Model-Building, Prediction, & Cross-Validation

Fundamental Techniques in Data Science

Utrecht University

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

Model-Building

Model-Based Prediction

- Interval Estimates for Prediction

Building & Evaluating Predictive Models

- Over-fitting
- Training vs. Testing Errors

Cross Validation

Model-Building Example

Let's walk through an example of the model-building process.

- We'll take $Y_{bp} = \beta_0 + \beta_1 X_{age.30} + \varepsilon$ as our baseline model.
- Next, we simultaneously add predictors of LDL and HDL cholesterol.

```
diabetes <- readRDS("../data/diabetes.rds")

## Center predictor variables:
diabetes <- mutate(diabetes,
                   ldl100 = ldl - 100,
                   hdl60 = hdl - 60,
                   age30 = age - 30)

## Baseline model:
out1 <- lm(bp ~ age30, data = diabetes)

## Simultaneously add two predictors:
out2 <- lm(bp ~ age30 + ldl100 + hdl60, data = diabetes)
```

Model-Building Example

```
partSummary(out1, -1)

Residuals:
    Min      1Q  Median      3Q     Max
-31.188  -8.897  -1.209   8.612  39.952

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 88.09330    1.07470  81.970  < 2e-16
age30        0.35391    0.04739   7.469 4.39e-13

Residual standard error: 13.04 on 440 degrees of freedom
Multiple R-squared:  0.1125,Adjusted R-squared:  0.1105
F-statistic: 55.78 on 1 and 440 DF,  p-value: 4.393e-13
```

Model-Building Example

```
partSummary(out2, -1)

Residuals:
    Min      1Q  Median      3Q     Max
-33.297  -8.106  -0.979   8.141  40.677

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 86.53984    1.13885  75.989  < 2e-16
age30        0.32178    0.04784   6.727 5.43e-11
ldl100       0.04166    0.02097   1.987  0.04757
hdl60       -0.14740    0.04824  -3.055  0.00239

Residual standard error: 12.84 on 438 degrees of freedom
Multiple R-squared:  0.1439,Adjusted R-squared:  0.1381
F-statistic: 24.55 on 3 and 438 DF,  p-value: 1.064e-14
```

Interpretations

- The expected average blood pressure for a 30 year old patient with LDL = 100 and HDL = 60 is 86.54.
- For each additional year older, average blood pressure is expected to increase by 0.32, after controlling for LDL and HDL levels.
- For each additional unit of LDL level, average blood pressure is expected to increase by 0.04, after controlling for age and HDL.
- For each additional unit of HDL level, average blood pressure is expected to decrease by -0.15, after controlling for age and LDL.

Model Comparison

```
## Compute change in R^2:
summary(out2)$r.squared - summary(out1)$r.squared

[1] 0.03142445

## Significance test for change in R^2:
anova(out1, out2)

Analysis of Variance Table

Model 1: bp ~ age30
Model 2: bp ~ age30 + ldl100 + hdl60
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    440 74873
2    438 72222  2    2651.1 8.0391 0.0003726 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Comparison

```
(mse1 <- MSE(y_pred = predict(out1), y_true = diabetes$bp))

[1] 169.3963

(mse2 <- MSE(y_pred = predict(out2), y_true = diabetes$bp))

[1] 163.3983

AIC(out1, out2)

     df      AIC
out1  3 3528.792
out2  5 3516.858

BIC(out1, out2)

     df      BIC
out1  3 3541.066
out2  5 3537.314
```

Interpretations

- Age, LDL, and HDL explain a combined 14.4% of the variation in blood pressure.
  - This proportion of variation explained is significantly greater than zero.
- Adding LDL and HDL produces a model that explains 3.1% more variation in blood pressure than a model with age as the only predictor.
  - This increase in variation explained is significantly greater than zero.
- Adding LDL and HDL produces a model with lower prediction error (i.e., MSE = 163.4 vs. MSE = 169.4).
- Both the AIC and the BIC also suggest that adding LDL and HDL produces a better model.

Continue Building the Model

So far we've established that age, LDL, and HDL are all significant predictors of average blood pressure.

- We've also established that adding LDL and HDL, together, explain significantly more variation than age alone.

Next, we'll add BMI to see what additional explanatory role it can play above and beyond age and cholesterol.

```
## Center BMI:
diabetes <- mutate(diabetes, bmi25 = bmi - 25)

## Now, add bmi:
out3 <- lm(bp ~ age30 + ldl100 + hdl60 + bmi25, data = diabetes)
```

Model-Building Example

```
partSummary(out3, -1)

Residuals:
    Min      1Q  Median      3Q     Max
-29.970  -8.145  -0.300   8.456  41.135

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 87.46233    1.08944  80.282  < 2e-16
age30        0.27949    0.04582   6.099 2.35e-09
ldl100       0.01646    0.02024   0.814    0.416
hdl60       -0.03478    0.04856  -0.716    0.474
bmi25        1.01743    0.14568   6.984 1.07e-11

Residual standard error: 12.19 on 437 degrees of freedom
Multiple R-squared:  0.2299,Adjusted R-squared:  0.2228
F-statistic: 32.61 on 4 and 437 DF,  p-value: < 2.2e-16
```

Interpretations

BMI seems to have a pretty strong effect on average blood pressure, after controlling for age and cholesterol levels.

- After controlling for BMI, cholesterol levels no longer seem to be important predictors.
- Let's take a look at what happens to the cholesterol effects when we add BMI:

	LDL	HDL
Without BMI	0.042	-0.147
With BMI	0.016	-0.035

Model Comparison

How much additional variability in blood pressure is explained by BMI above and beyond age and cholesterol levels?

```
r2.3 <- summary(out3)$r.squared
r2.3 - r2.2

[1] 0.08595543
```

Model Comparison

Is the additional 8.6% variation explained a significant increase?

```
anova(out2, out3)

Analysis of Variance Table

Model 1: bp ~ age30 + ldl100 + hdl60
Model 2: bp ~ age30 + ldl100 + hdl60 + bmi25
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    438 72222
2    437 64970  1    7251.7 48.776 1.074e-11 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Comparison

```
mse3 <- MSE(y_pred = predict(out3), y_true = diabetes$bp)

mse2

[1] 163.3983

mse3

[1] 146.9918

AIC(out2, out3)

     df      AIC
out2  5 3516.858
out3  6 3472.088

BIC(out2, out3)

     df      BIC
out2  5 3537.314
out3  6 3496.636
```

Model Modification

Maybe cholesterol levels are not important features once we've accounted for BMI.

