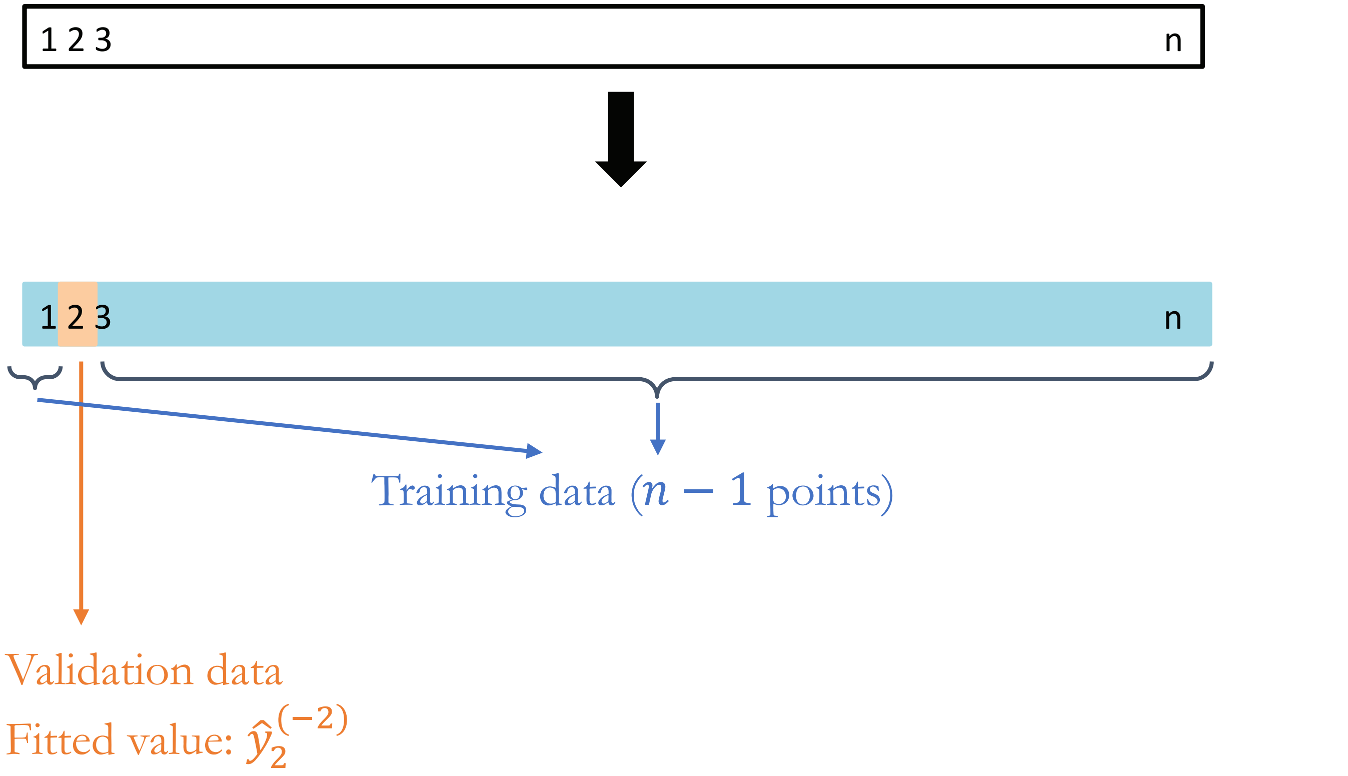


Validation data

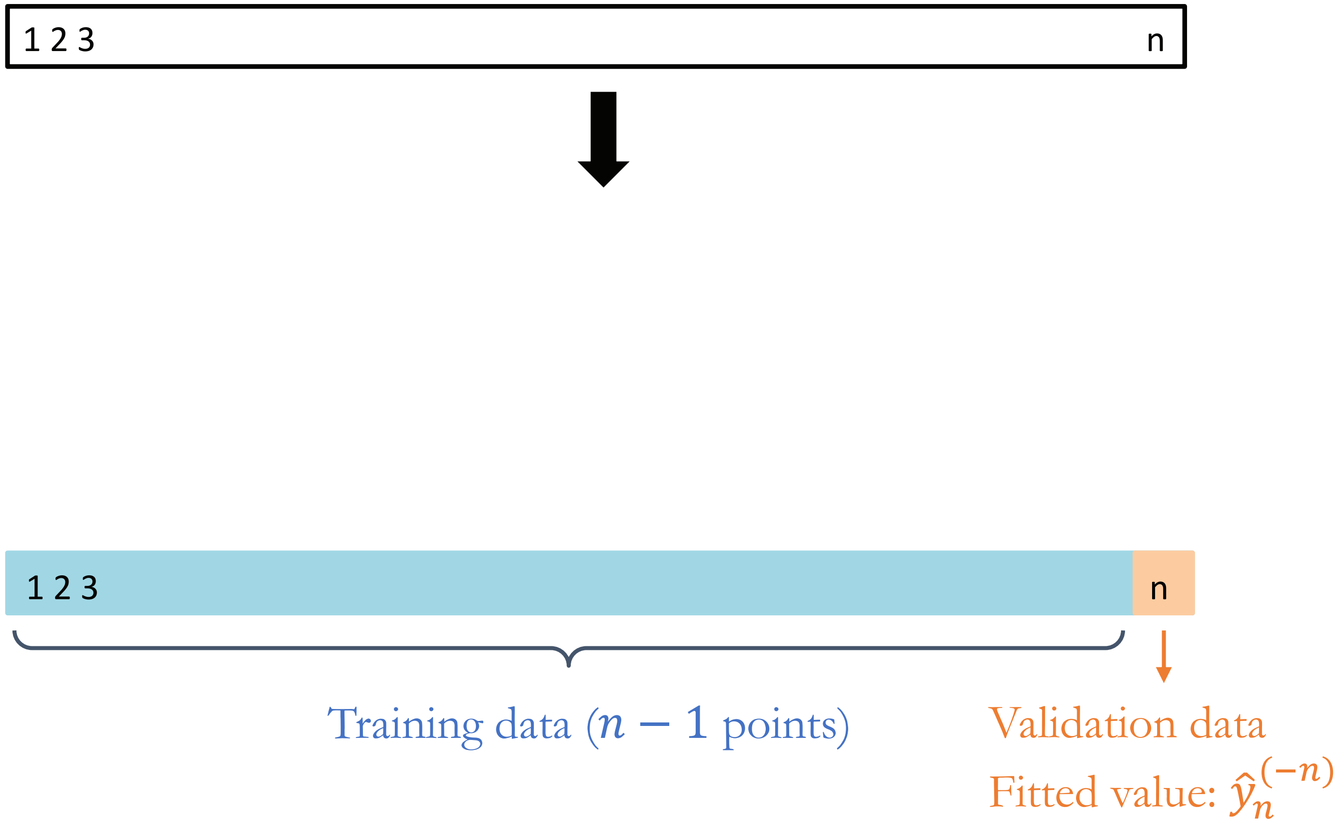
Fitted value: $\hat{y}_1^{(-1)}$

Training data ($n - 1$ points)

