Prediction, or "Machine Learning"

Econ 2560, Fall 2023

Prof. Josh Abel

(Chapter 14)

Introduction

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 \cdot X_{1,i} + ... + \hat{\beta}_k \cdot X_{k,i}$$

- Have focused intensely on regression coefficients ($\hat{\beta}$s)
  - How are they estimated?
  - Are they statistically significant?
  - What is their interpretation re: conditional mean?
  - Do they have a causal interpretation?
- Will now shift dramatically to focusing on predictions ($\hat{Y}$s)
  - "If I didn't know Y_i, could I predict it well?"
- This "prediction problem" has theory underlying it, but solutions are largely pragmatic rather than theoretical
  - Will work through an example to demonstrate key points
  - Less about a specific tool than a broad approach

Manufactured Dataset

- 200 observations
  - Observations 1-100 are available for model estimation
  - Observations 101-200 are for "out-of-sample testing"
- 99 X variables ($200 \cdot 99$ values)
  - Each X_{ik} value is drawn from N(0,1) and they are all uncorrelated
- 100 β coefficients
  - One for each X regressor, and a constant
  - Each β_k value is drawn from N(0,1) and they are all uncorrelated
- Y is then generated, with noise $u \sim N(0, 25)$

$$Y_i = \beta_0 + \beta_1 \cdot X_{1,i} + ... + \beta_{99} \cdot X_{99,i} + u_i$$

Prediction Problem

- Given observations 1-100...
- ...come up with an algorithm that generates a prediction $\hat{Y}_i$ from observed $X_{1,i}, ..., X_{99,i}$.
- You will then be given observations 101-200...
- ...you will apply the algorithm to the Xs to come up with predictions, $\hat{Y}_{101}, ..., \hat{Y}_{200}$
- Predictions will be evaluated based on "Mean Squared Error":

$$MSE = \frac{1}{100} \sum_{i=101}^{200} (Y_i - \hat{Y}_i)^2$$

- May the best algorithm win!

OLS

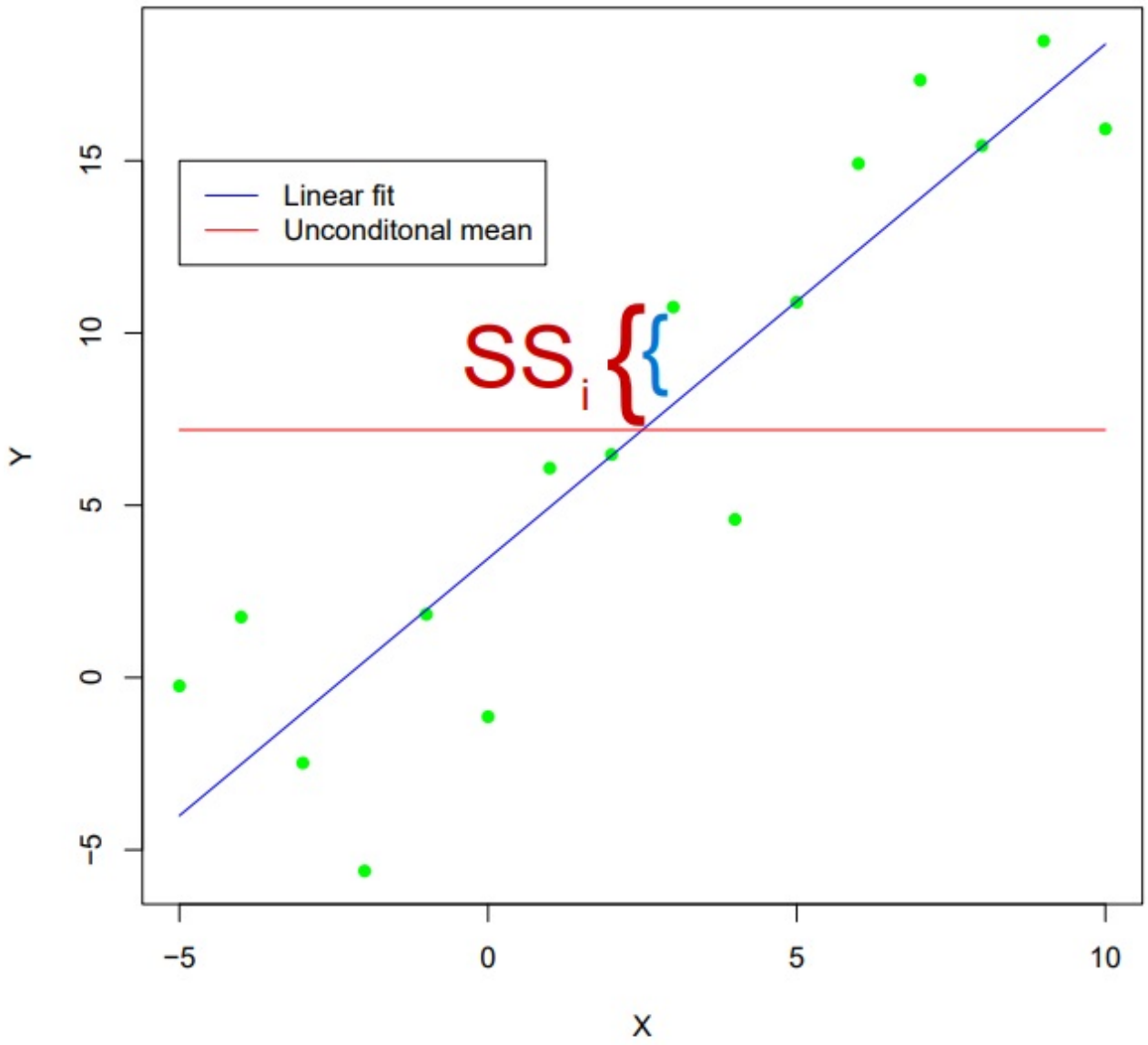
- OLS seems promising here.
- By construction, OLS minimizes sum of squared residuals, which seems helpful:

$$\frac{1}{100} \cdot SSR = \frac{1}{100} \cdot \sum_{i=1}^{100} (Y_i - \hat{Y}_i)^2$$