- Let's try a model including BMI but excluding cholesterol levels.

```
## Take out the cholesterol variables:
out4 <- lm(bp ~ age30 + bmi25, data = diabetes)
```

Model-Building Example

```
partSummary(out4, -1)

Residuals:
    Min      1Q  Median      3Q     Max
-29.287  -8.198  -0.178   8.413  41.026

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 87.85488    1.00420  87.487  < 2e-16
age30        0.28651    0.04504   6.362 5.02e-10
bmi25        1.08053    0.13363   8.086 6.06e-15

Residual standard error: 12.18 on 439 degrees of freedom
Multiple R-squared:  0.2276,Adjusted R-squared:  0.224
F-statistic: 64.66 on 2 and 439 DF,  p-value: < 2.2e-16
```

Model Comparison

How much explained variation did we loose by removing the LDL and HDL variables?

```
r2.4 <- summary(out4)$r.squared
r2.3 - r2.4

[1] 0.002330906
```

Model Comparison

Is this 0.23% loss in explained variance significant?

```
anova(out4, out3)

Analysis of Variance Table

Model 1: bp ~ age30 + bmi25
Model 2: bp ~ age30 + ldl100 + hdl60 + bmi25
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1    439 65167
2    437 64970  2    196.65 0.6613 0.5167
```

Model Comparison

```
mse4 <- MSE(y_pred = predict(out4), y_true = diabetes$bp)

mse3

[1] 146.9918

mse4

[1] 147.4367

AIC(out3, out4)

     df      AIC
out3  6 3472.088
out4  4 3469.424

BIC(out3, out4)

     df      BIC
out3  6 3496.636
out4  4 3485.789
```

Model-Based Prediction

Prediction

So far, we've focused mostly on inferences about the estimated regression coefficients.

- Asking questions about how X is related to Y.

We can also use linear regression for *prediction*.

- Given a new observation, X_m, what outcome value, $\hat{Y}_m$, does our model attribute to the *m*th observation?

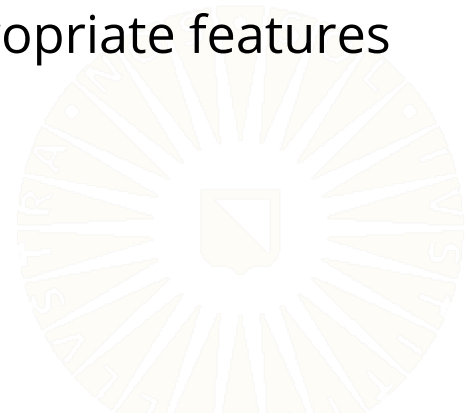
Prediction

Train a model to predict psychological well-being from diet-related and exercise-related features.

- Plug-in new feature values corresponding to an experimental wellness program to see the expected well-being for a hypothetical patient treated with the new program.

Predict future gasoline prices based on geo-political events in oil-producing countries.

- If conflict escalates in the Middle East, adjust the appropriate features and project likely changes in gasoline prices.



Prediction Example

To fix ideas, let's reconsider the *diabetes* data and the following model:

$$Y_{LDL} = \beta_0 + \beta_1 X_{BP} + \beta_2 X_{gluc} + \beta_3 X_{BMI} + \varepsilon$$

Training this model on the first $N = 400$ patients' data produces the following fitted model:

$$\hat{Y}_{LDL} = 22.135 + 0.089 X_{BP} + 0.498 X_{gluc} + 1.48 X_{BMI}$$

Prediction Example

To fix ideas, let's reconsider the *diabetes* data and the following model:

$$Y_{LDL} = \beta_0 + \beta_1 X_{BP} + \beta_2 X_{gluc} + \beta_3 X_{BMI} + \varepsilon$$

Training this model on the first $N = 400$ patients' data produces the following fitted model:

$$\hat{Y}_{LDL} = 22.135 + 0.089 X_{BP} + 0.498 X_{gluc} + 1.48 X_{BMI}$$

Suppose a new patient presents with $BP = 121$, $gluc = 89$, and $BMI = 30.6$. We can predict their LDL score by:

$$\begin{aligned} \hat{Y}_{LDL} &= 22.135 + 0.089(121) + 0.498(89) + 1.48(30.6) \\ &= 122.463 \end{aligned}$$

Interval Estimates for Prediction

To quantify uncertainty in our predictions, we want to use an appropriate interval estimate.

- Two flavors of interval are applicable to predictions:
  1. Confidence intervals for $\hat{Y}_m$
  2. Prediction intervals for a specific observation, Y_m

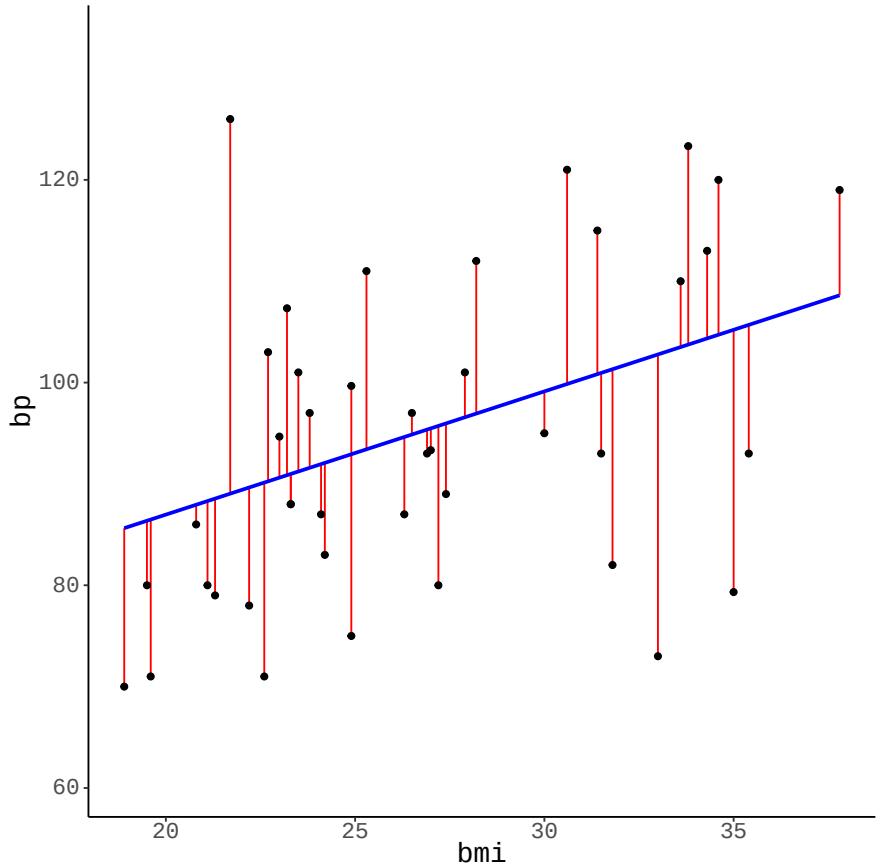
- The CI for $\hat{Y}_m$ gives a likely range (in the sense of coverage probability and "confidence") for the *m*th value of the true conditional mean.
  - CIs only account for uncertainty in the estimated regression coefficients, $\{\hat{\beta}_0, \hat{\beta}_p\}$.