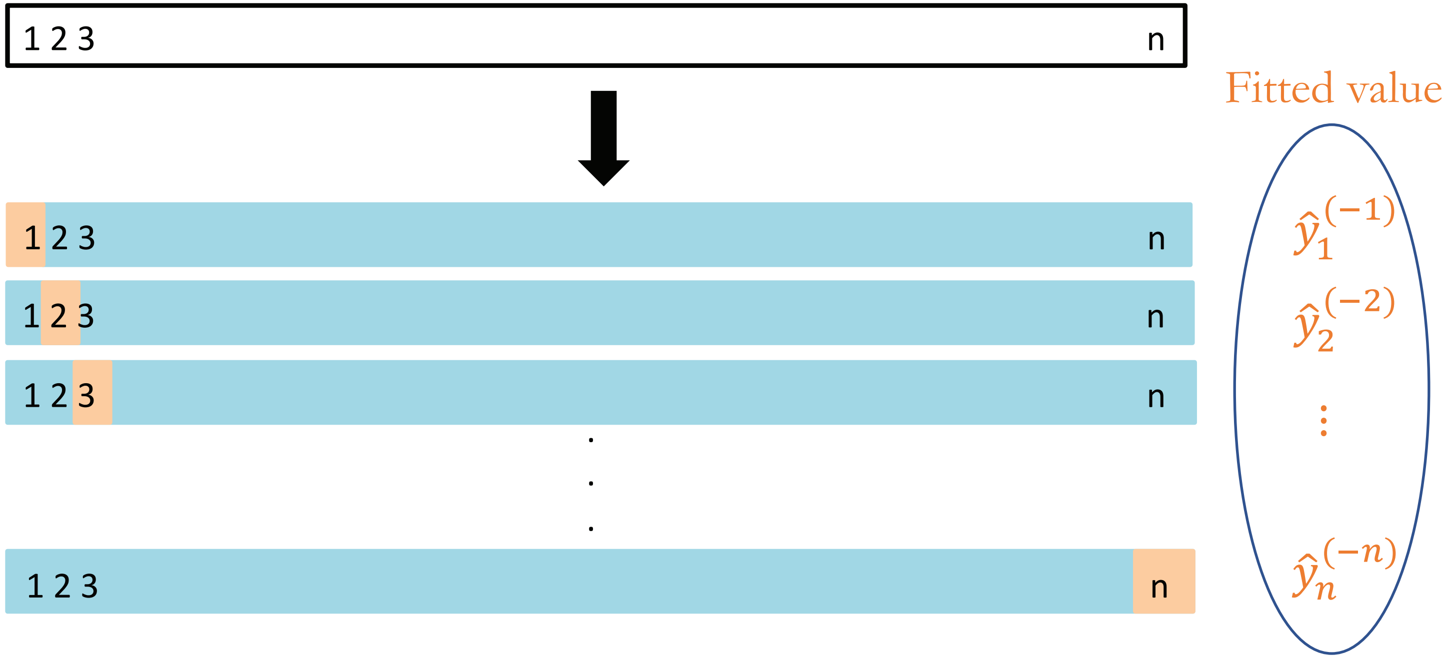
Leave one out cross-validation



Leave one out cross-validation



Leave one out cross-validation



Estimate cross-validation error

LOOCV

- **Regression**
  - $\hat{y}_i^{(-i)}$: Prediction for the ith sample without using the ith sample

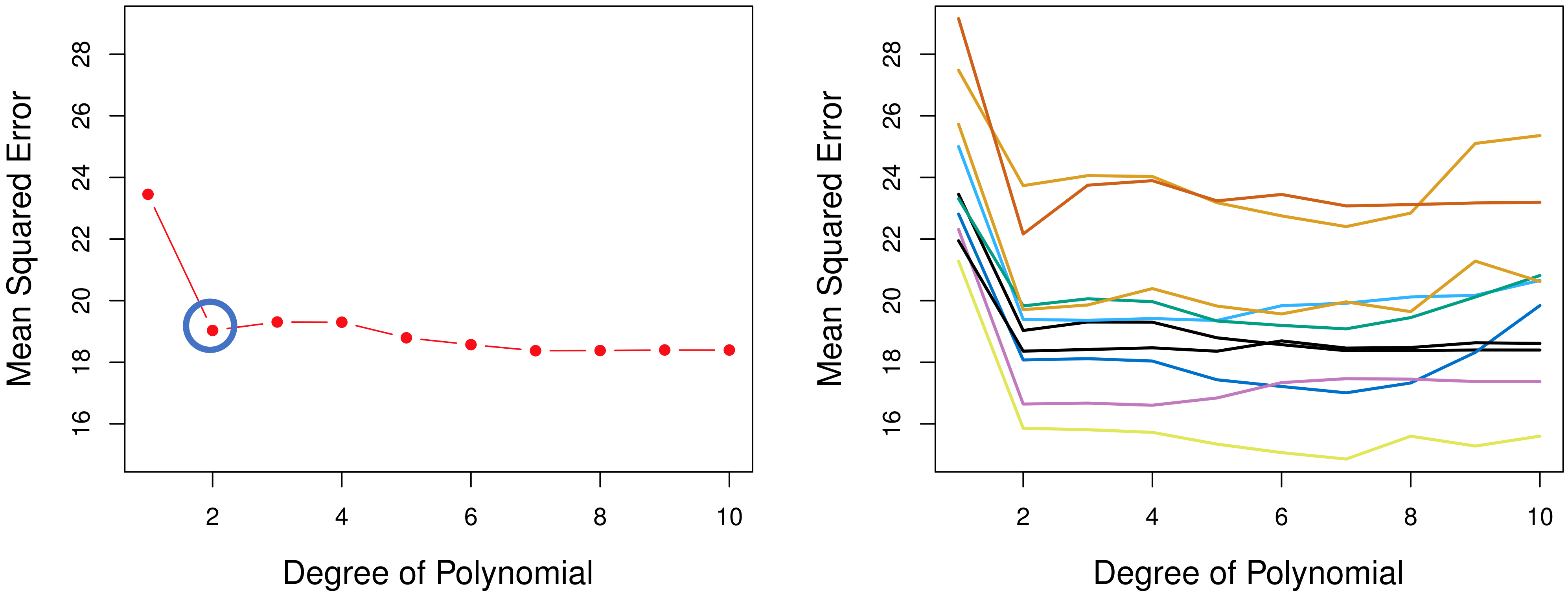
$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{(-i)}\right)^2$$

