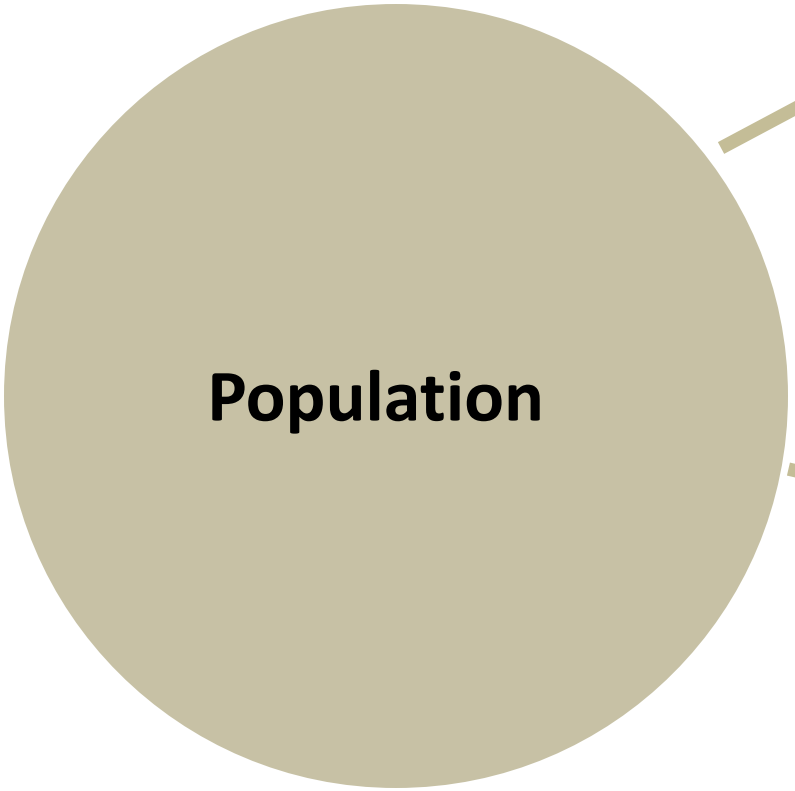
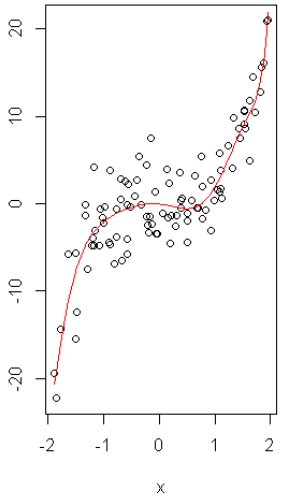
Cross-validation

STAT 315, 27/03

In the example last time, we compared the models by their performance on an independent test set from the same population as the data

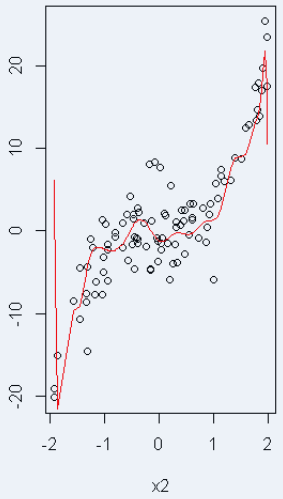


Data



Fit (aka train)