- The prediction interval for Y_m gives a likely range (in the same sense as CIs) for the *m*th outcome value.
  - Prediction intervals also account for the regression errors, ε.

Confidence vs. Prediction Intervals

Let's visualize the predictions from a simple model:

$$Y_{BP} = \hat{\beta}_0 + \hat{\beta}_1 X_{BMI} + \hat{\varepsilon}$$

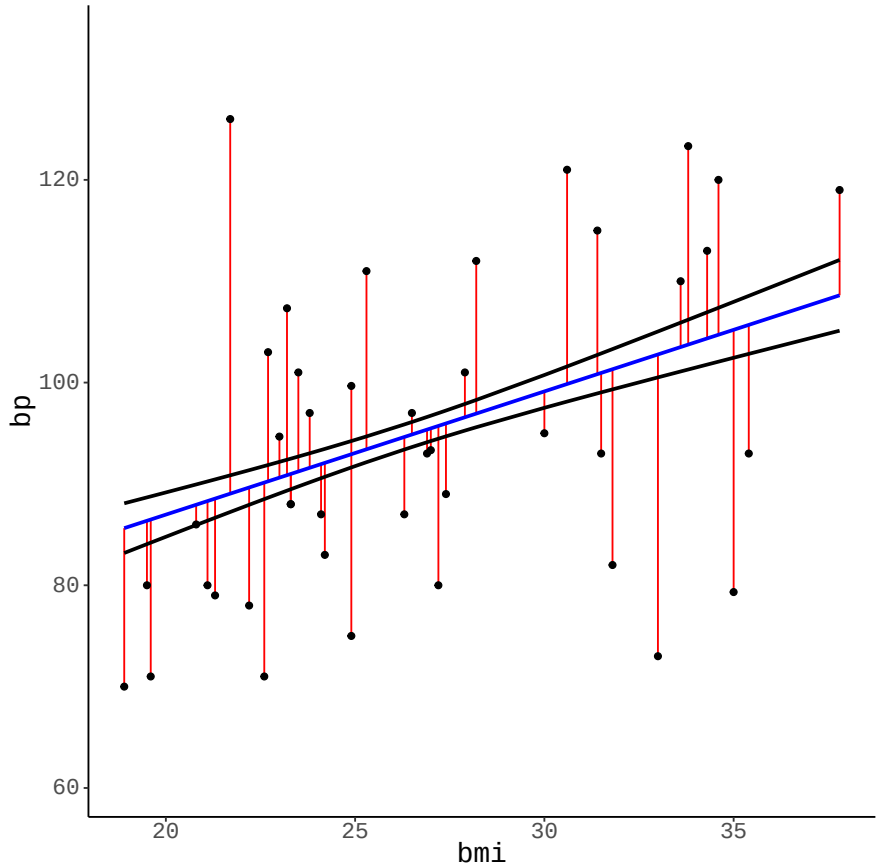


Confidence vs. Prediction Intervals

Let's visualize the predictions from a simple model:

$$Y_{BP} = \hat{\beta}_0 + \hat{\beta}_1 X_{BMI} + \hat{\varepsilon}$$

- CIs for $\hat{Y}$ ignore the errors, ε.
  - They only care about the best-fit line, $\beta_0 + \beta_1 X_{BMI}$.

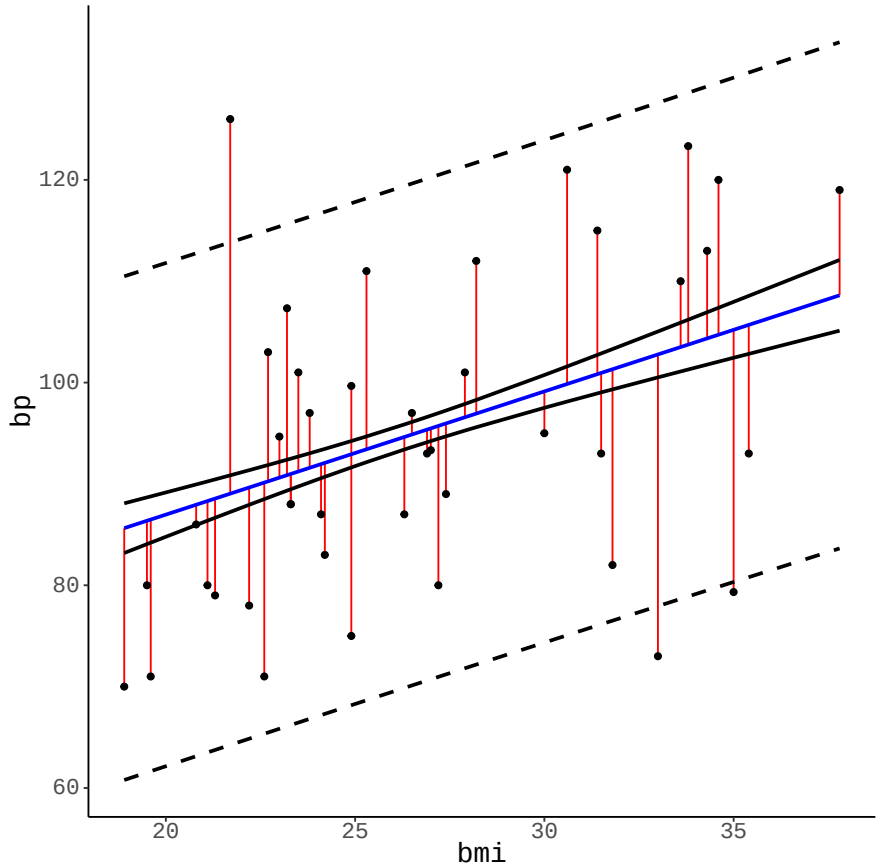


Confidence vs. Prediction Intervals

Let's visualize the predictions from a simple model:

$$Y_{BP} = \hat{\beta}_0 + \hat{\beta}_1 X_{BMI} + \hat{\varepsilon}$$

- CIs for $\hat{Y}$ ignore the errors, ε.
  - They only care about the best-fit line, $\beta_0 + \beta_1 X_{BMI}$.
- Prediction intervals are wider than CIs.
  - They account for the additional uncertainty contributed by ε.