- **Classification**
  - $\hat{y}_i^{(-i)}$: Prediction for the ith sample without using the i-th sample

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\left[y_i \neq \hat{y}_i^{(-i)}\right]}$$

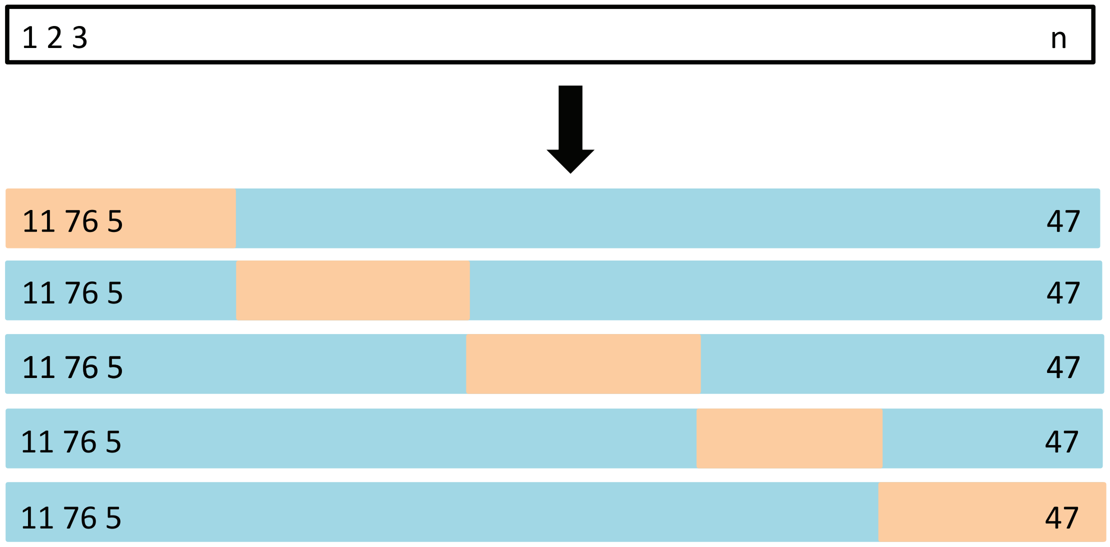
Back to our example

- Estimate miles per gallon (mpg) from engine horsepower
- LOOCV curve vs. random splitting

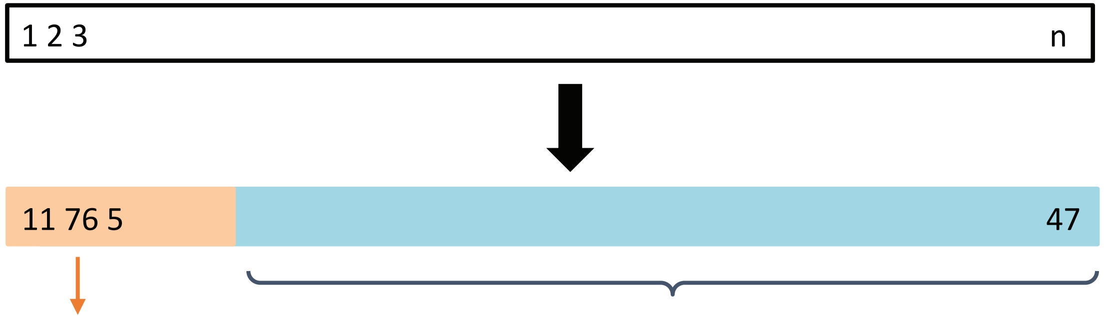


k-fold cross-validation

- Split the data into k subsets/folds
- For every $i = 1, \cdots, k$
  - Train the model on every fold except the i-th fold
  - Compute the test error on the i-th fold
  - Average the test errors



