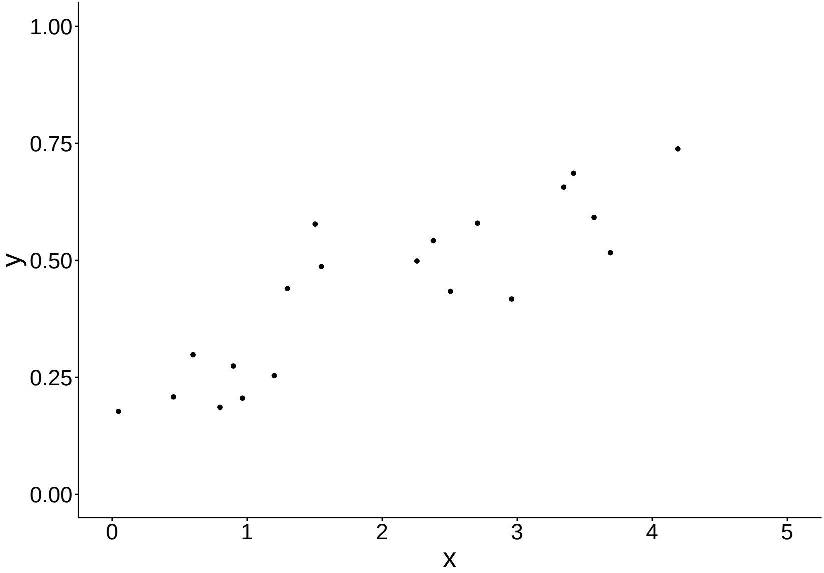
Mean Squared Error: In and Out of Sample

Review of key concepts

Use data to learn a model. What does that mean?

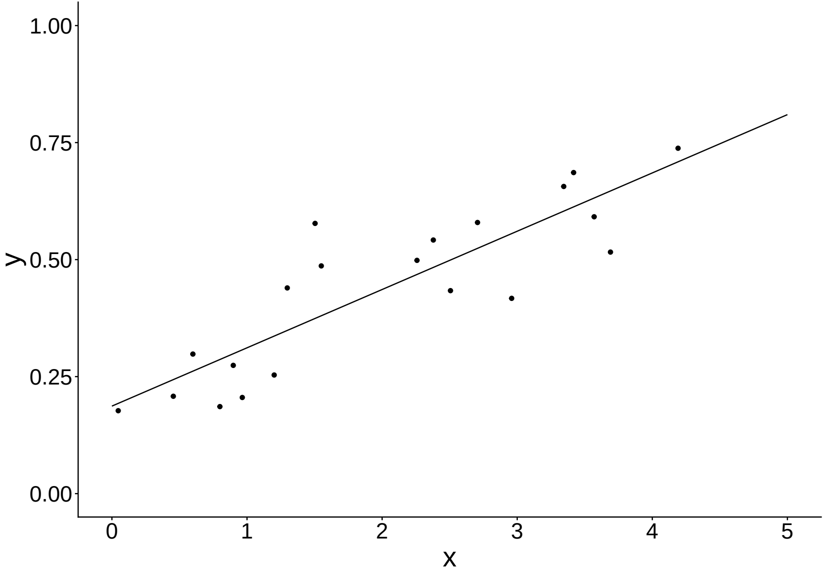
Begin with some data. Assume a linear model.



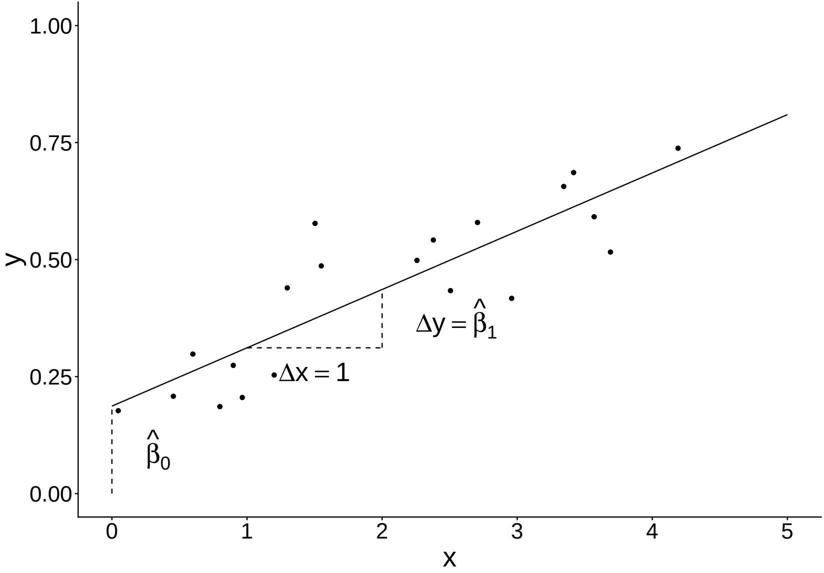
Estimate a linear model

```
Call:
lm(formula = y ~ x, data = simulated)

Residuals:
      Min        1Q    Median        3Q       Max
-0.138289 -0.069877  0.006674  0.056056  0.203069

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.18696    0.03969    4.71 0.000175 ***
x            0.12455    0.01688    7.38 7.58e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

Residual standard error: 0.09133 on 18 degrees of freedom
Multiple R-squared:  0.7516,    Adjusted R-squared:  0.7378
F-statistic: 54.47 on 1 and 18 DF,  p-value: 7.578e-07
```