Test



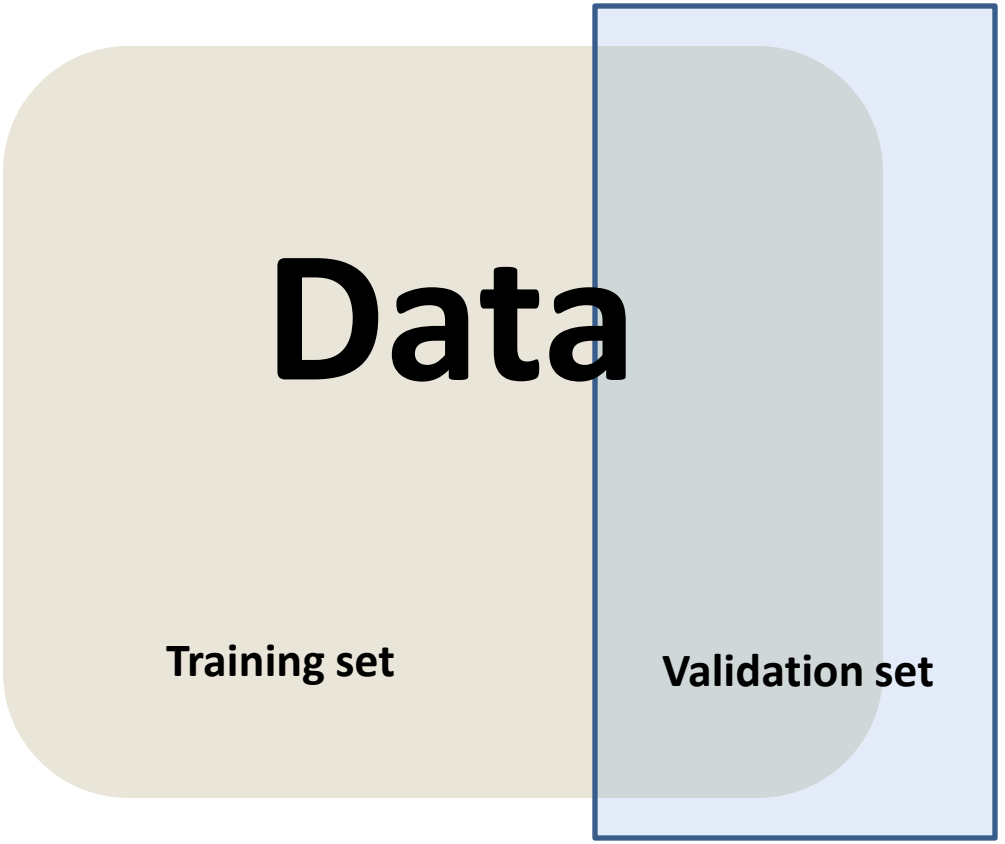
Oops!

We can’t always do this because

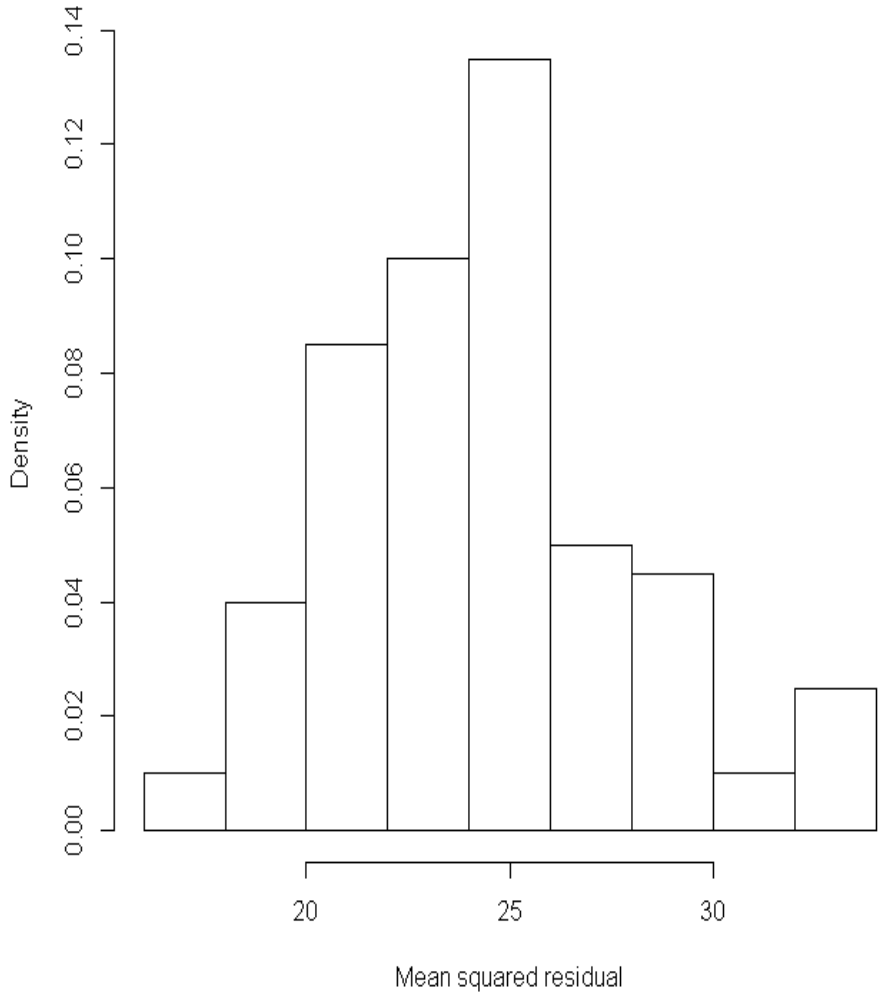
- We don’t know the population from which the data came, so we can’t sample from it.
- We don’t know the process which generated the data.