k-fold cross-validation

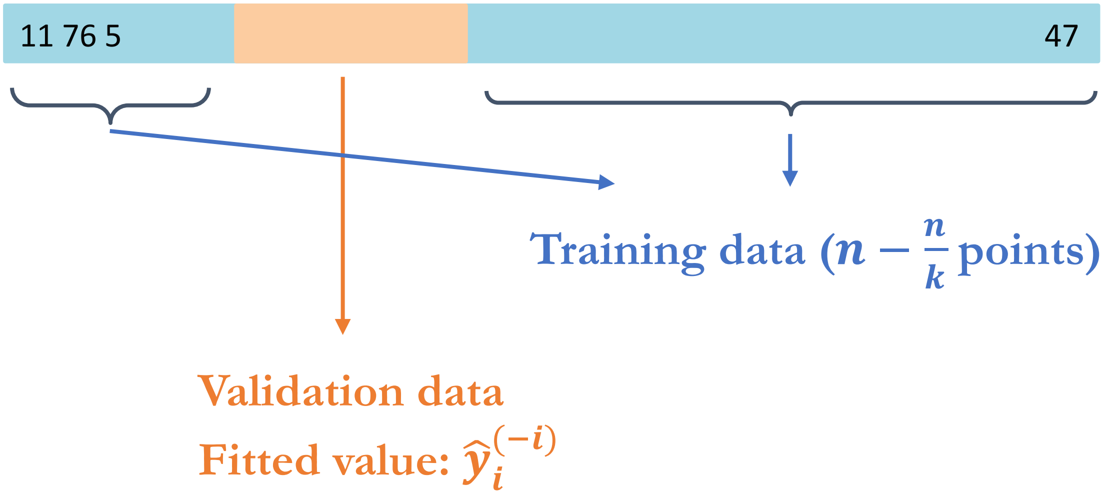


Validation data

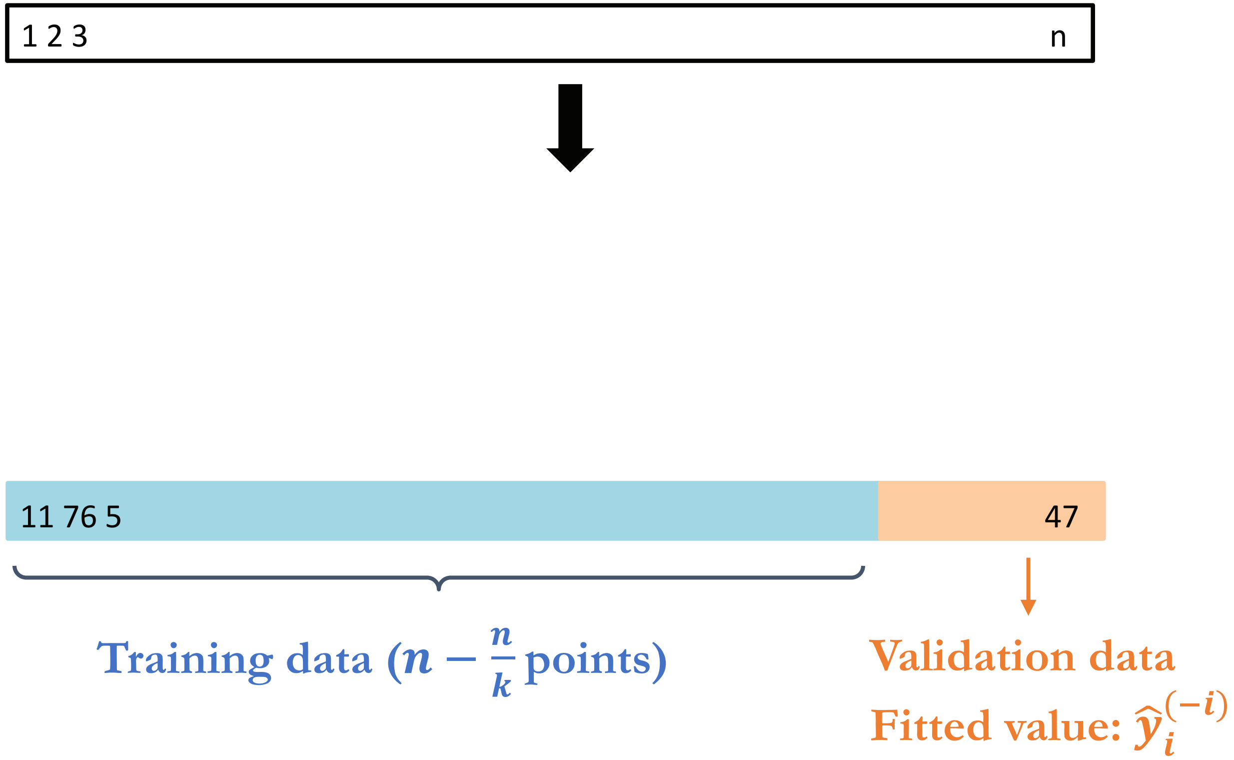
Fitted value: $\hat{y}_i^{(-i)}$

Training data ($n - \frac{n}{k}$ points)