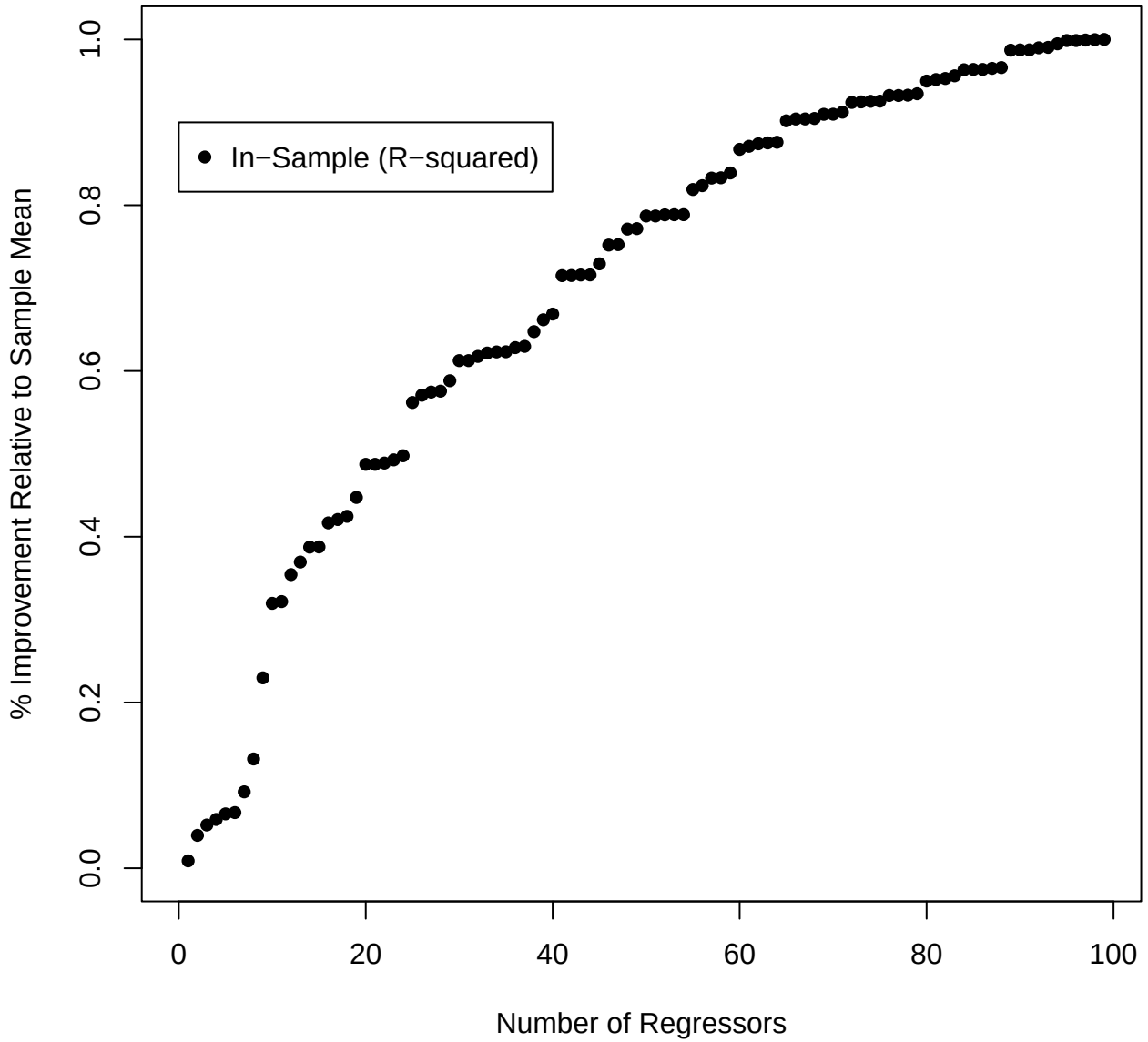
- Maximizes R^2 ("in-sample fit"):

$$R^2 = 1 - \frac{\sum_{i=1}^{100} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{100} (Y_i - \bar{Y})^2}$$