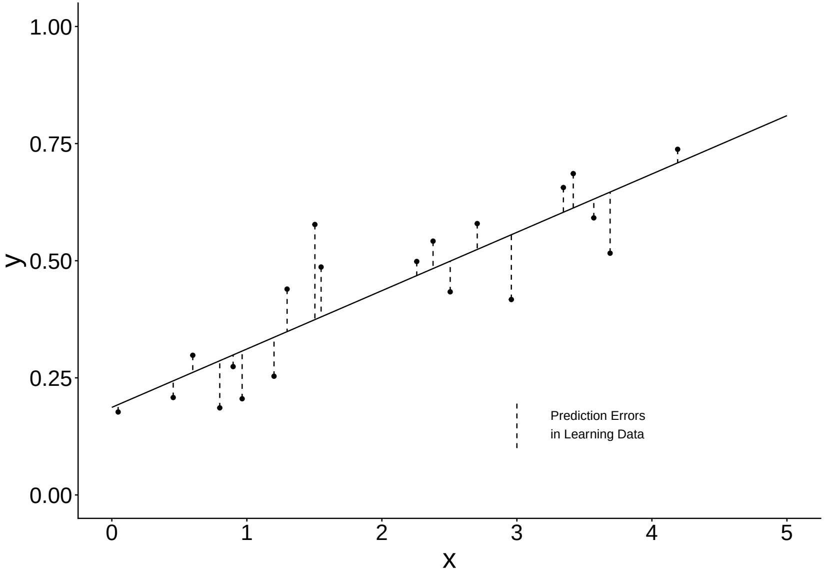
Estimate a linear model



The model **learned** coefficient values from the data.



How? It learned by minimizing the **sum of squared errors**



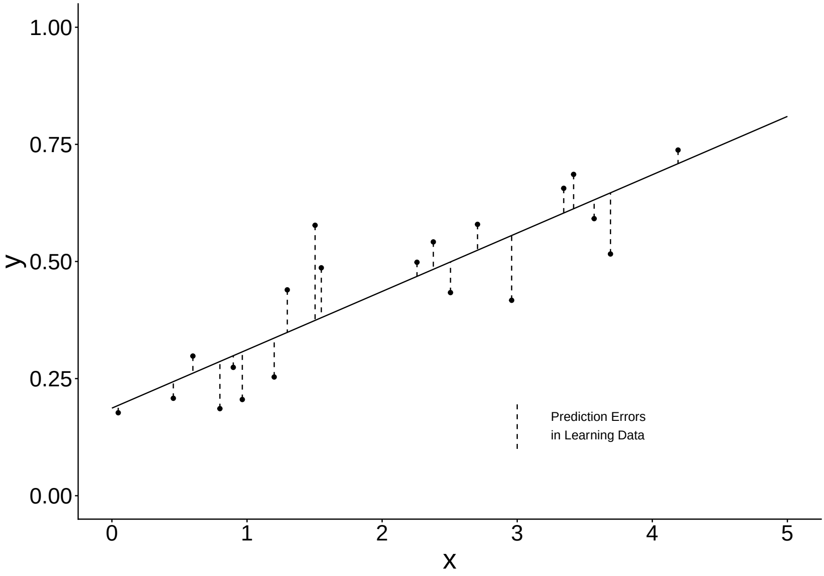
How? It learned by minimizing the **sum of squared errors**