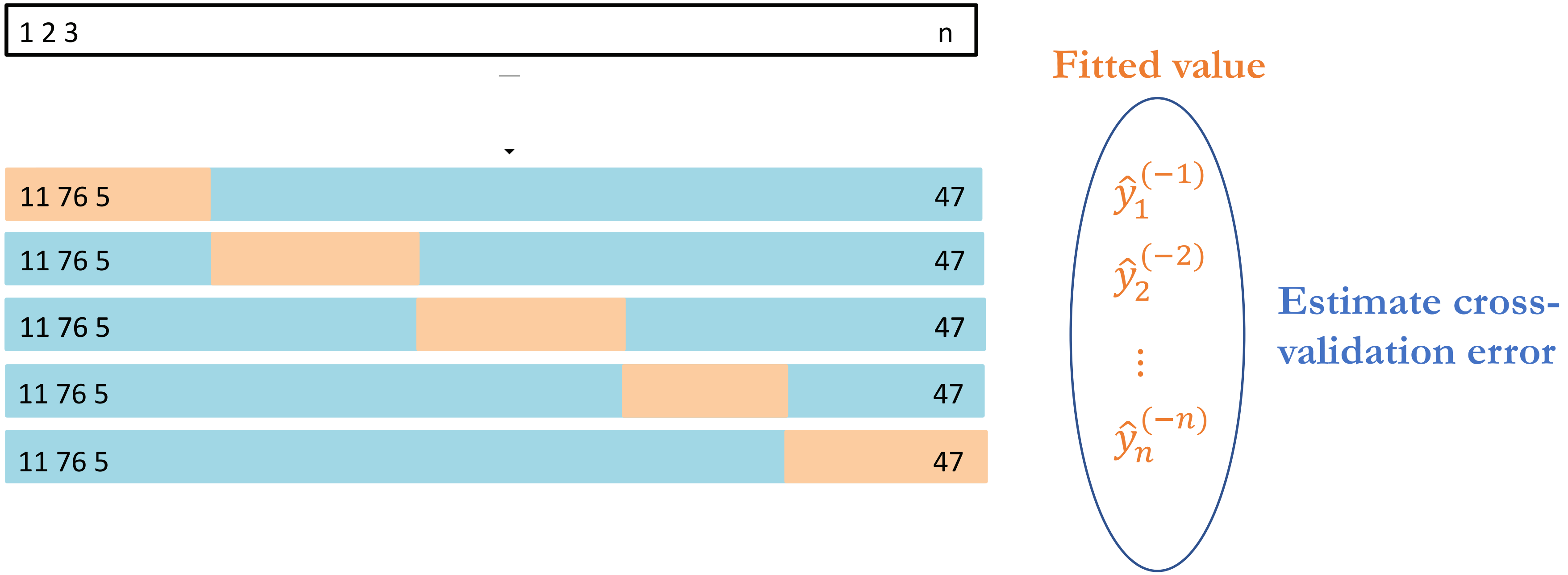
k-fold cross-validation



k-fold cross-validation

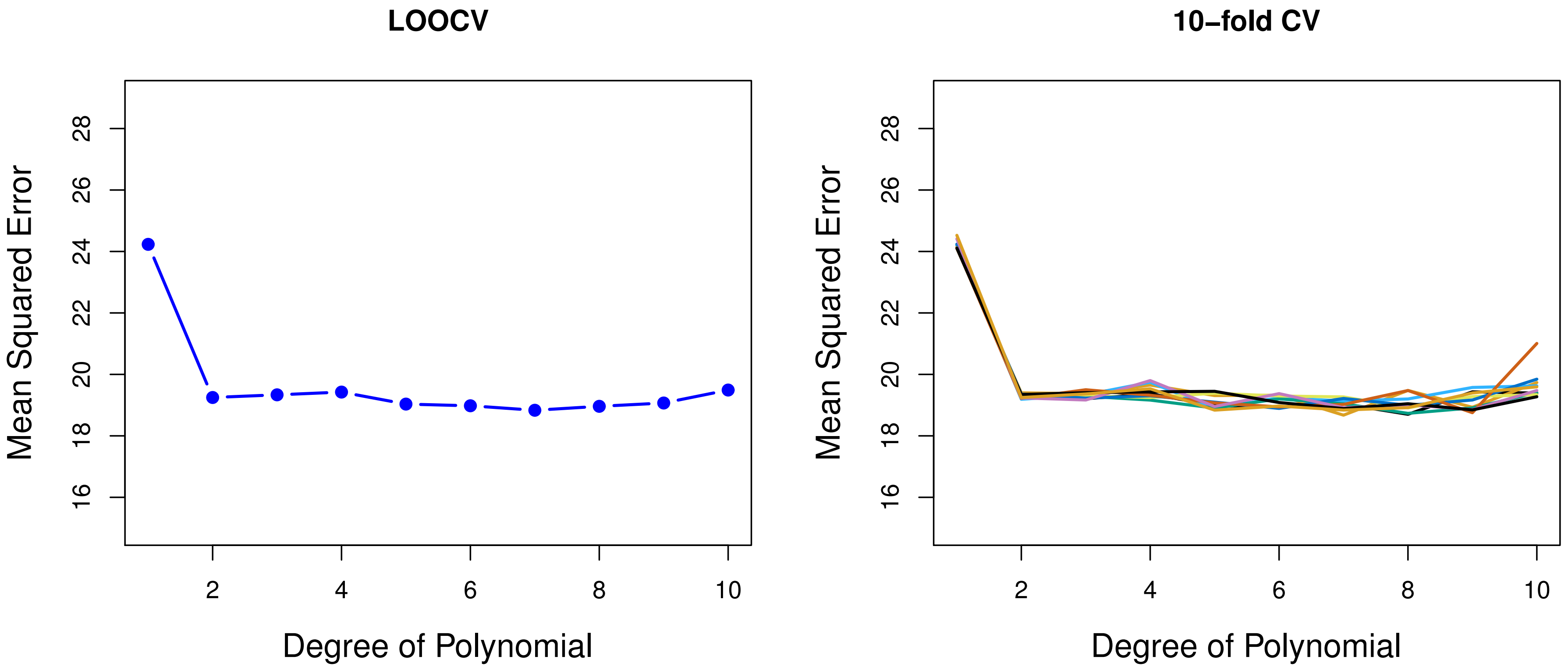


k-fold cross-validation



LOOCV vs. k-fold cross-validation

- Estimate miles per gallon (mpg) from engine horsepower
- The LOOCV error curve vs. ten-fold cross-validation error curve



LOOCV vs. k-fold cross-validation

LOOCV

- Gives approximately unbiased estimates of the test error, as each training dataset contains $n - 1$ observations
- Average of n fitted models, each of which is trained on an almost identical set of observations

k-fold cross-validation

- Each training dataset contains $n - \frac{n}{k}$ observations
- Average of k fitted models that are less correlated with each other (overlapping training observations are $n - \frac{2n}{k}$)
- **Rule of thumb**: Use $k = 5$ or $k = 10$