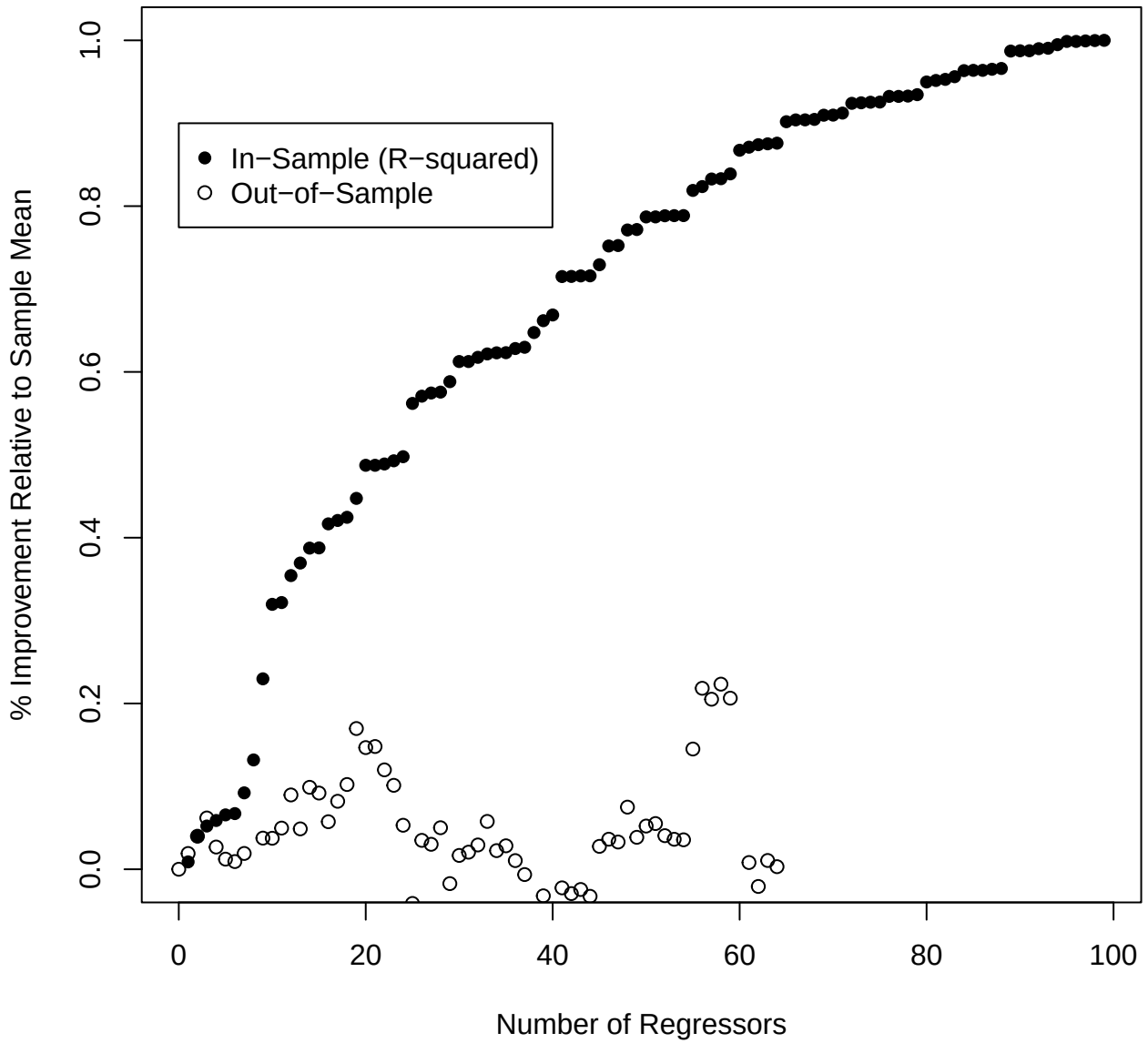
R-squared, Visualized



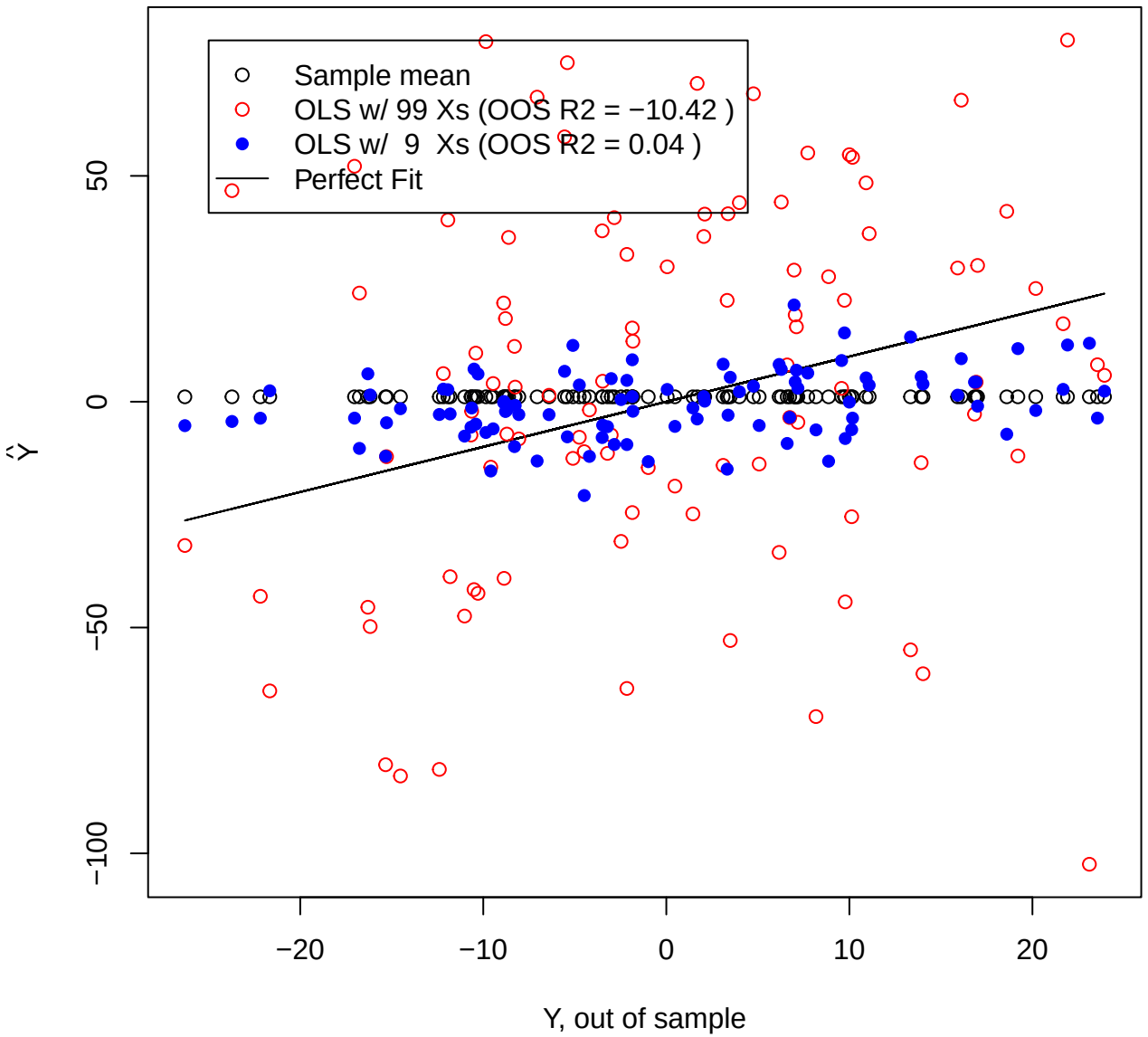
Results: OLS R-squared



Results: OLS Out-of-Sample Fit



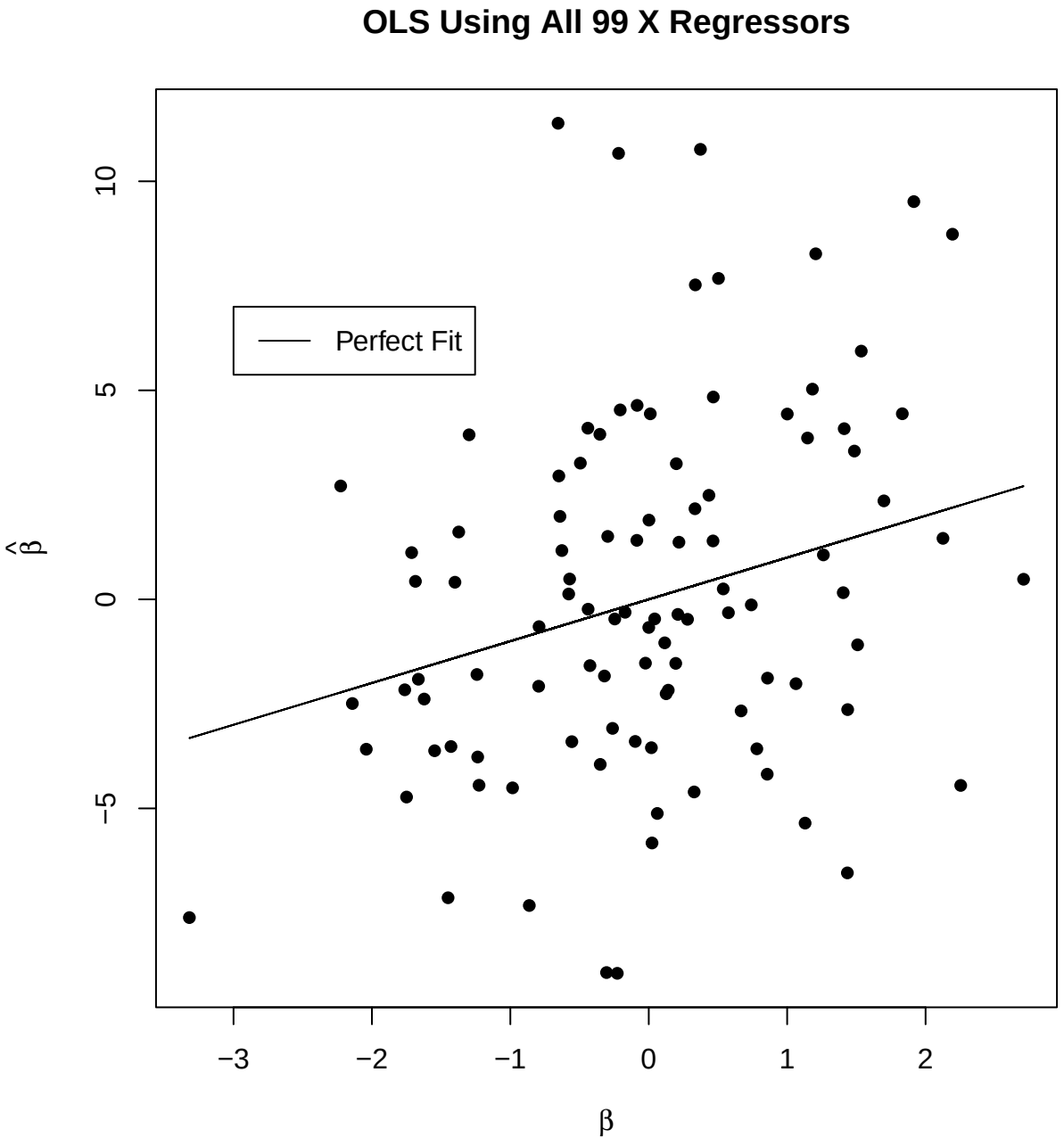
Results: OLS Predictions



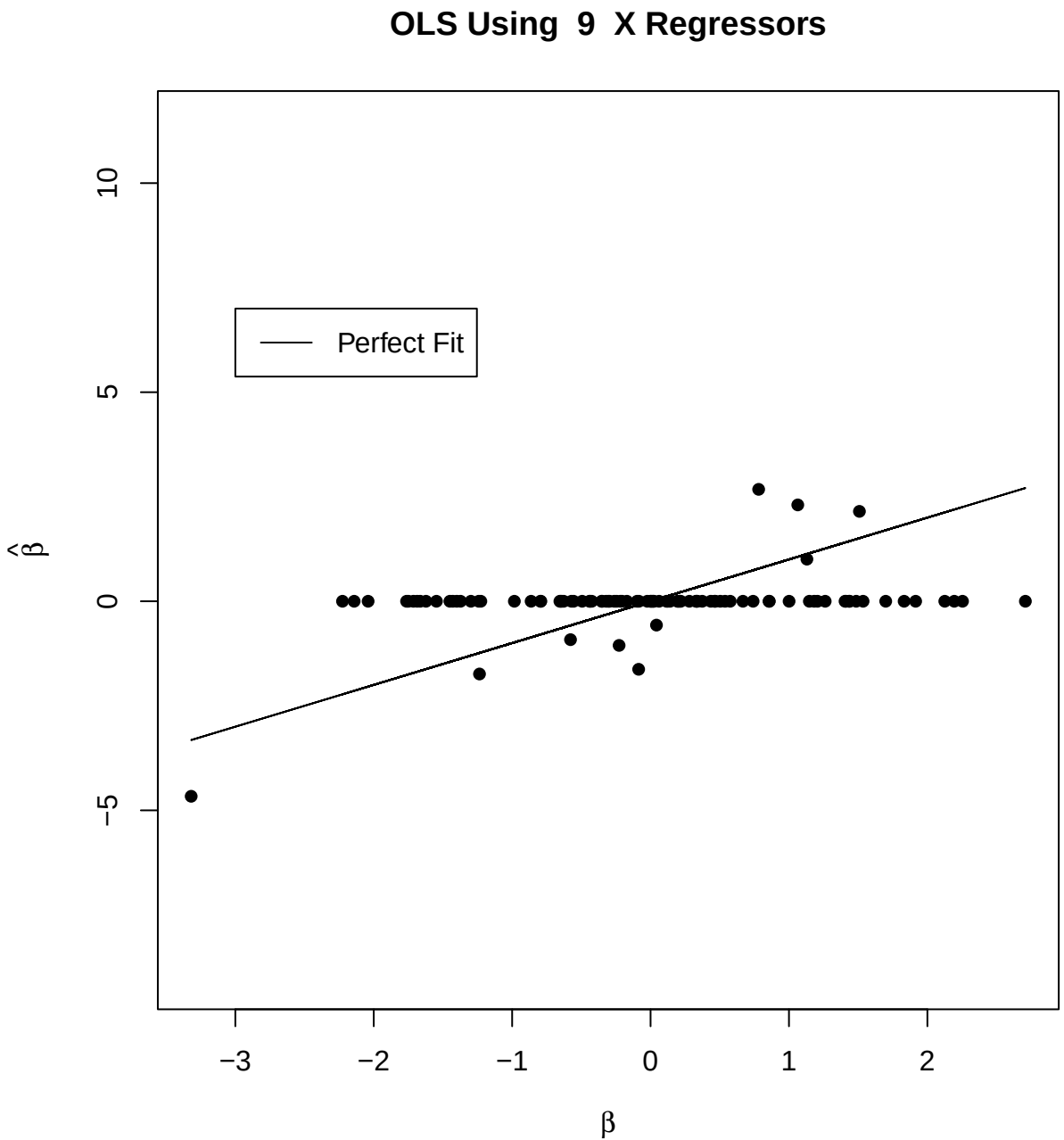
Overfitting

- Using 99 regressors was an example of "overfitting"
- If you give OLS enough coefficients, it will match the data well
  - **In-sample!** This is what it's designed to do.
- Data has two types of variation
  1. Signal (systematic trend)
  2. Noise (unpredictable details)
- With few coefficients, OLS can only try to explain general trend
- But with many coefficients, it gets more ambitious – "explains" the noise!
  - It finds patterns that do not actually exist
  - This leads to a disaster out-of-sample, because those old patterns will not exist in the new data

Results: OLS Coefficients



Results: OLS Coefficients



Bias-Variance Tradeoff

- With many regressors, prediction is all about Bias-Variance Tradeoff
- When we used only 9 regressors, we "assumed" $\beta_{10} = ... = \beta_{99} = 0$
  - Totally arbitrary/wrong ("bias")
- But it led to a big improvement by drastically reducing variance
  - Coefficients far less erratic