Idea: set aside some of the data. Fit the model to the rest. Use the set-aside data for validation.

Validation set

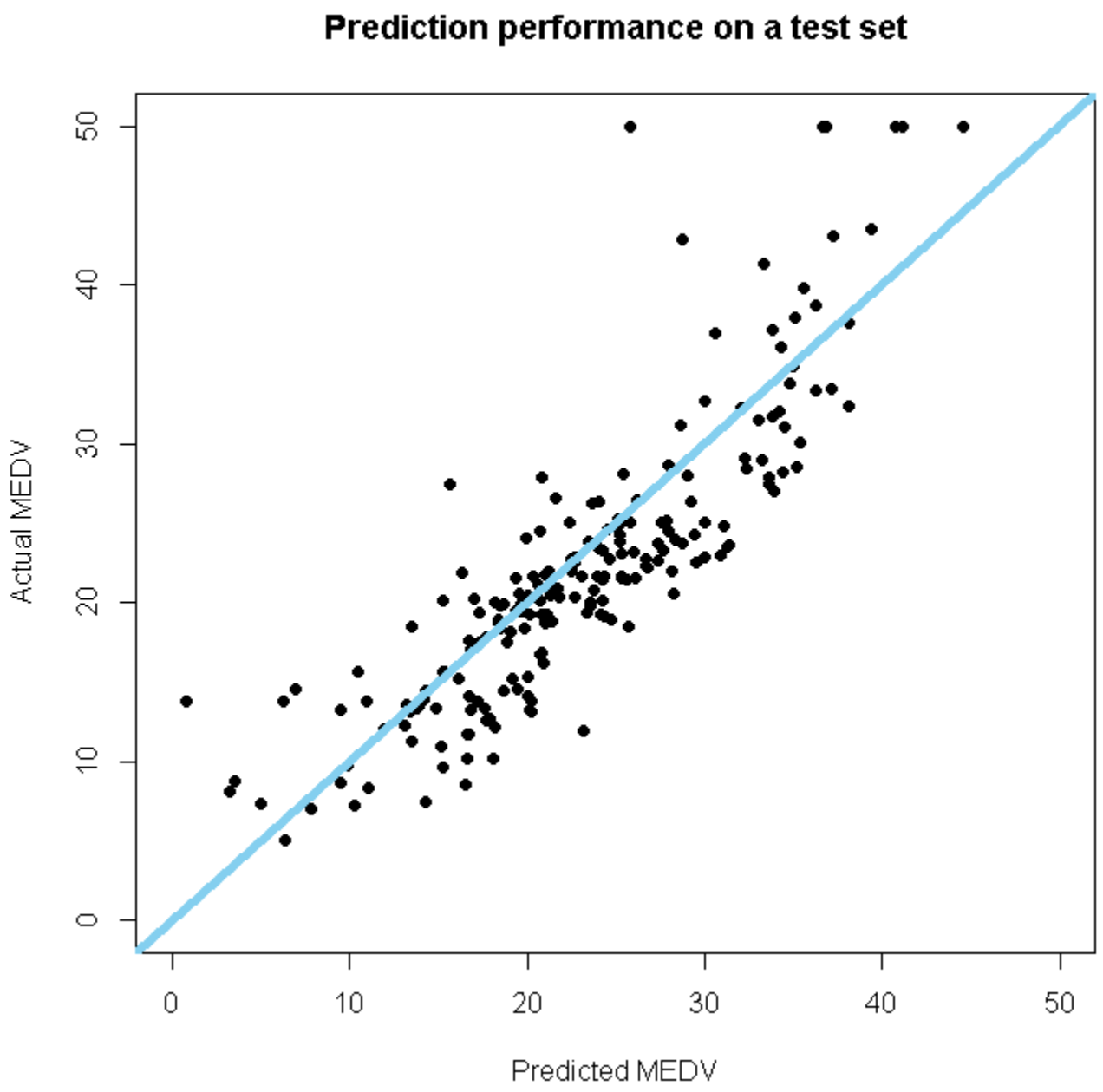


Example: Boston housing data. Hold out 206 of 506 obs as validation set. Train on 300 obs and calculate mean squared residual (=RSS/206)



Main problem:

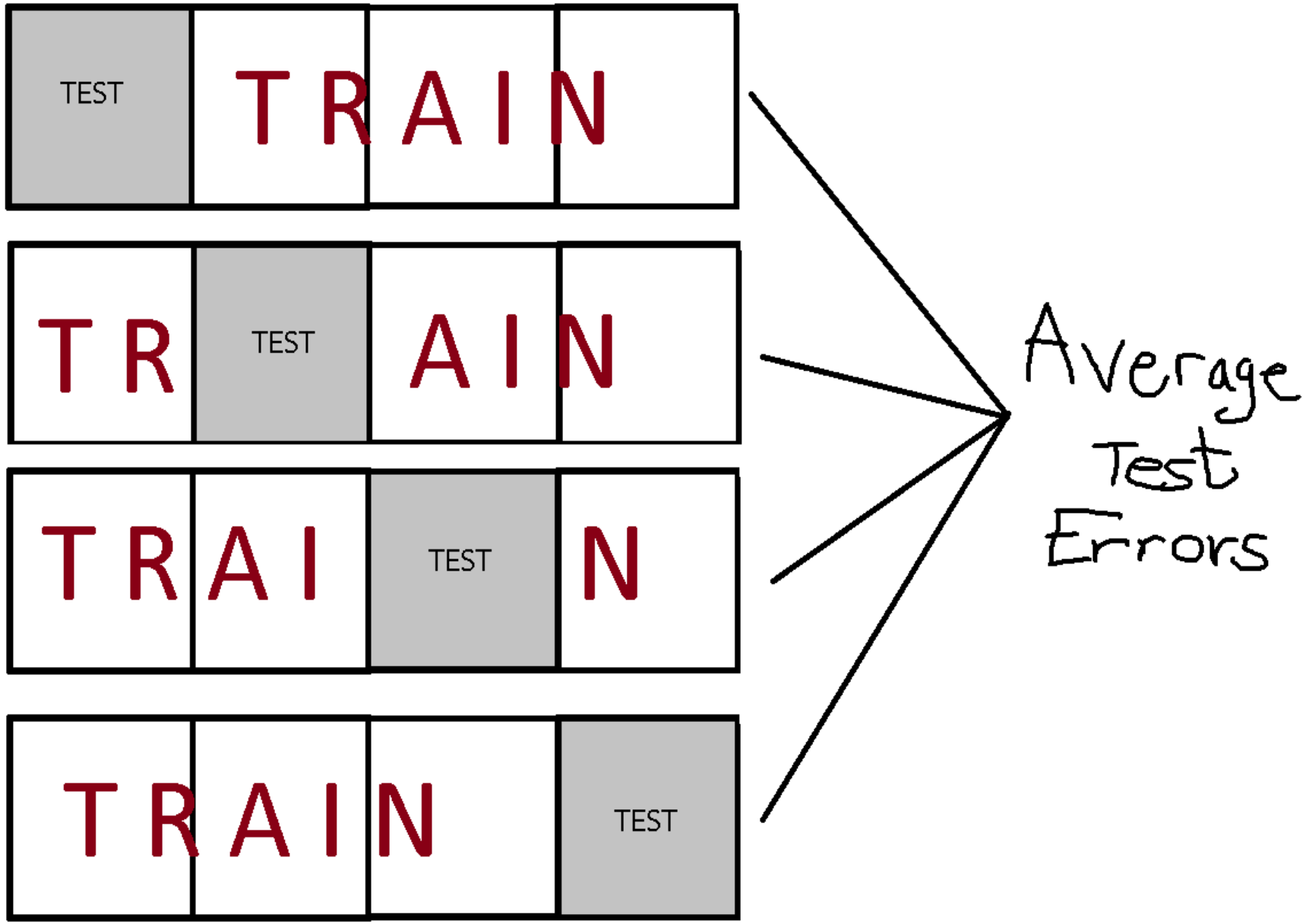
σ = 3.6; rather wide


Prediction performance on a test set
Actual MEDV
Predicted MEDV
0
10
20
30
40
50

A better way of getting an estimate of the MSE on a test set is to hold out many different test sets and then average them.

Cross-validation

is an efficient way of doing this.