Lecture plan

- **Bootstrap**

Cross-validation vs. Bootstrap

- Cross-validation: Provide the test error with an independent validation set
- Bootstrap: Provide the standard error of model estimates

```
Residuals:
    Min      1Q  Median      3Q     Max
-15.594  -2.730  -0.518   1.777  26.199

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.761e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-Squared: 0.7406,     Adjusted R-squared: 0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Example

- Investing in two assets: suppose X and Y are the returns of two assets
- These returns are observed every day: $(x_1, y_1), \cdots, (x_n, y_n)$

Example

- A fixed amount of money to invest: α fraction on X and $1-\alpha$ fraction on Y. Expected return: $\alpha X + (1-\alpha)Y$
- Minimize variance: Solve α from the first order derivative $\frac{\partial \operatorname{Var}(\alpha X+(1-\alpha)Y)}{\partial \alpha} = 0$ (exercise)
- Optimum: $\frac{\sigma_Y^2 - \operatorname{Cov}(X,Y)}{\sigma_X^2+\sigma_Y^2-2\operatorname{Cov}(X,Y)}$, σ_X^2 is variance of X, σ_Y^2 is variance of Y, $\operatorname{Cov}(X,Y)$ is covariance between X and Y
- Can approximate these quantities with empirical data

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \widehat{\operatorname{Cov}}(X,Y)}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\operatorname{Cov}}(X,Y)}$$

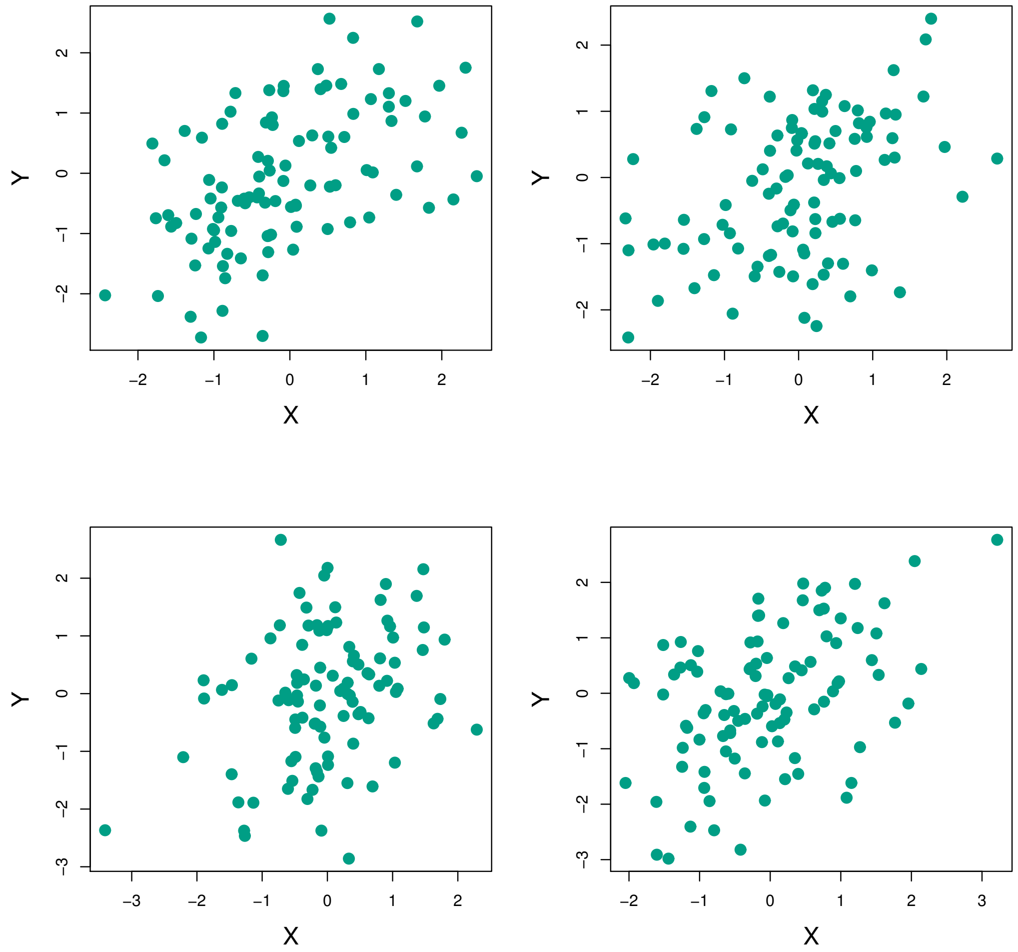
Resampling

- Suppose we compute the estimate $\hat{\alpha} = 0.6$. Do we have some confidence about this? E.g., if we resample the observations, would we get a wildly different $\hat{\alpha}$ (say 0.1)?

- Here we have the joint distribution $Pr(X, Y)$, let's resample the n observations

Resample the X, Y

Resample n observations

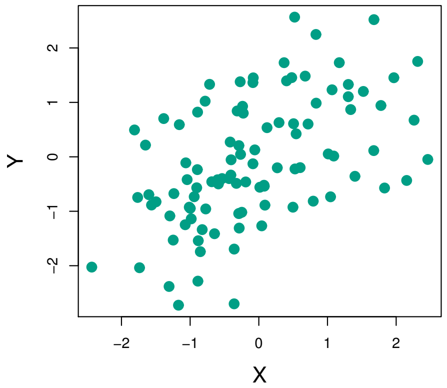


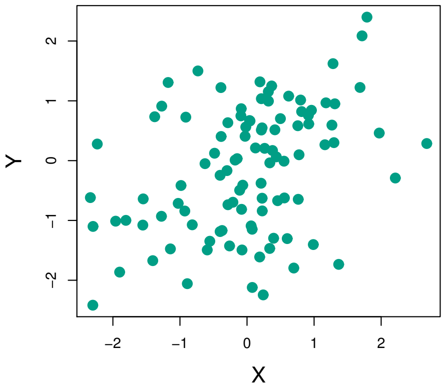
Thought experiment

- Estimate $\hat{\alpha}$ from each sample

$\left(x_1^{(1)}, \cdots, x_n^{(1)}\right)$

Get $\hat{\alpha}^{(1)}$



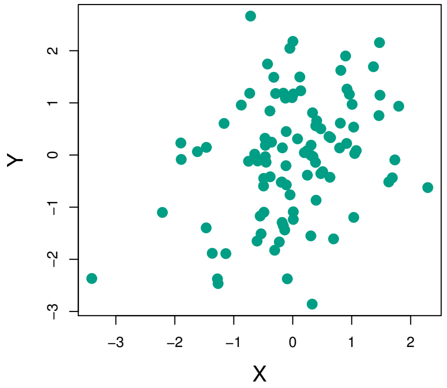


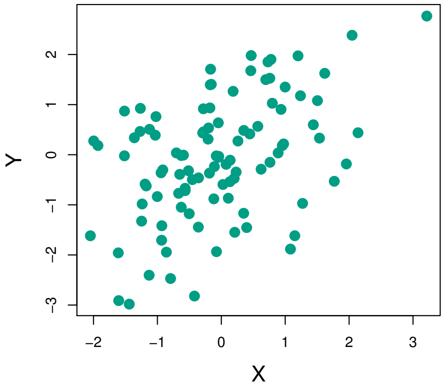
$\left(x_1^{(2)}, \cdots, x_n^{(2)}\right)$

Get $\hat{\alpha}^{(2)}$

$\left(x_1^{(3)}, \cdots, x_n^{(3)}\right)$

Get $\hat{\alpha}^{(3)}$





$\left(x_1^{(4)}, \cdots, x_n^{(4)}\right)$

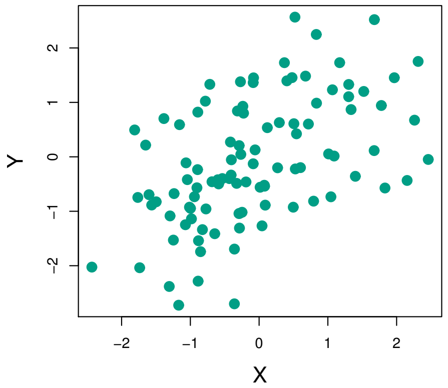
Get $\hat{\alpha}^{(4)}$

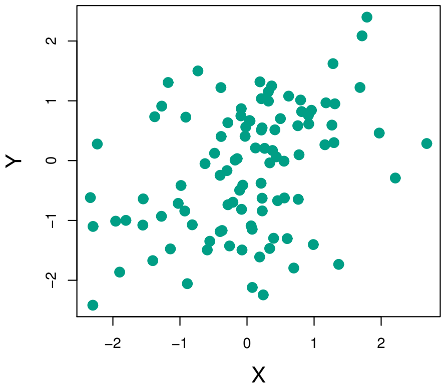
Thought experiment