Interval Estimates Example

Going back to our hypothetical "new" patient, we get the following 95% interval estimates:

$$95\%\ CI_{\hat{Y}} = [115.6; 129.33]$$

$$95\%\ PI = [66.56; 178.37]$$

- We can be 95% confident that the average *LDL* of patients with *Glucose* = 89, *BP* = 121, and *BMI* = 30.6 will be somewhere between 115.6 and 129.33.
- We can be 95% confident that the *LDL* of a specific patient with *Glucose* = 89, *BP* = 121, and *BMI* = 30.6 will be somewhere between 66.56 and 178.37.

Building & Evaluating Predictive Models

Specifying Predictive Models

When focused on inferences about regression coefficients, we care very much about the predictors entered into the model.

- Partial regression coefficients must be interpreted as controlling for all other predictors.

Specifying Predictive Models

When focused on inferences about regression coefficients, we care very much about the predictors entered into the model.

- Partial regression coefficients must be interpreted as controlling for all other predictors.

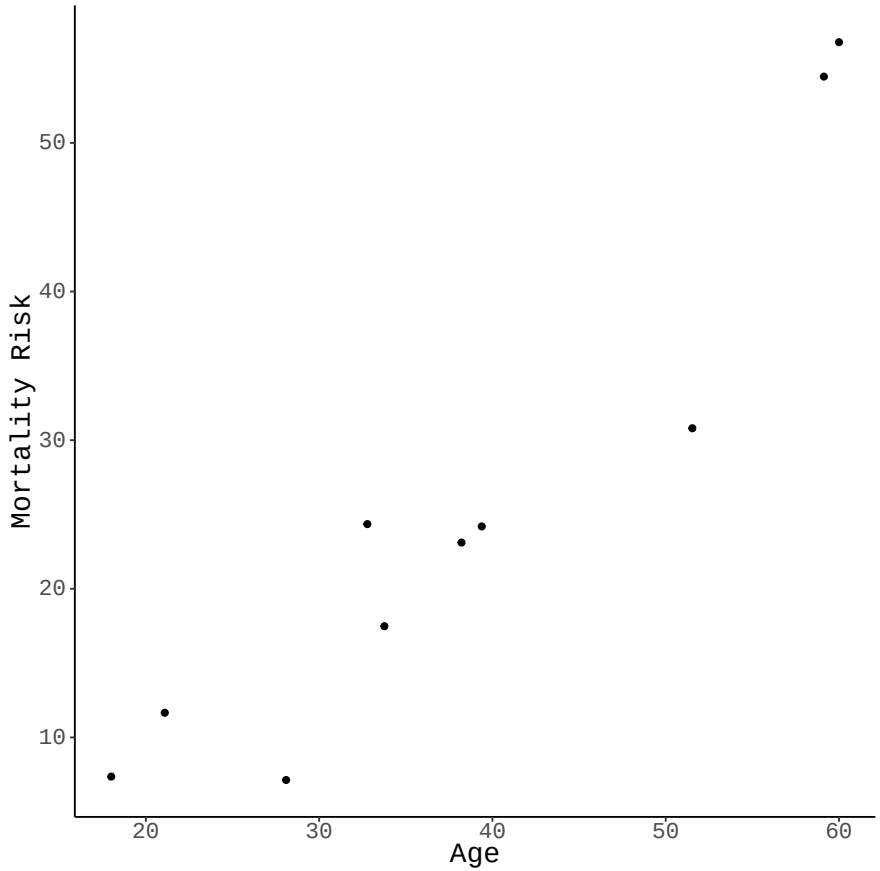
When focused on prediction, we often don't care as much about the specific variables that enter the model.

- We prefer whatever set of features produces the best predictive performance.
- We may want to know which are the "best" predictors.
  - We usually want the data to "give" us this answer.

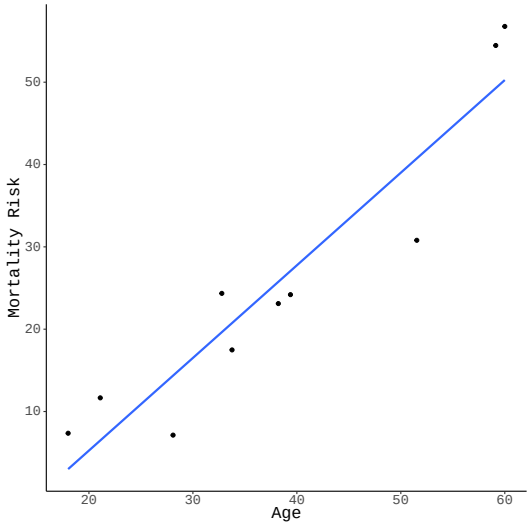
Evaluating Predictive Performance

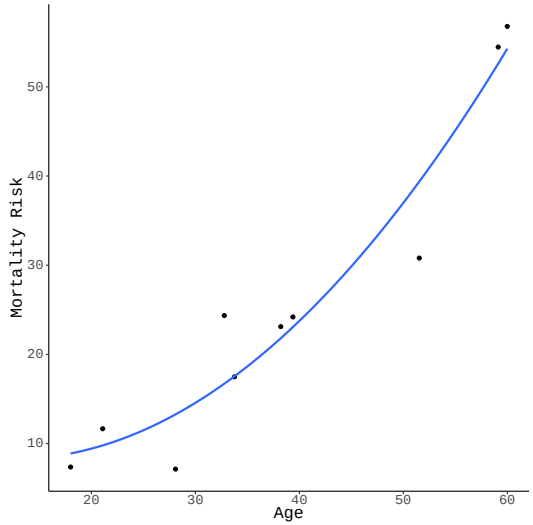
How do we assess “good” prediction?