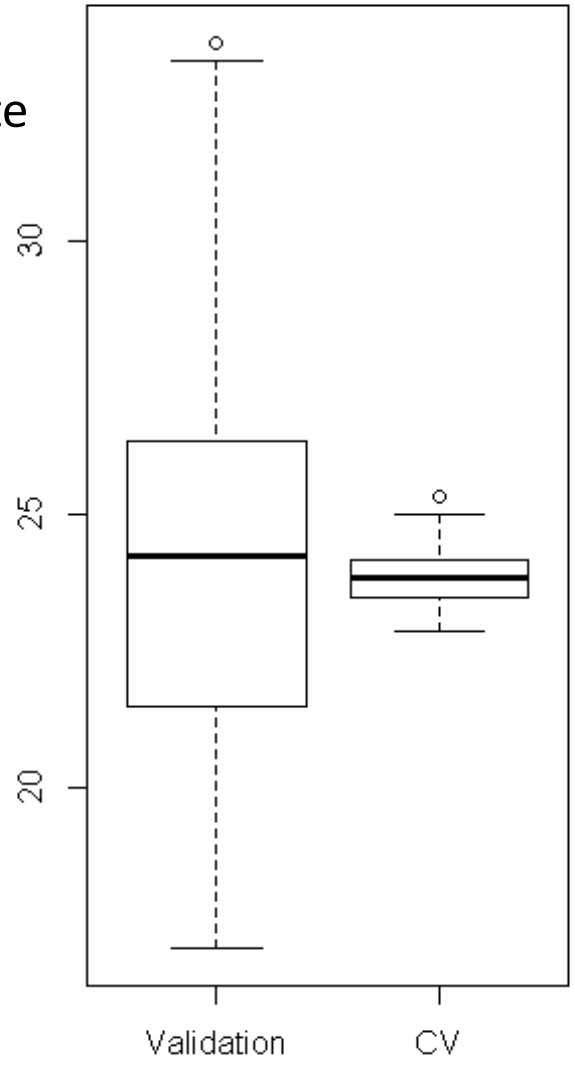

TEST
TRAIN
TR
TEST
AIN
TRAI
TEST
N
TRAIN
TEST
Average
Test