- **Standard error** of $\hat{\alpha}$ is **approximated** by the **standard deviation** of $\hat{\alpha}^{(1)}, \hat{\alpha}^{(2)}, \hat{\alpha}^{(3)}, \hat{\alpha}^{(4)}, \ldots$

$\left(x_1^{(1)}, \cdots, x_n^{(1)}\right)$

Get $\hat{\alpha}^{(1)}$



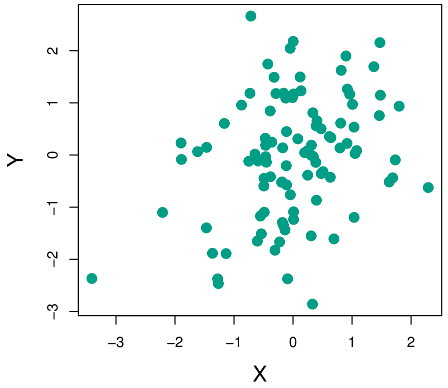


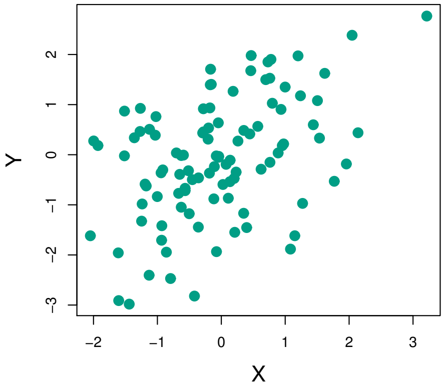
$\left(x_1^{(2)}, \cdots, x_n^{(2)}\right)$

Get $\hat{\alpha}^{(2)}$

$\left(x_1^{(3)}, \cdots, x_n^{(3)}\right)$

Get $\hat{\alpha}^{(3)}$





$\left(x_1^{(4)}, \cdots, x_n^{(4)}\right)$

Get $\hat{\alpha}^{(4)}$

Bootstrap

- In reality, we cannot resample the data. However, we can use the training data set to approximate the joint distribution of X and Y
- **Bootstrap**: Resample the data by drawing n samples **with replacement (meaning that we allow repetitions in them)** from the actual observations

Bootstrap

Original Data (Z)

Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

$\longrightarrow \hat{\alpha}^{*1}$

Z^{*2}

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

$\longrightarrow \hat{\alpha}^{*2}$

$\vdots$

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

$\longrightarrow \hat{\alpha}^{*B}$

A fixed amount of investment: α on X and $1-\alpha$ on Y

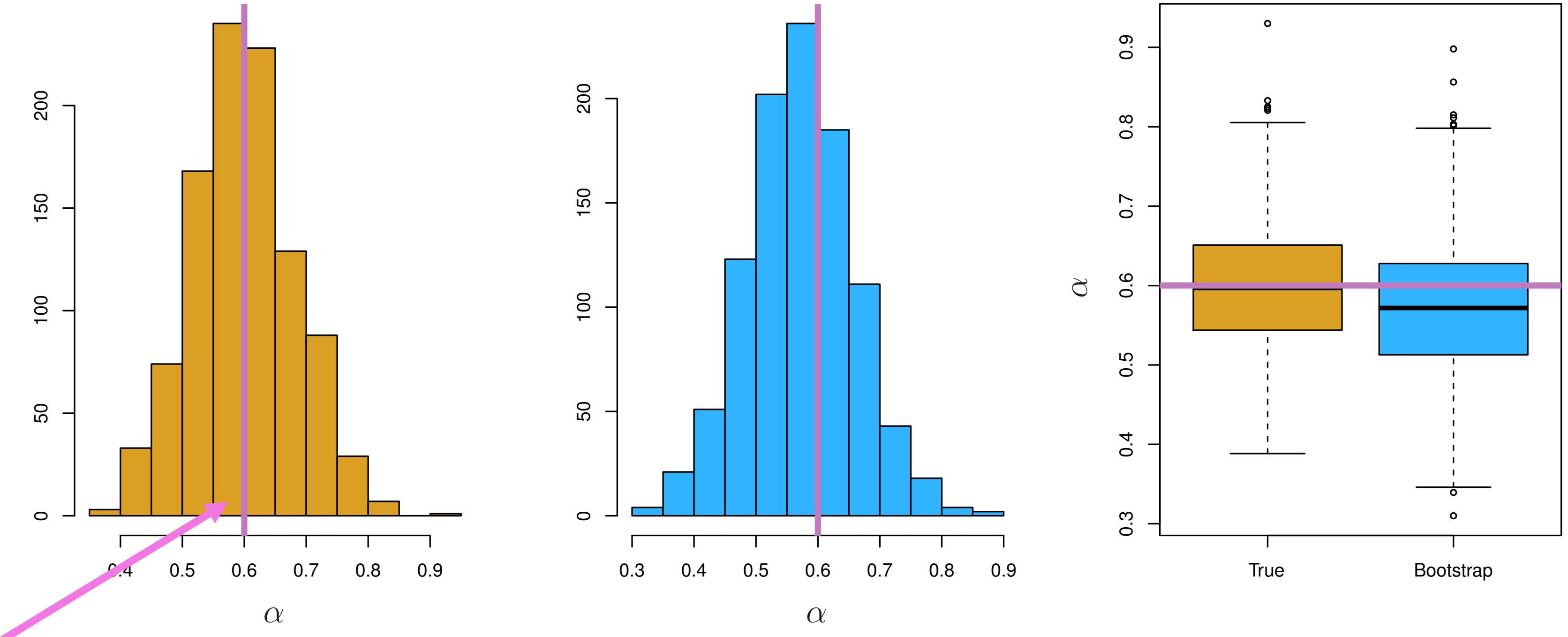
Estimate standard error

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \widehat{\mathrm{Cov}}(X,Y)}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\mathrm{Cov}}(X,Y)}$$

Use standard error of $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}$ to approximate standard error of $\hat{\alpha}$

Bootstrap vs. resampling from the true distribution

Histogram of the estimates of α obtained by generating 1,000 simulated datasets from the true population



True value of α

Histogram of the estimates of α obtained from 1,000 bootstrap samples from a single dataset

Quiz

- In bootstrap, how large is the resampled set?

- How many distinct samples are there in the resampled set (in expectation)?

Lecture plan

- **Subset selection**

Example

Predict whether customers default on their credit card debt with 11 features:

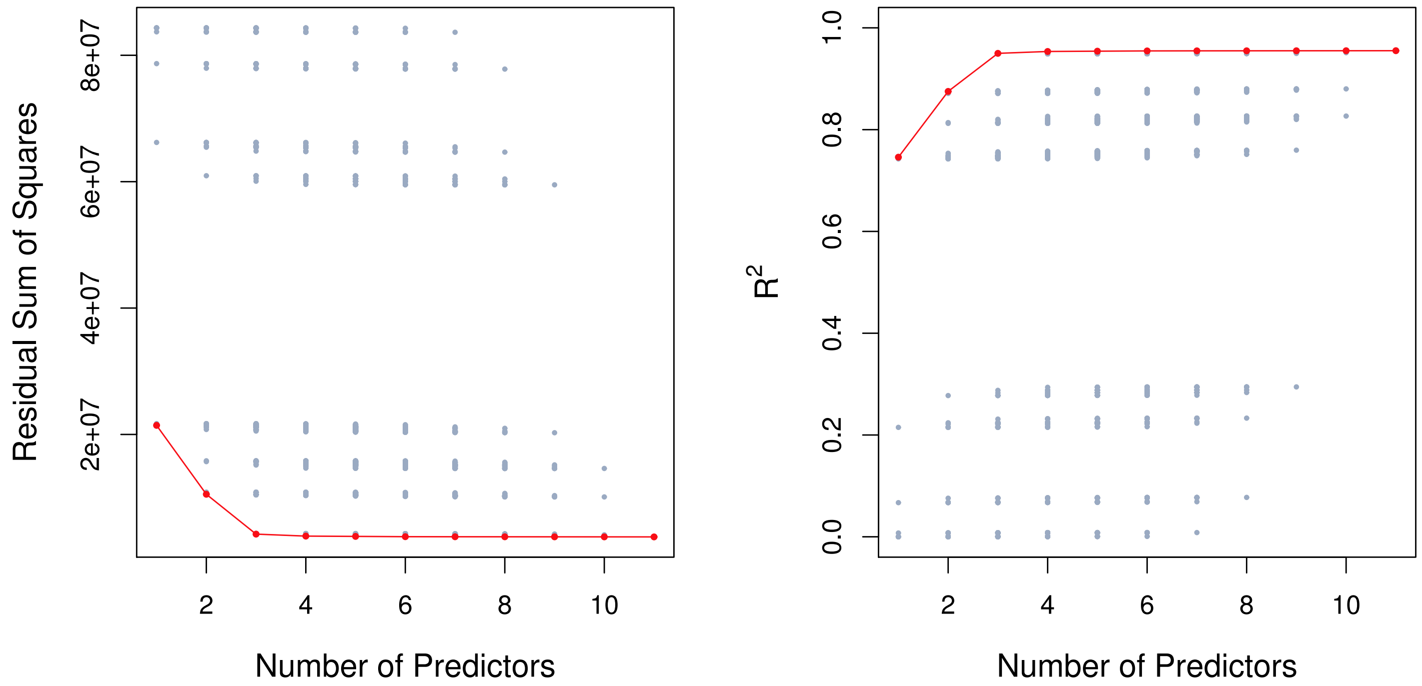
- Income: Income in $1,000's
- Limit: Credit limit
- Rating: Credit rating
- Cards: Number of credit cards
- Age: Age in years
- Education: Number of years of education
- Gender: A factor with levels Male and Female
- Student: A factor with levels No and Yes indicating the individual was a student