Shrinkage

- Arbitrarily dumping 90% of our regressors led to a big improvement, due to reduced variance
- "Machine Learning" techniques are more-systematic, less-arbitrary ways of doing the same thing
- These are broadly known as "shrinkage estimators"
  - Keep all regressors, but shrink the prediction back toward $\bar{Y}$

LASSO

(**L**east **A**bsolute **S**hrinkage and **S**election **O**perator)

Choose $\hat{\beta}$s to minimize:

$$\underbrace{\sum_{i=1}^{n_1=100}\left(Y_i - \sum \beta_k \cdot X_{k,i}\right)^2}_{\text{sum of squared residuals}} + \underbrace{\lambda \cdot \sum_{k=0}^{K} |\beta_k|}_{\text{penalty}}$$

LASSO

(**L**east **A**bsolute **S**hrinkage and **S**election **O**perator)

Choose $\hat{\beta}$s to minimize:

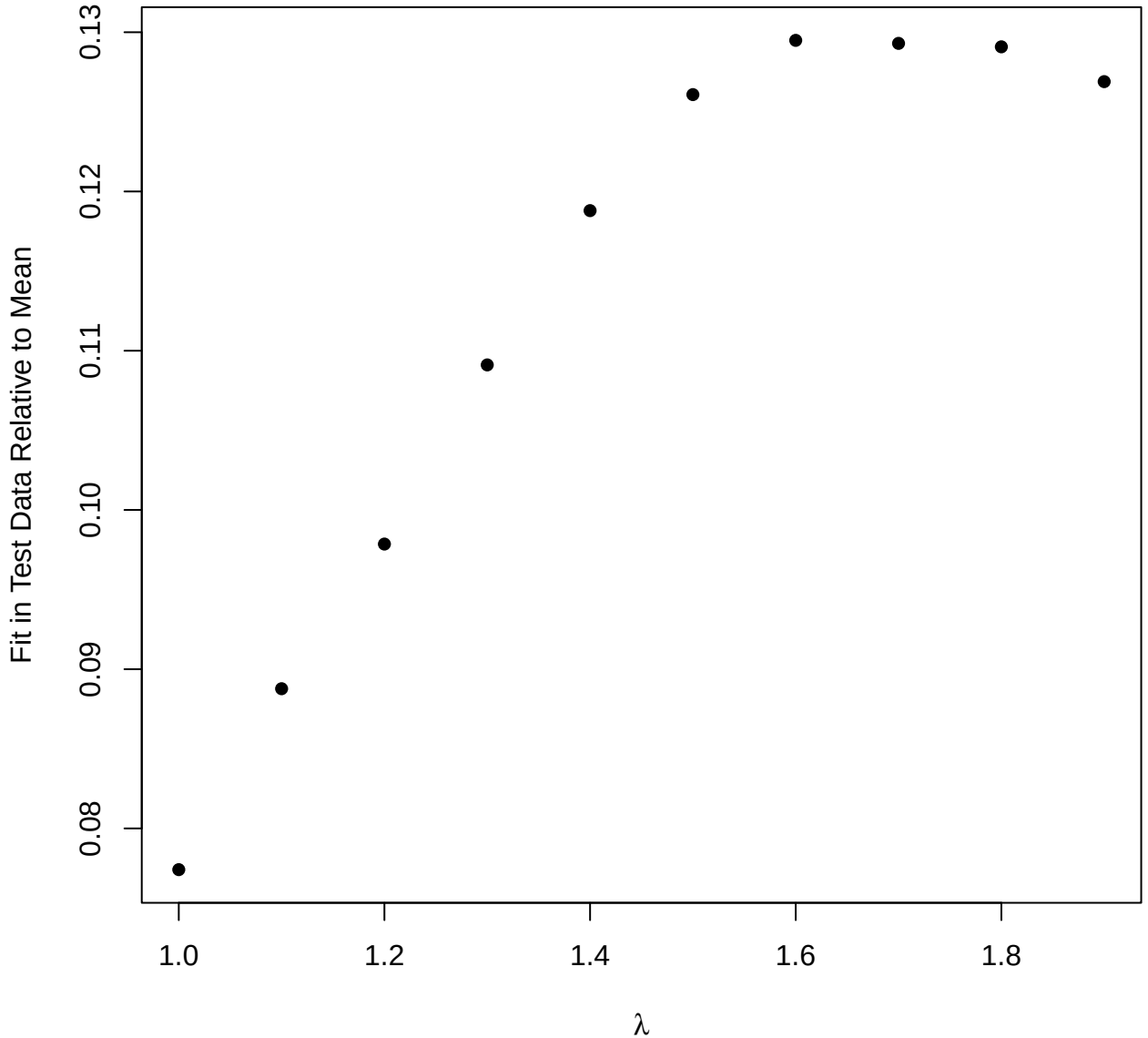
$$\underbrace{\sum_{i=1}^{n_1=100}\left(Y_i - \sum \beta_k \cdot X_{k,i}\right)^2}_{\text{sum of squared residuals}} + \underbrace{\lambda \cdot \sum_{k=0}^{K} |\beta_k|}_{\text{penalty}}$$

"Try to match the data...but don't try too hard."