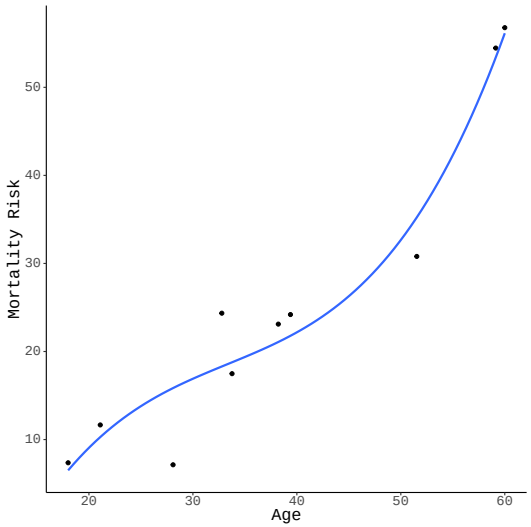
- Can we simply find the model that best predicts the data used to train the model?
- What are we trying to do when building a predictive model?
- Can we quantify this objective with some type of fit measure?

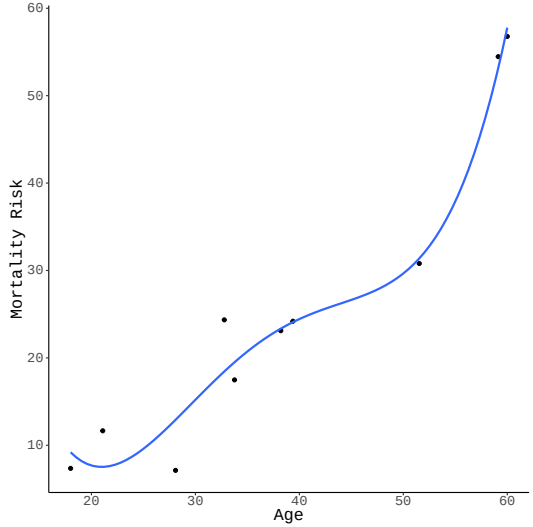


Different Possible Models





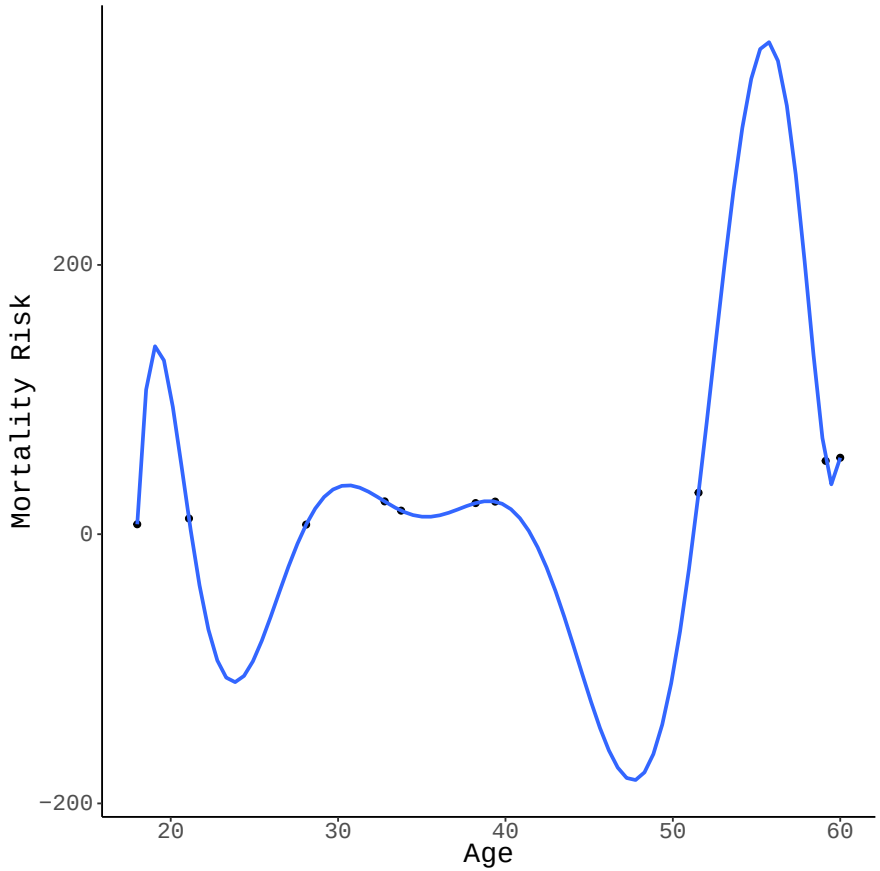




Over-fitting

We can easily go too far.

- Enough polynomial terms will exactly replicate any data.
- Is this what we're trying to do?
- What kind of issues arise in the extreme case?



Consequences of Over-fitting

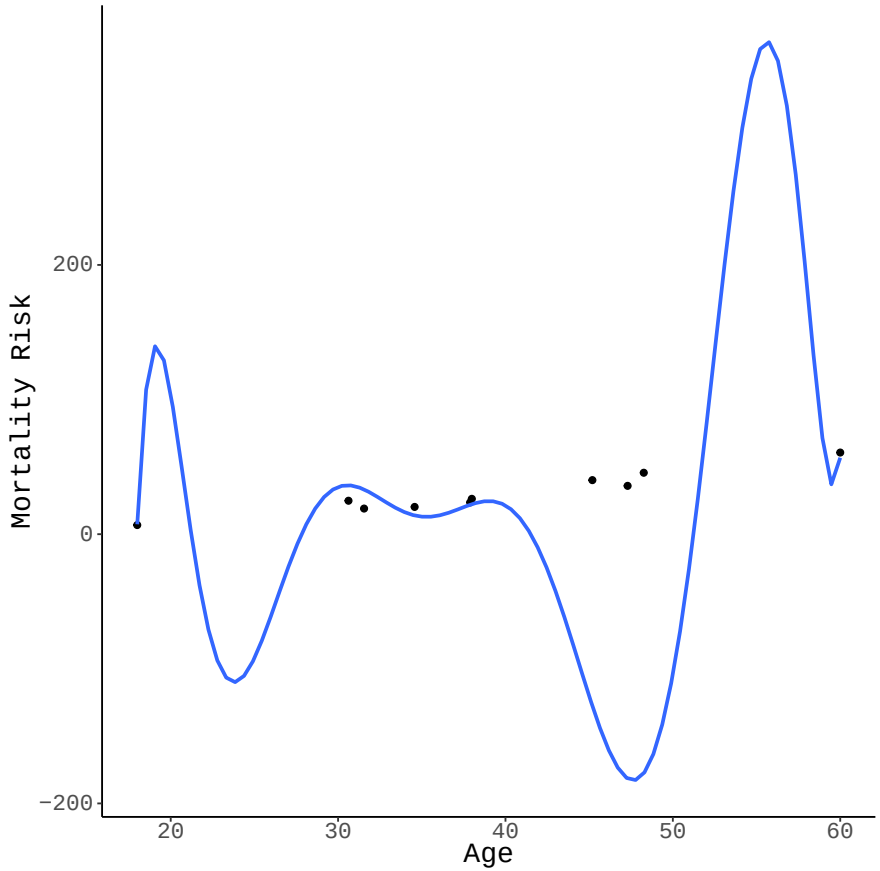
Should we be pleased to be able to perfectly predict mortality risk?