- Married: A factor with levels No and Yes indicating whether the individual was married
- Ethnicity: A factor with levels African American, Asian, and Caucasian indicating the individual's ethnicity
- Balance: Average credit card balance in $

Subset selection

- What if not all of the features are useful? How would we select a subset of them (say k)
- Naïve solution: Compare all models with k predictors (and choose one with smallest RSS)
  - Recall that p is the number of predictors ($k \leq p$)
  - There are $\binom{p}{k} = \frac{p!}{k!(p-k)!}$ possible ways of choosing k predictors
  - Doing this for every possible combination is too slow

Example

- Best model for a fixed number of predictors



- Both RSS and R^2 improve as we increase k: Three features suffices

Best subset selection

- How could we find this best subset among 2^k options?

- Cross-validation is one approach to estimate test error, but we still need to enumerate 2^k subsets, which are exponential in k

Forward stepwise selection

- Step 1: No features (fit one model)
- Step 2: Select the best model with one feature (fit p models)
- Step 3: Given the model with one feature, select the best model with two features (fit $p-1$ models)
- Step 4: Given the model with two features, select the best model with three features (fit $p-2$ models)
- …
- In each step, best is defined as having smallest RSS / MSE / highest R^2
- Select a single best model with the optimal number of predictors using cross-validation

Forward stepwise selection

- Step 1: No features (fit one model)
- Step 2: Select the best model with one feature (fit p models)
- Step 3: Given the model with one feature, select the best model with two features (fit $p-1$ models)
- Step 4: Given the model with two features, select the best model with three features (fit $p-2$ models)
- …

Fit $1 + p + (p-1) + \cdots + 1 = 1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ models in total

- Much fewer than $\binom{p}{k}$ (exhaustive enumeration)

Summary: stepwise selection

Forward stepwise selection

- Start with a model with no predictors
- Add predictors to the model one-at-a-time
- Fit $1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ models: Much fewer than $\binom{p}{k}$

Backward stepwise selection is similar but in the reverse direction

- Start with a model with p predictors
- Remove the least useful feature, one at a time

Fit $1 + \frac{p(p+1)}{2}$ models