Sum of squared error: $$\sum_{i \in \text{learning}} (y_i - \hat{y}_i)^2$$

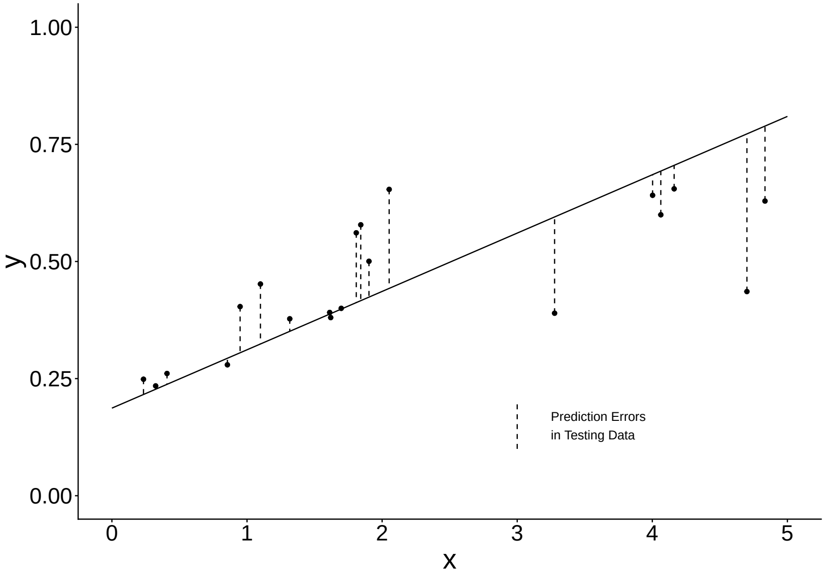
Mean squared error: $$\frac{1}{n_{\text{Learning}}} \sum_{i \in \text{learning}} (y_i - \hat{y}_i)$$

- We estimated the line in `learning` data
- Now we evaluate the estimated line in `testing` data

Learning data (used to estimate the line)



Testing data (used to evaluate the already-learned line)



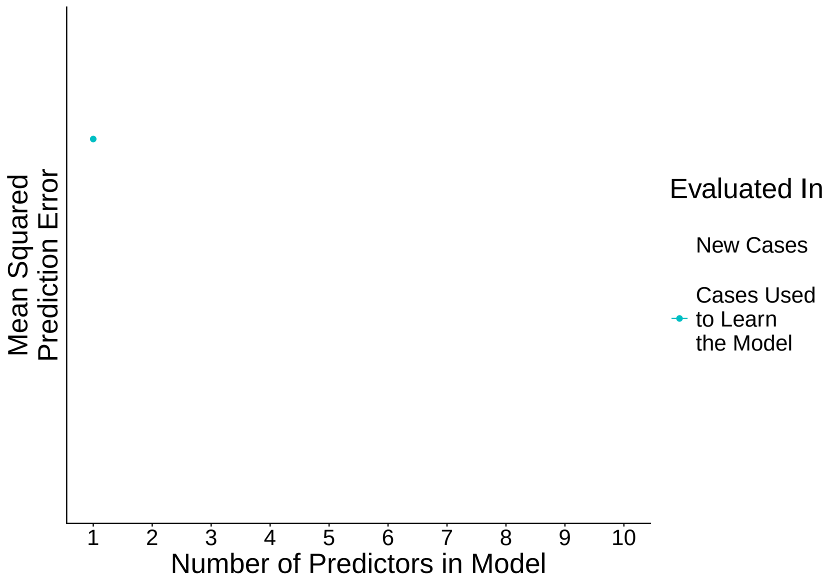
When will prediction errors in learning and testing data differ?

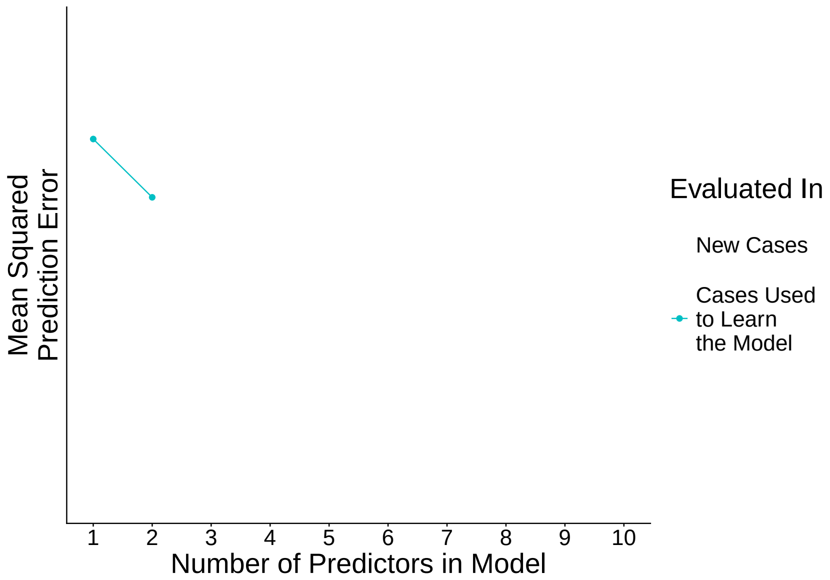
True model:

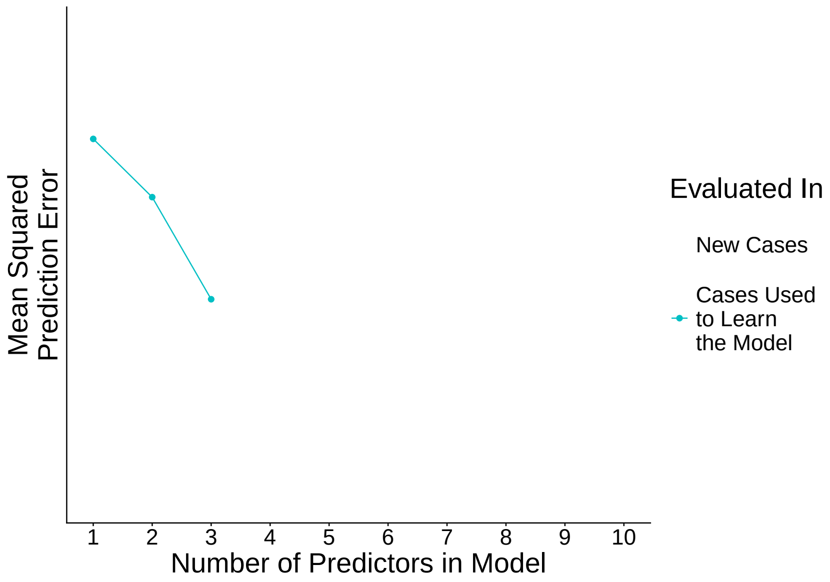
$$\mathsf{E}\left(Y \mid \vec{X}\right) = X_1\beta_1 + X_2\beta_2 + ... + X_{10}\beta_{10}$$

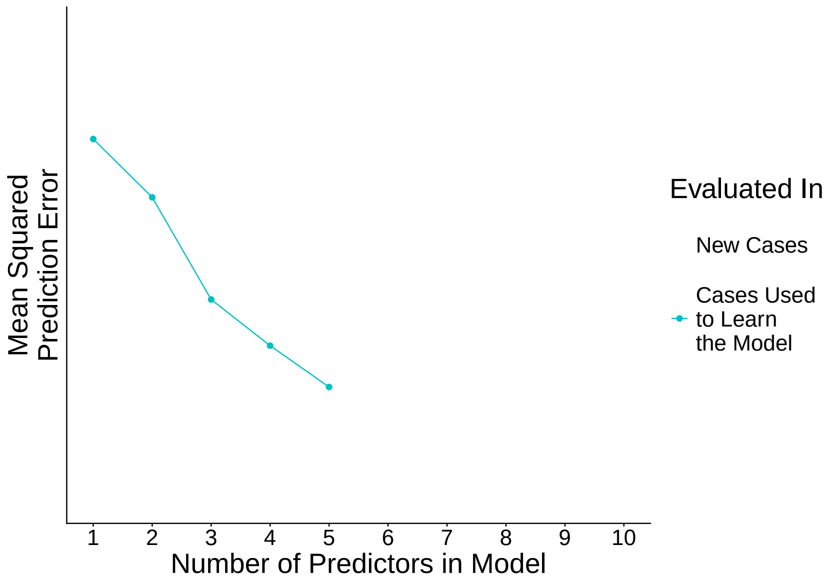
with $\beta_1 = .9$, $\beta_2 = 0.8$, ..., $\beta_9 = 0.1$, $\beta_{10} = 0$.

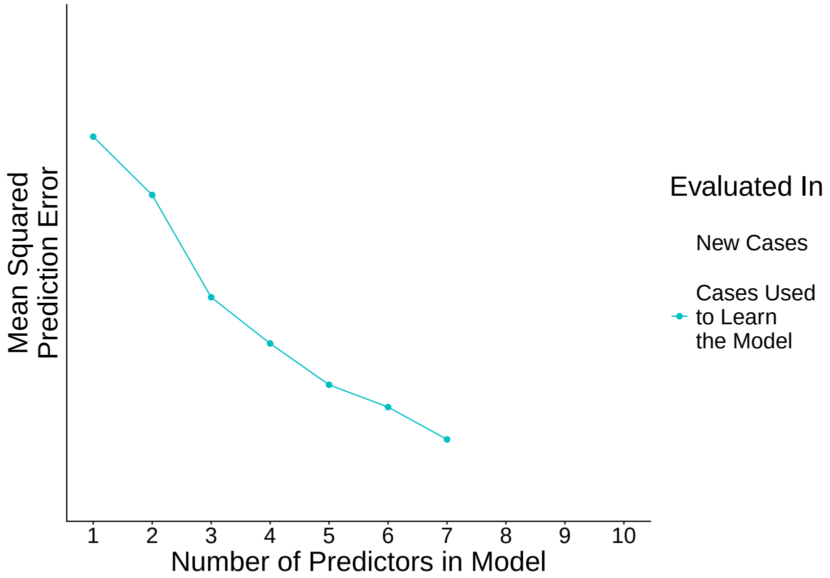
We observe $n = 300$ cases.


Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

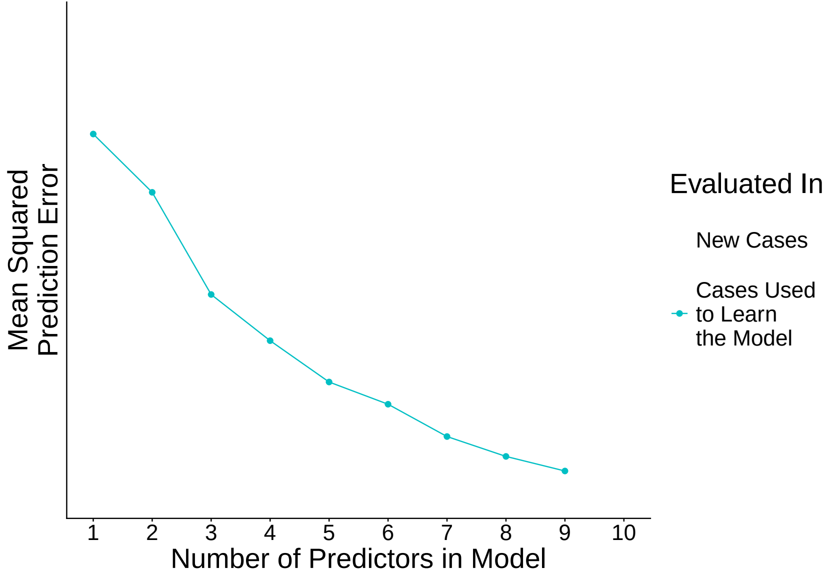

Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model


Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

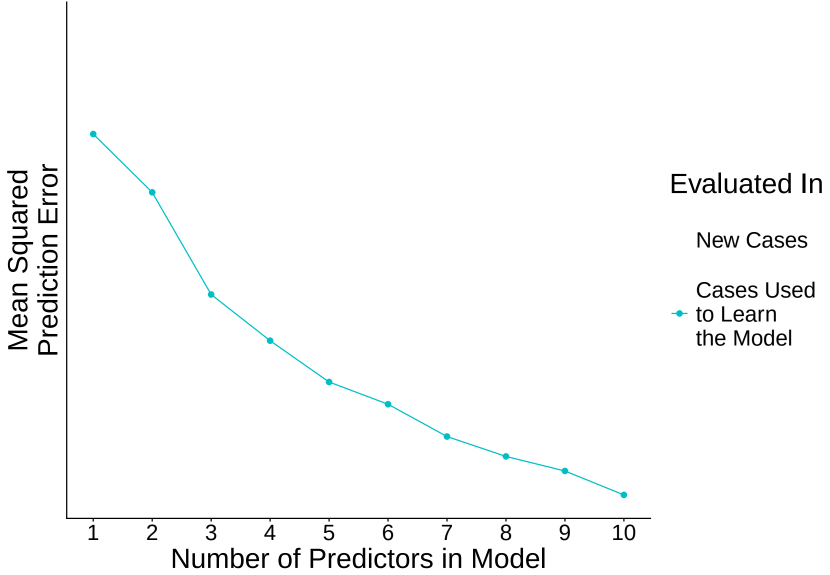

Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

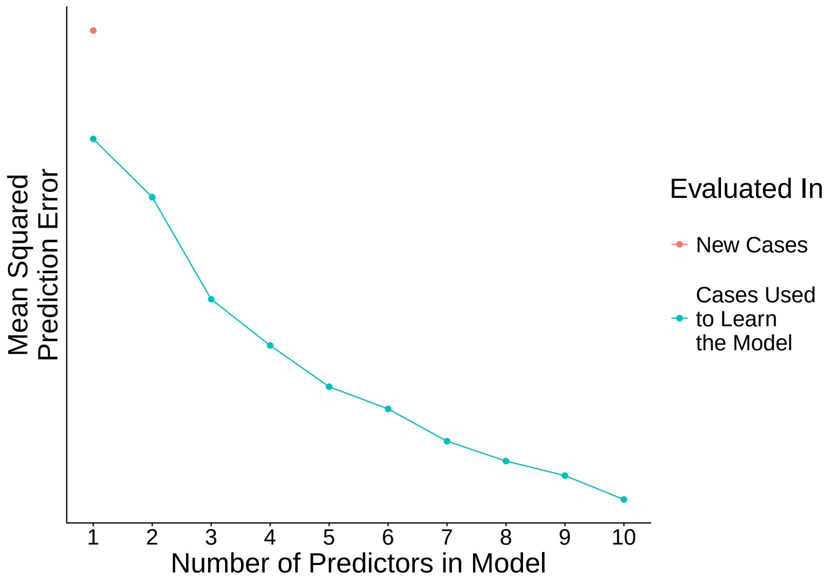


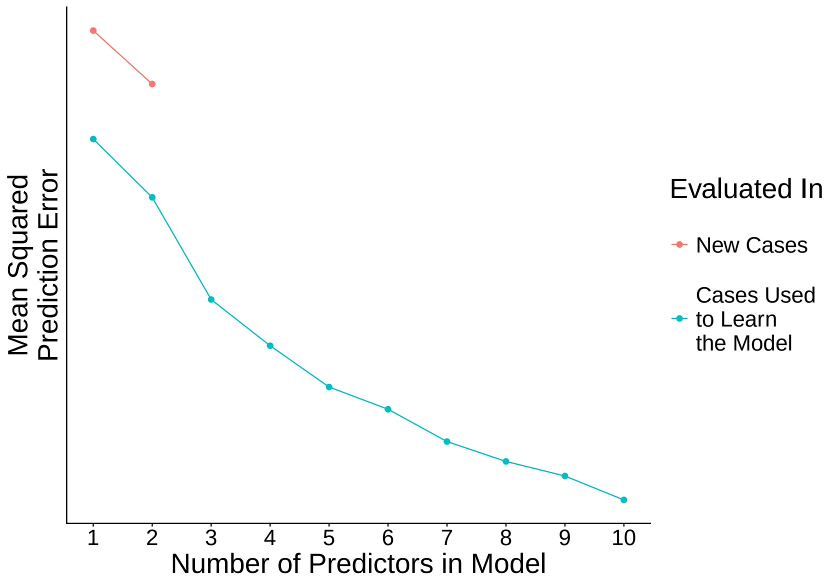
Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