- Is our model useful?
- What happens if we try to apply our fitted model to new data?

Consequences of Over-fitting

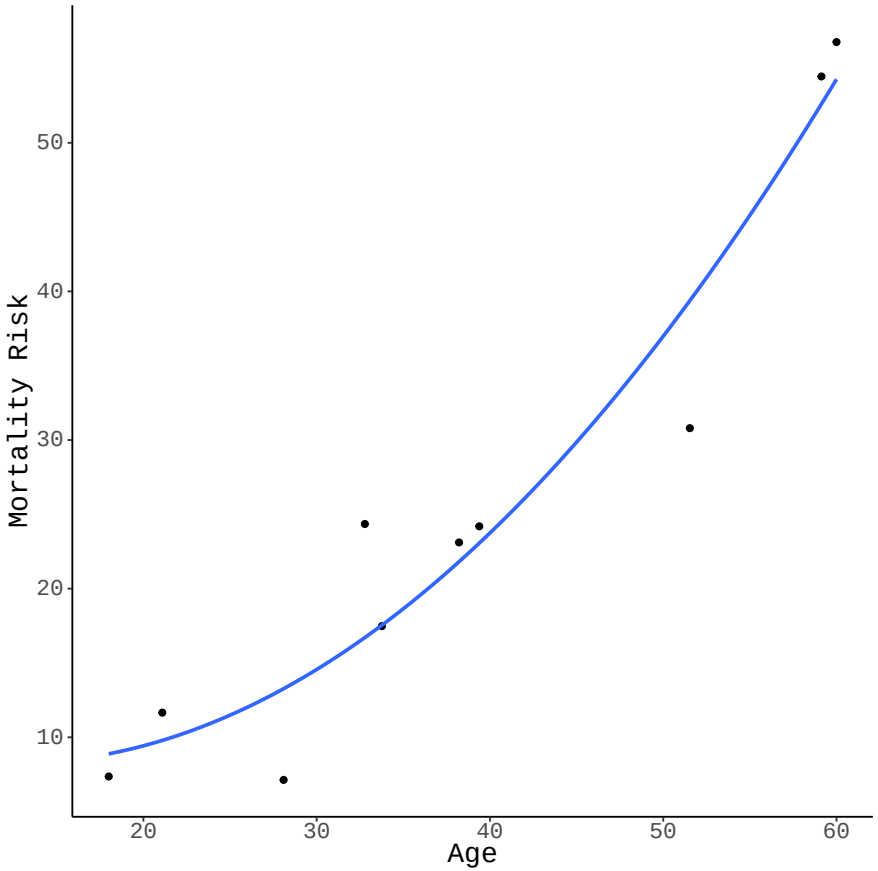
Should we be pleased to be able to perfectly predict mortality risk?

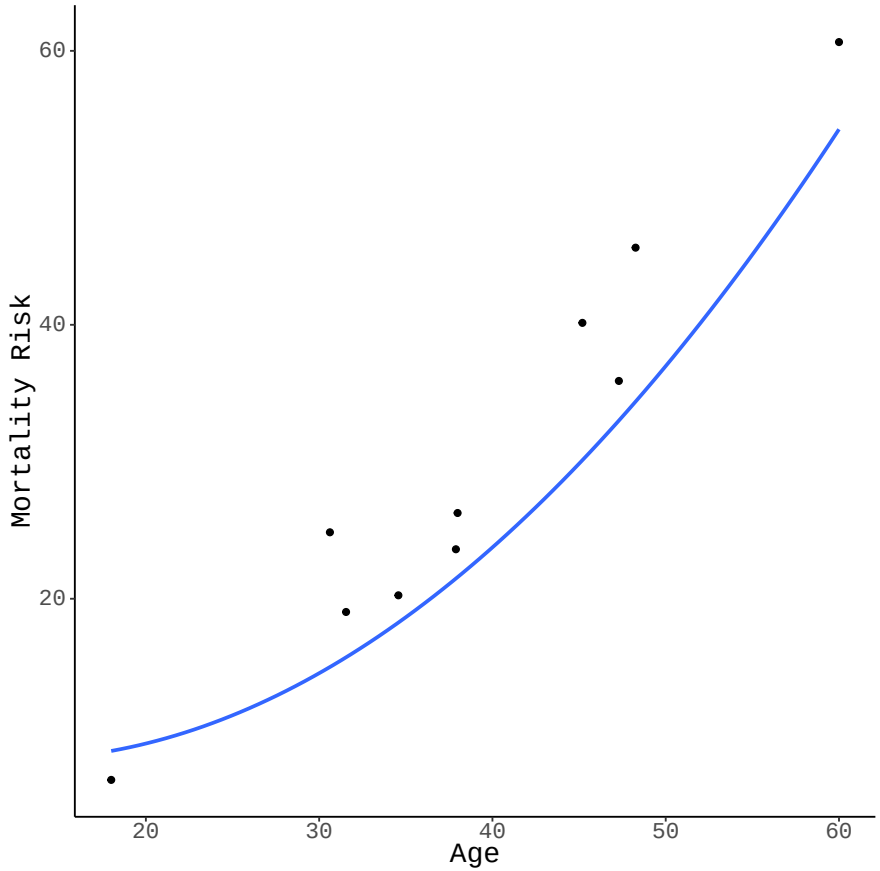
- Is our model useful?
- What happens if we try to apply our fitted model to new data?