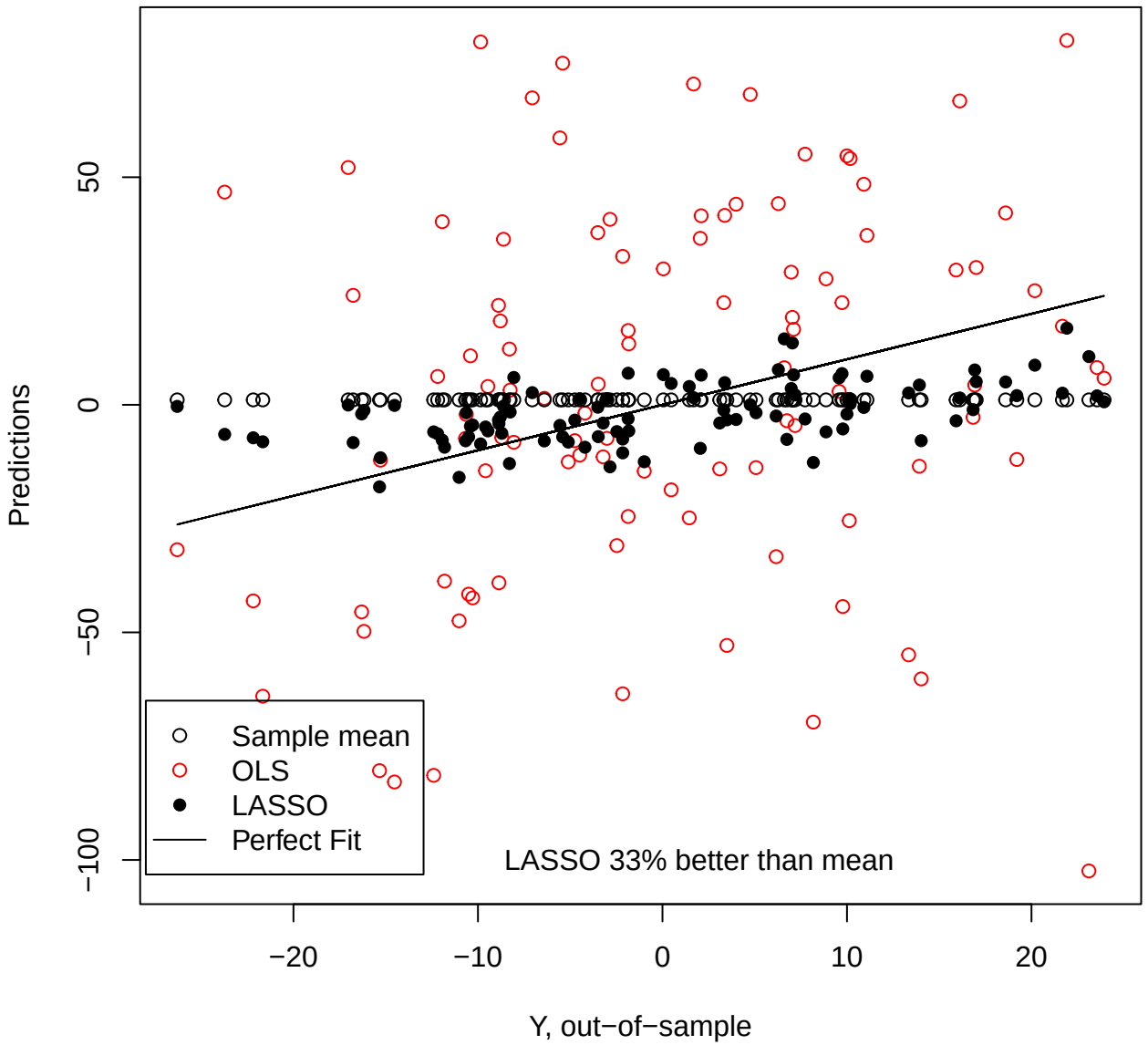
Training Data

- Abandoning OLS opens up Pandora's Box
- There are many shrinkage estimators, not just LASSO
  - Even with LASSO, we still have to choose λ
- Will do the following:
  1. Estimate LASSO for different λs on observations 1-50
  2. Evaluate performance on observations 51-100
  3. Choose the λ with best "pseudo out-of-sample" performance
  4. Re-estimate LASSO with chosen λ on observations 1-100
  5. Use that to predict Y for observations 101-200