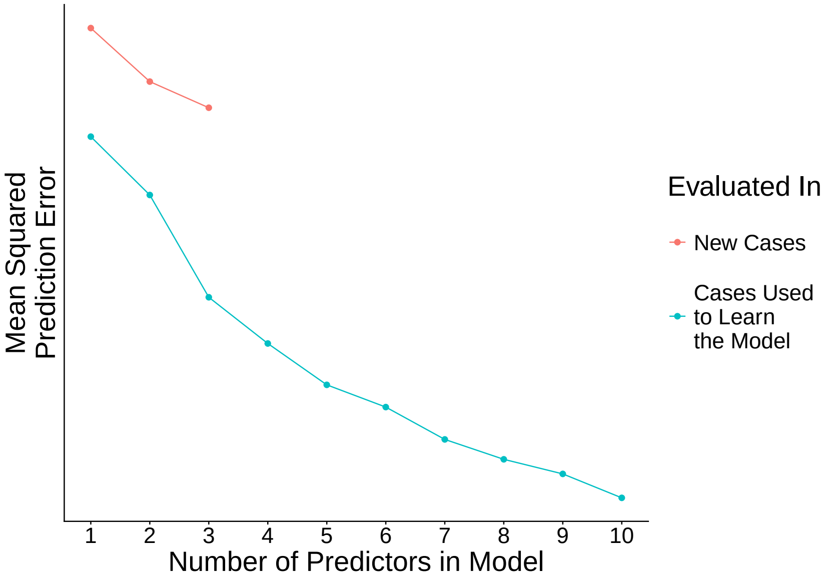


Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

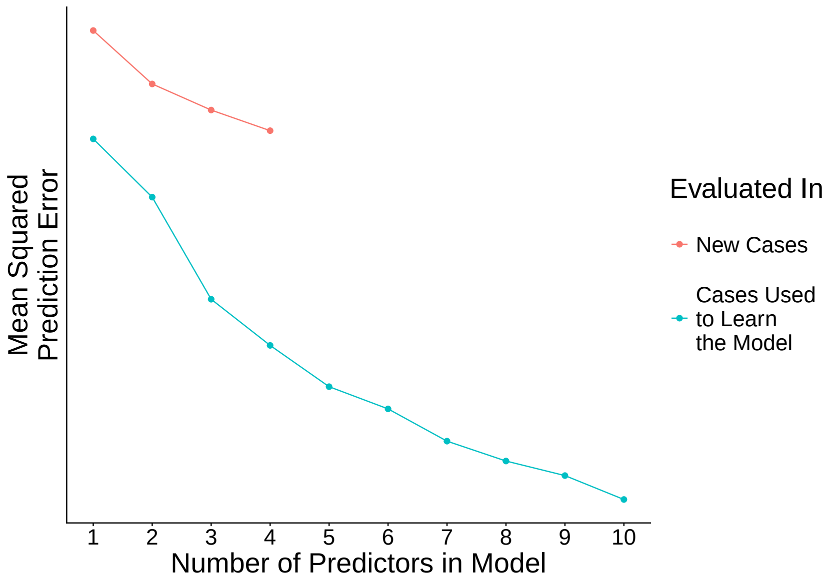

Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

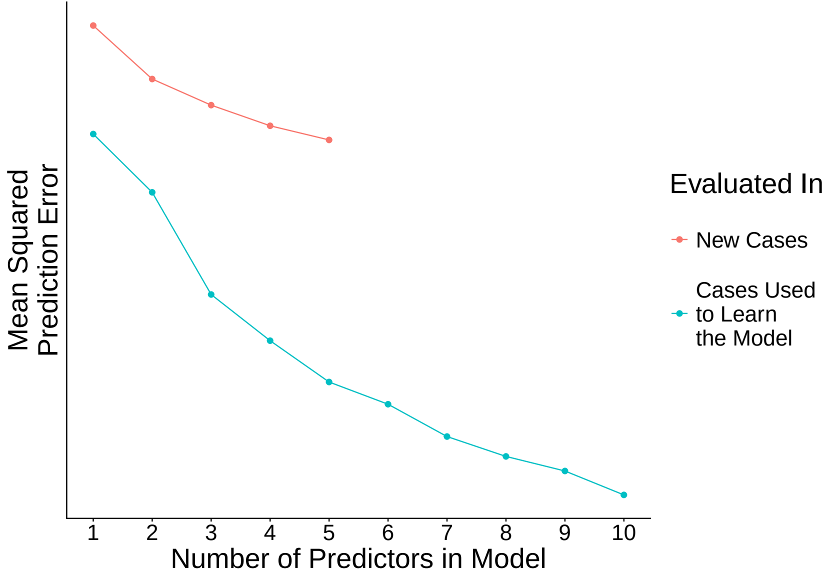

Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model


Mean Squared Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used to Learn the Model