Correct Fit

Let's try something a bit more reasonable.





A Sensible Goal

Our goal is to train a model that can best predict *new data*.

- The predictive performance on the training data is immaterial.
- We can always fit the training data arbitrarily well.
- Fit to the training data will always be at-odds with fit to future data.

This conflict the driving force behind the *bias-variance trade-off*.

Model Fit for Prediction

When assessing predictive performance, we will most often use the *mean squared error* (MSE) as our criterion.

$$
\begin{aligned}
MSE &= \frac{1}{N}\sum_{n=1}^{N}\left(Y_n - \hat{Y}_n\right)^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\left(Y_n - \hat{\beta}_0 - \sum_{p=1}^{P}\hat{\beta}_p X_{np}\right)^2 \\
&= \frac{RSS}{N}
\end{aligned}
$$

Training vs. Testing MSE

The MSE on the preceding slide is computing based entirely on training data.

- *Training MSE*

What we want is a measure of fit to new, *testing* data.

- *Testing MSE*
- Given M new observations $\{Y_m, X_{m1}, X_{m2}, \ldots, X_{mP}\}$, and a fitted regression model, $f(\mathbf{X})$, defined by the coefficients $\{\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_P\}$, the *testing MSE* is given by:

$$MSE = \frac{1}{M}\sum_{m=1}^{M}\left(Y_m - \hat{\beta}_0 - \sum_{p=1}^{P}\hat{\beta}_p X_{mp}\right)^2$$

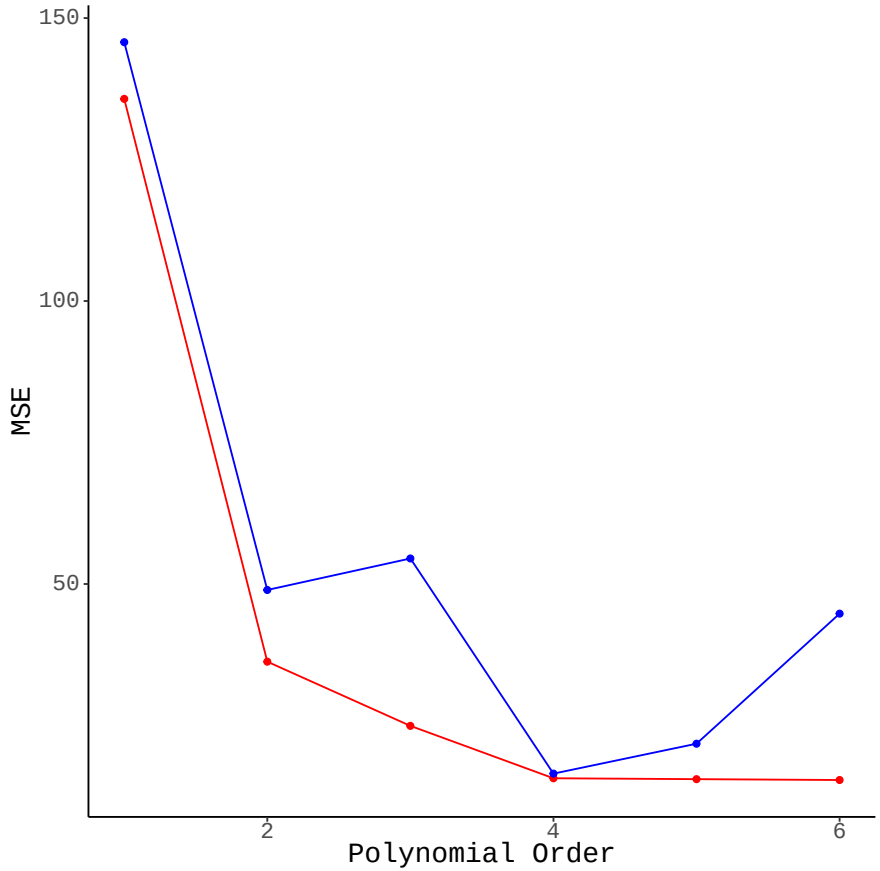
Training vs. Testing MSE

Training MSE will always decrease in response to increased model complexity.

- Note the red line in the plot

Testing MSE will reach a minimum at some "optimal" level of model complexity.

- Further complicating the model will increase the testing MSE.
- Note the blue line.



Training vs. Testing MSE

At the end of our model building example, we compared the following two models:

$$Y_{BP} = \beta_0 + \beta_1 X_{age} + \beta_2 X_{LDL} + \beta_3 X_{HDL} + \beta_4 X_{BMI} + \varepsilon \quad (1)$$

$$Y_{BP} = \beta_0 + \beta_1 X_{age} + \beta_2 X_{BMI} + \varepsilon \quad (2)$$

- The ΔR^2 test suggested that the loss in fit between Model 1 and Model 2 was trivial.
- The AIC and BIC both suggested that Model 2 should be preferred over Model 1.
- The training MSE values suggested that Model 1 should be preferred.

What happens when we do the comparison based on testing MSE instead of training MSE?

Training vs. Testing MSE

```r
set.seed(235711)

## Split data into training and testing sets:
ind  <- sample(1 : nrow(diabetes))
dat0 <- diabetes[ind[1 : 400], ]   # Training data
dat1 <- diabetes[ind[401 : 442], ] # Testing data

## Fit the models:
outF <- lm(bp ~ age + bmi + ldl + hdl, data = dat0)
outR <- lm(bp ~ age + bmi, data = dat0)

## Compute training MSEs:
trainMseF <- MSE(y_pred = predict(outF), y_true = dat0$bp)
trainMseR <- MSE(y_pred = predict(outR), y_true = dat0$bp)
```

Training vs. Testing MSE

```
## Compute testing MSEs:
testMseF <- MSE(y_pred = predict(outF, newdata = dat1),
                y_true = dat1$bp)
testMseR <- MSE(y_pred = predict(outR, newdata = dat1),
                y_true = dat1$bp)
```

Compare the two approaches:

	Full	Restricted
Train	147.72	148.44
Test	141.25	138.02

MSE Values

Cross Validation

Cross Validation

To train a model that best predicts new data, we can use *cross-validation* to evaluate the expected predictive performance on new data.