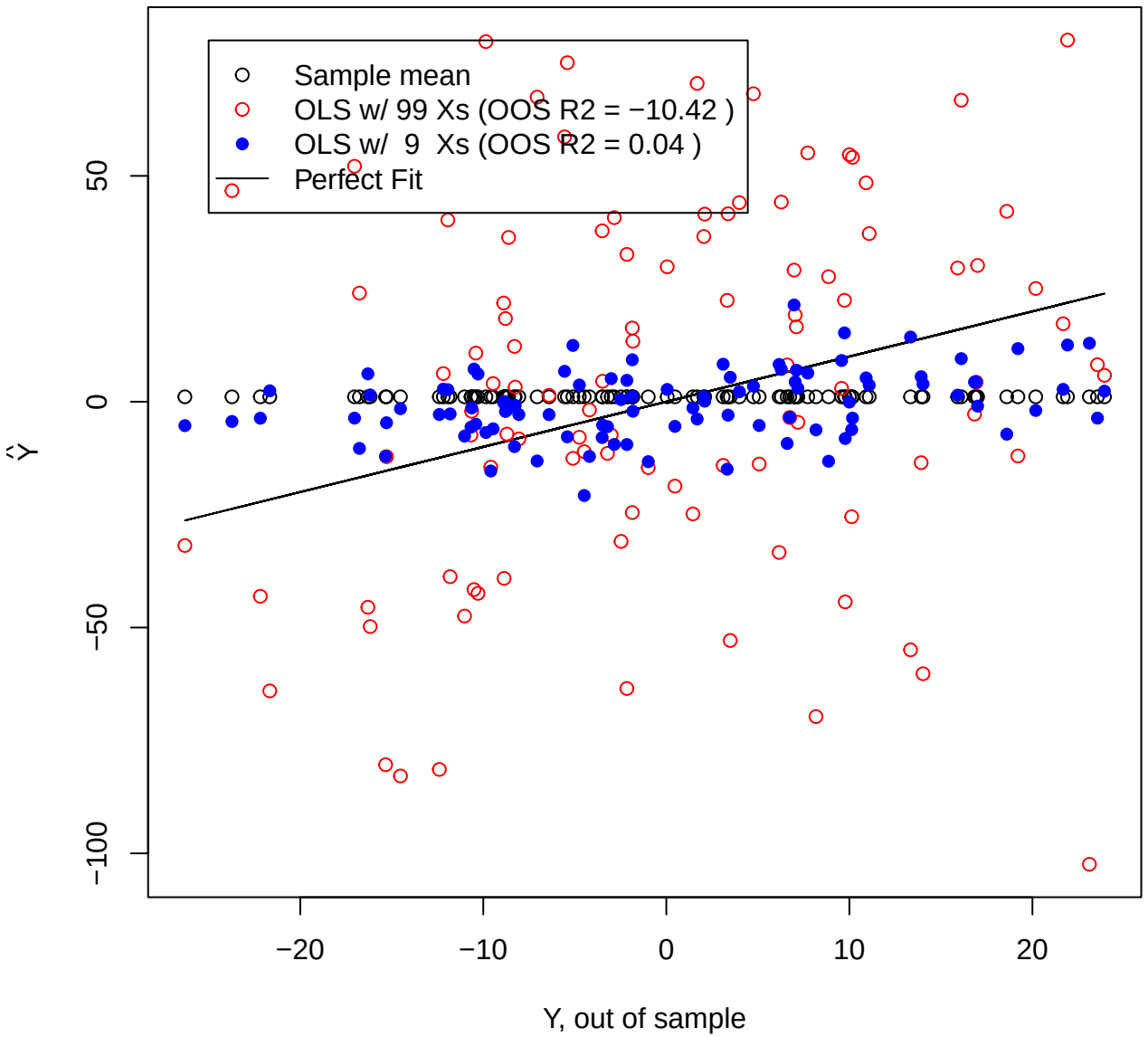
Results: Choosing λ



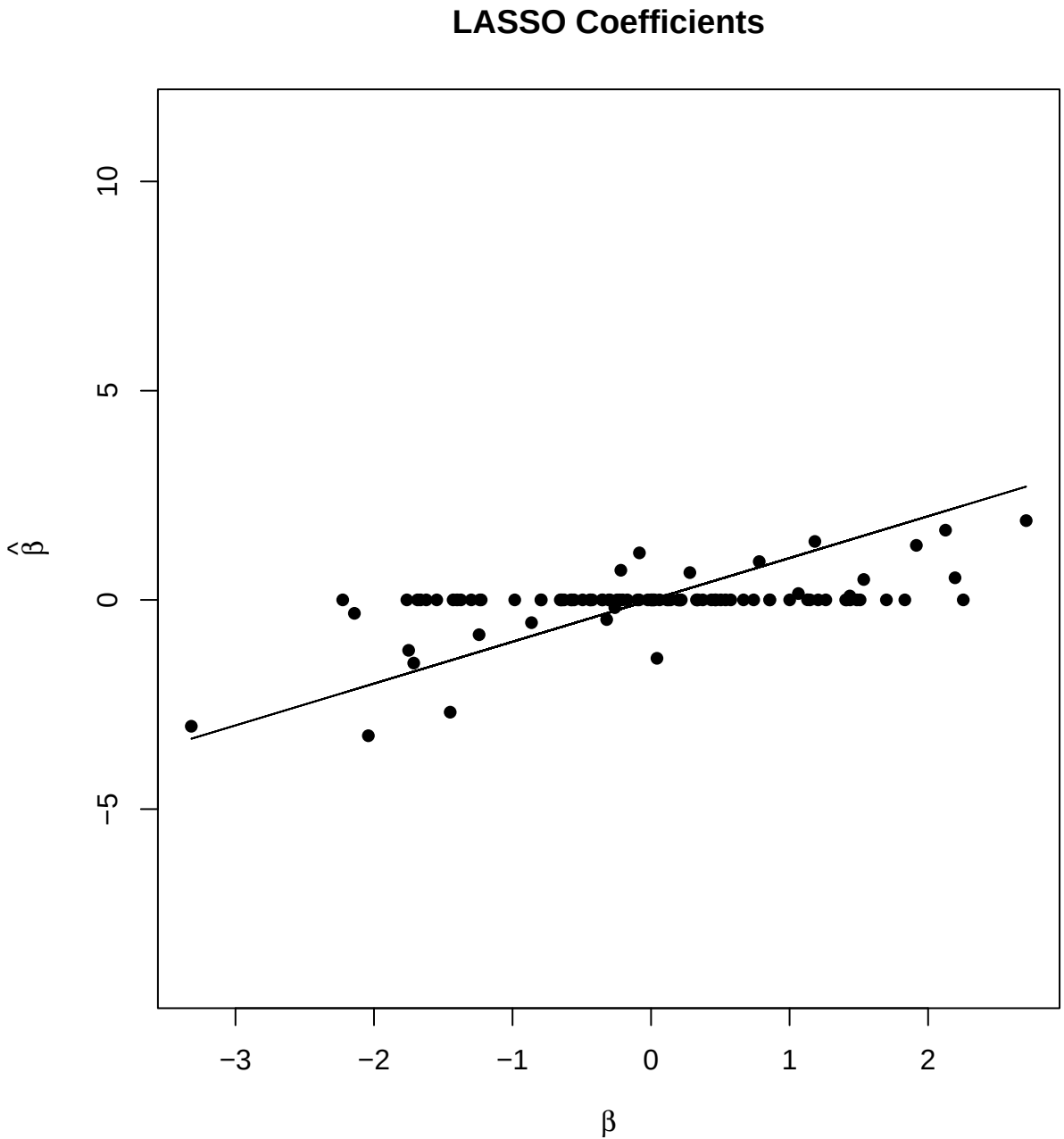
Results: LASSO predictions



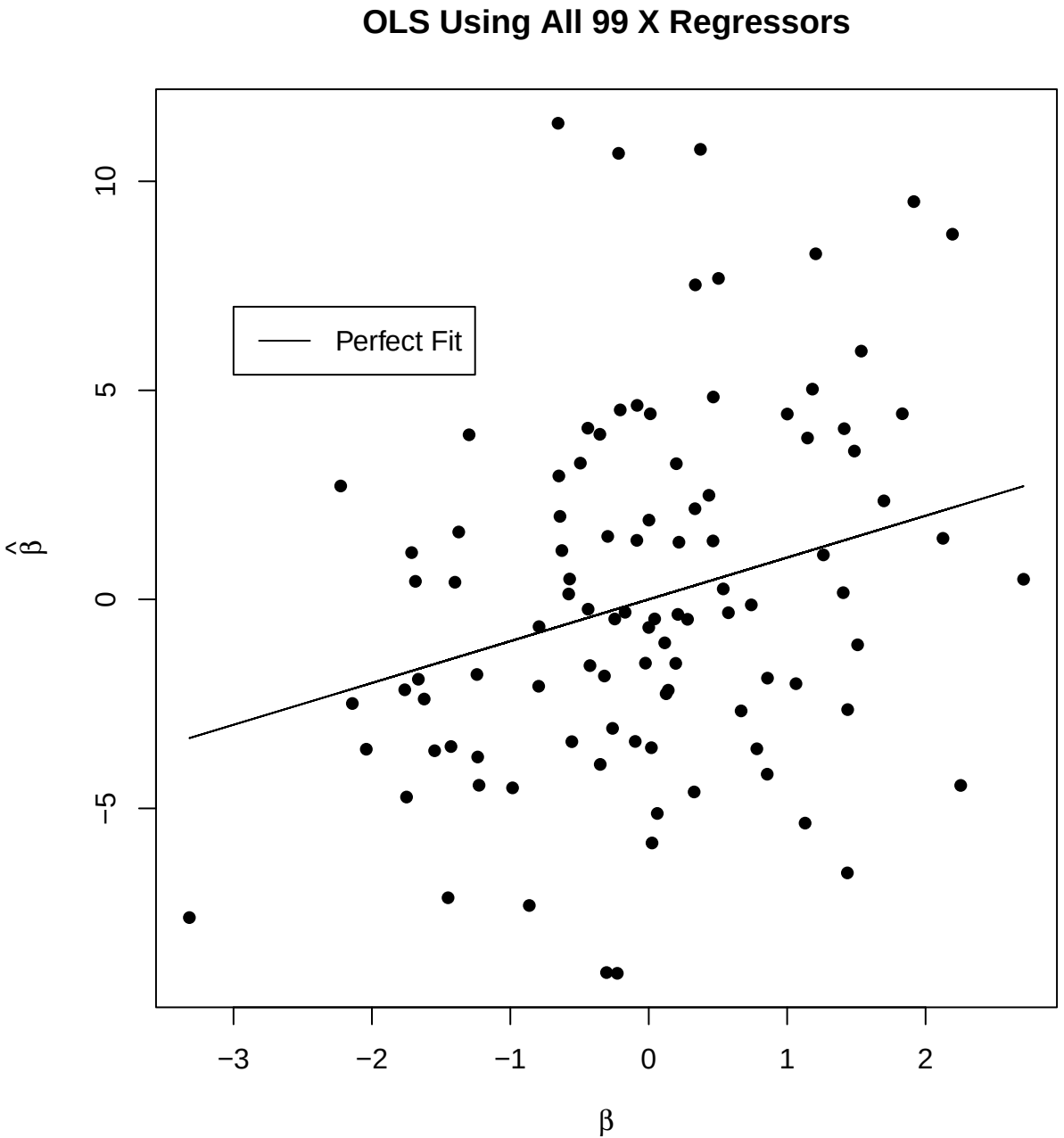
Results: LASSO predictions



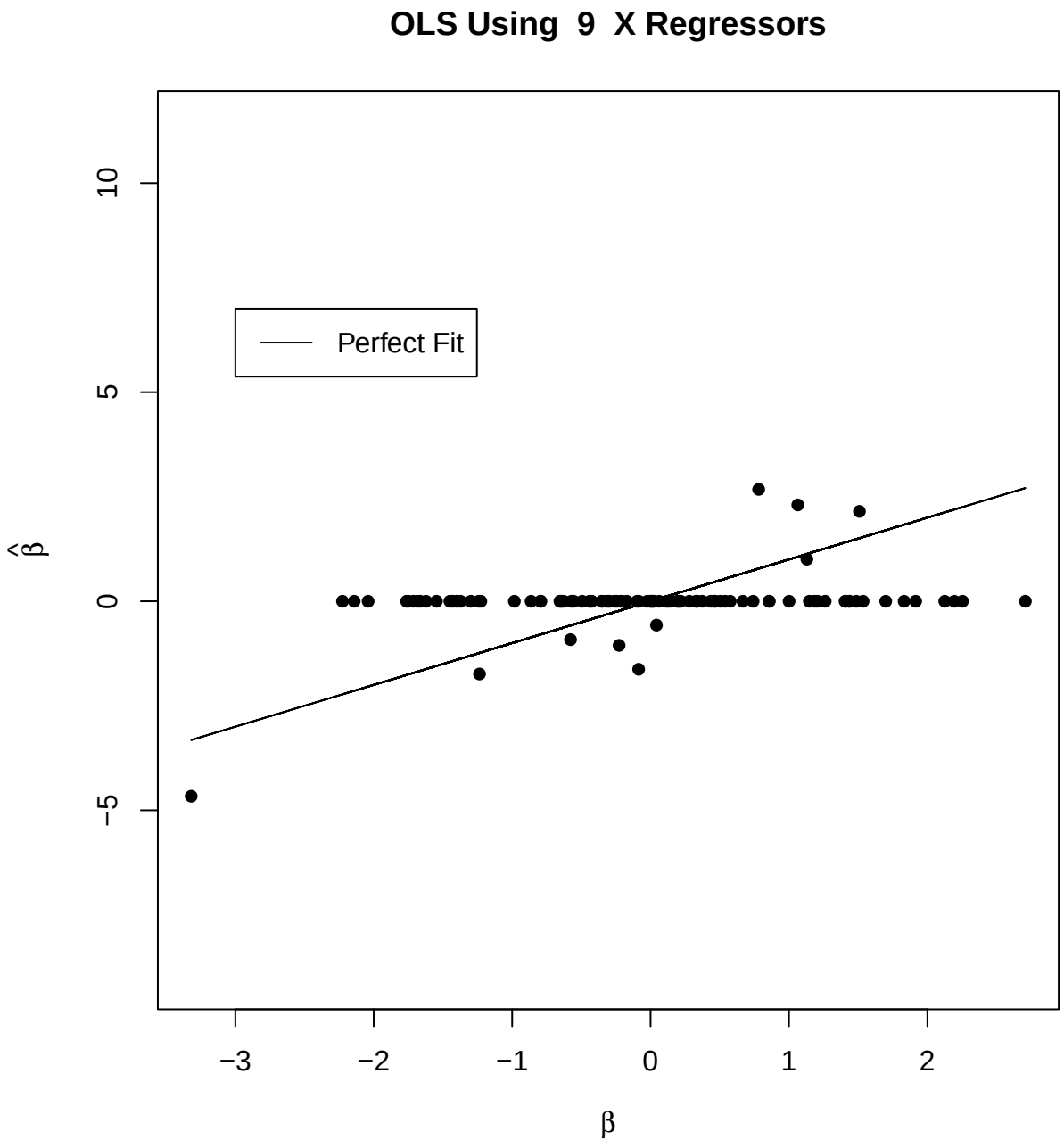
Results: LASSO coefficients



Results: LASSO coefficients



Results: LASSO coefficients



m-fold Cross-Validation

- To find λ, I performed "cross-validation:"
  1. Split the training data into 2 parts
  2. Estimated LASSO for different λs on part 1
  3. Evaluated them on the part 2
- Could have then done the reverse, chosen λ that does best on average
- Or...
  1. Split the training data into 3 parts
  2. Estimated LASSO for different λs on (combination of) parts 1 and 2
  3. Evaluated on part 3
  4. Repeat 2 more times (for parts 1 and 2)
  5. Choose λ performs best on average
     - This would be "3-fold cross-validation"

"Leave-one-out Estimator"

- Taken to the limit, I could do the following:
  1. Estimate LASSO for different λs using all observations 2-100
  2. Use those to predict Y_1
  3. Repeat the process for every observation 1-100
  4. Choose the λ that does best average
     - Known as "leave-one-out": every observation in the training data is its own fold
- When data is small enough, this is best practice
  - But it can be very computation- and time-intensive, so in large datasets may need to be less ambitious

Summary

- When $n >> K$, OLS is fine
  - Model that maximizes in-sample fit does similarly out-of-sample
- When K gets close to n, OLS will do terribly
  - Overfitting
- When $n \leq K$, cannot even estimate OLS
- So unless $n >> K$, OLS can be improved by dropping variables
  - Bias-Variance Tradeoff!
- Shrinkage estimators are a less-arbitrary way of doing this
  - Still arbitrary...
- Cross-validation is key
  - Test the estimator on data that was not used to estimate it