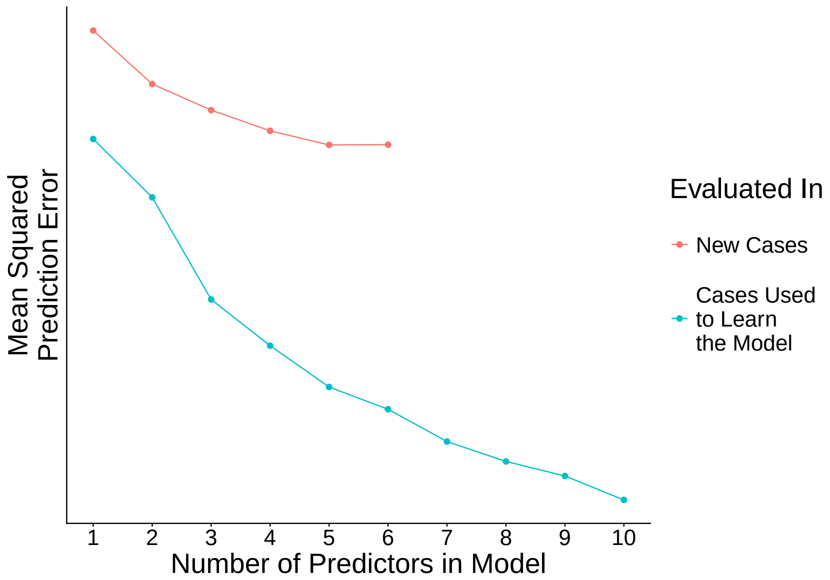


Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

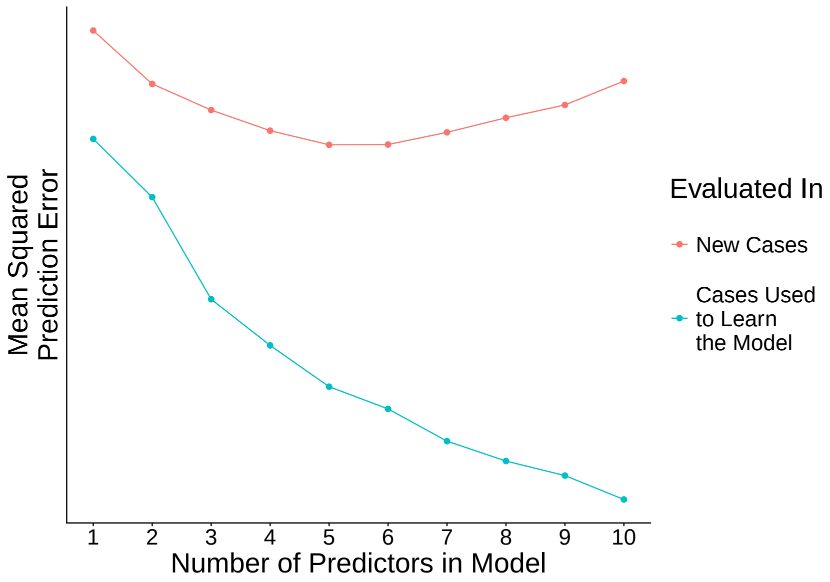

Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model



Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model



Mean Squared
Prediction Error
Number of Predictors in Model
1
2
3
4
5
6
7
8
9
10
Evaluated In
New Cases
Cases Used
to Learn
the Model



Mean Squared
Prediction Error
1
2
3
4
5
6
7
8
9
10
Number of Predictors in Model
Evaluated In
New Cases
Cases Used
to Learn
the Model

Recap: Surprising facts

- An estimated model picks up both
  - **Signal.** True patterns linking $\vec{X}$ to Y
  - **Noise.** Random patterns particular to the learning data.

Recap: Surprising facts

- As you add predictors to the model
  - error in learning data goes down
  - error in testing data may go up

Why? With many predictors, the noise may dominate the signal.

A use case for train and test: **Tuning parameters**

A use case for train and test: **Tuning parameters**

```
for_sample_split <- read_csv("https://soc114.github.io/asse