1. Split the sample into two, disjoint sub-samples
   - *Training* data
   - *Testing* data
2. Estimate a candidate model, $f(\mathbf{X})$, on the training data.
3. Check the predictive performance of $\hat{f}(\mathbf{X})$ on the testing data.

Cross Validation

To train a model that best predicts new data, we can use *cross-validation* to evaluate the expected predictive performance on new data.

1. Split the sample into two, disjoint sub-samples
   - *Training* data
   - *Testing* data
2. Estimate a candidate model, $f(\mathbf{X})$, on the training data.
3. Check the predictive performance of $\hat{f}(\mathbf{X})$ on the testing data.

We can use this idea to select the best model from a pool of candidate models, $\mathcal{F} = \{f_1(X), f_2(X), \ldots, f_J(X)\}$

1. Repeat Steps 2 and 3 for all candidate models in $\mathcal{F}$.
2. Pick the $\hat{f}_j(\mathbf{X})$ that best predicts the testing data.

Different Flavors of Cross-Validation

In practice, the split-sample cross-validation procedure describe above can be highly variable.

- The solution is highly sensitive to the way the sample is split because each model is only training once.

Split-sample cross-validation can also be wasteful.

- We don't need to set aside an entire chunk of data for validation.

In most cases, we will want to employ a slightly more complex flavor of cross-validation:

- *K-fold cross-validation*

K-Fold Cross-Validation

1. Partition the data into K disjoint subsets $C_k = C_1, C_2, \ldots, C_K$.

K-Fold Cross-Validation

1. Partition the data into K disjoint subsets $C_k = C_1, C_2, \ldots, C_K$.
2. Conduct K training replications.
   - For each training replication, collapse $K - 1$ partitions into a set of training data, and use this training data to estimate the model.
   - Compute the test MSE for the kth partition, MSE_k, by using subset C_k as the test data for the kth fitted model.

K-Fold Cross-Validation

1. Partition the data into K disjoint subsets $C_k = C_1, C_2, \ldots, C_K$.
2. Conduct K training replications.
   - For each training replication, collapse $K - 1$ partitions into a set of training data, and use this training data to estimate the model.
   - Compute the test MSE for the kth partition, MSE_k, by using subset C_k as the test data for the kth fitted model.
3. Compute the overall *K*-fold cross-validation error as:

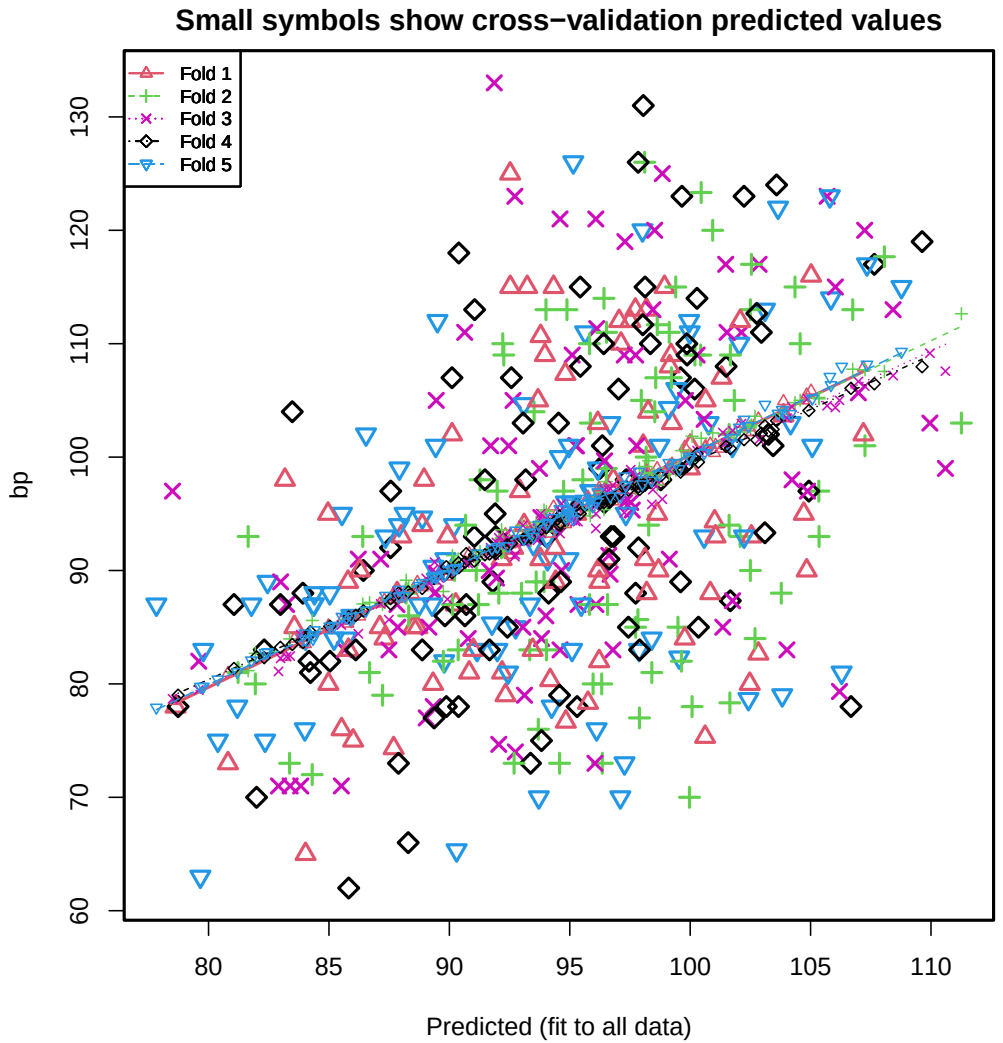
$$CVE = \sum_{k=1}^{K} \frac{N_k}{N} MSE_k,$$

Applying K-Fold CV to our Example

```
cvOutF <- CVlm(data = diabetes,
               form.lm = outF,
               m = 5,
               printit = FALSE,
               seed = 235711)
```

```
## Estimated CVE:
attr(cvOutF, "ms")

[1] 150.8718
```



Applying K-Fold CV to our Example

```
cvOutR <- CVlm(data = diabetes,
               form.lm = outR,
               m = 5,
               printit = FALSE,
               seed = 235711)
```

```
## Estimated CVE:
attr(cvOutR, "ms")

[1] 149.6954
```