```

A use case for train and test: **Tuning parameters**

```
for_sample_split <- read_csv("https://soc114.github.io/asse
```

- Predictors `x1` through `x100`
- Outcome `y`
- $n = 300$ cases

A use case for train and test: **Tuning parameters**

Recall penalized (ridge) regression. Chooses $\vec{\beta}$ to minimize

$$\underbrace{\sum_i \left(Y_i - \hat{Y}_i\right)^2}_{\text{Sum of Squared Error}} + \underbrace{\lambda \sum_j \beta_j^2}_{\text{Penalty Term}}$$

But how to choose λ?

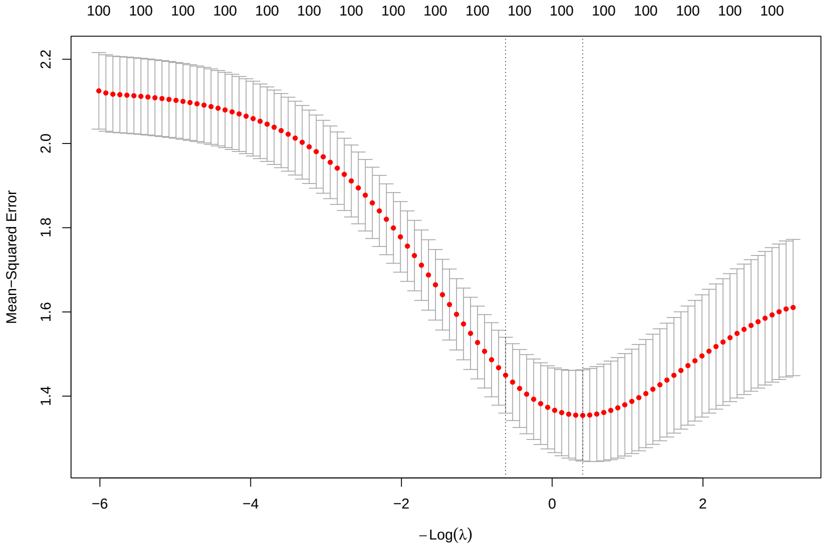
A use case for train and test: **Tuning parameters**

```
library(glmnet)
X <- model.matrix(y ~ ., data = for_sample_split)
y <- for_sample_split |> pull(y)
penalized_regression <- cv.glmnet(x = X, y = y, alpha = 0)
```

`cv.glmnet` chooses λ for you. How?

A use case for train and test: **Tuning parameters**

```
plot(penalized_regression)
```



It chooses λ to minimize out of sample prediction error

Recap: Data-driven model selection

When there are many candidate models, you can choose the one with the lowest out-of-sample mean squared error.