Errors

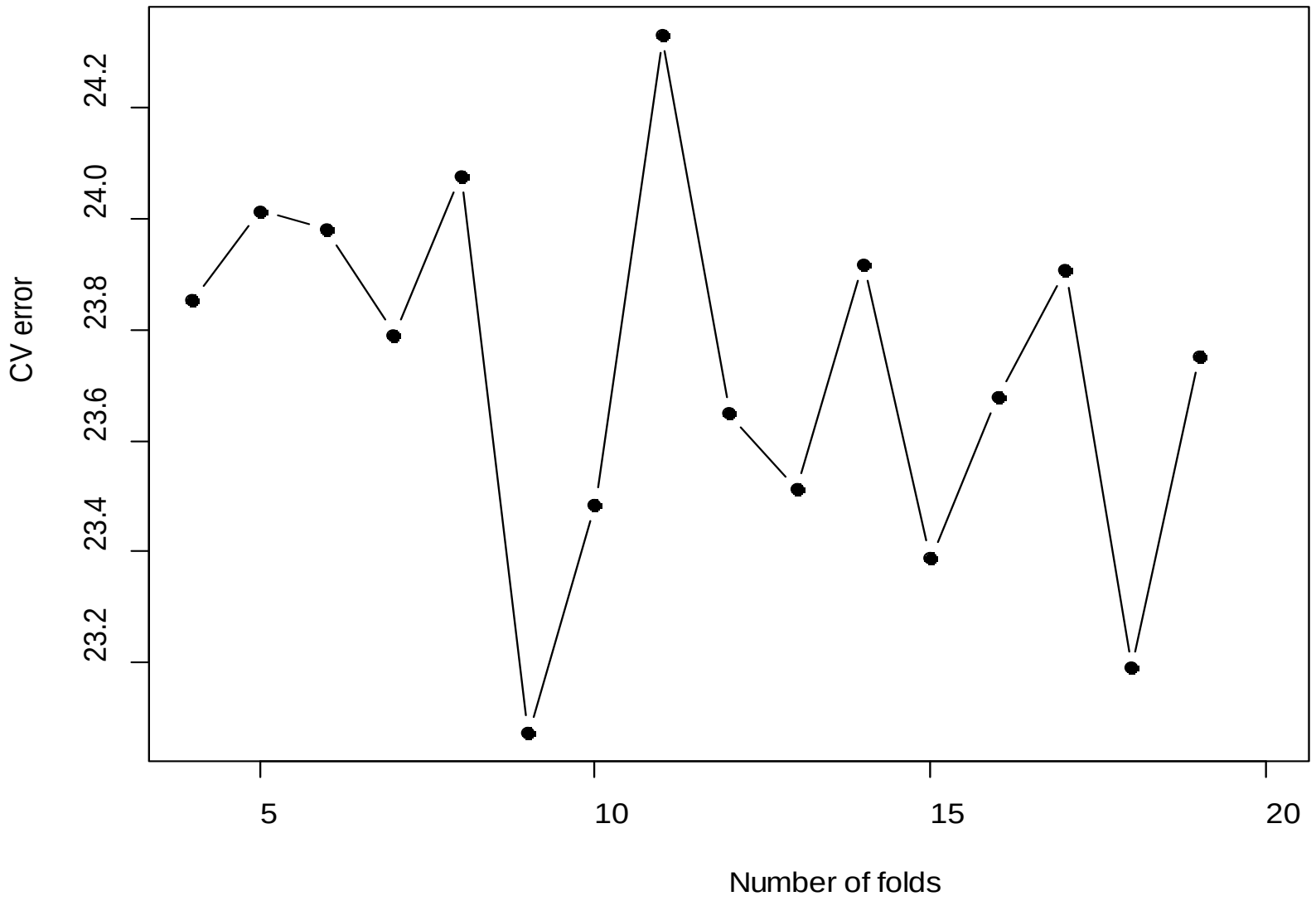
cv_k = test error on kth fold

$\overline{cv} = \frac{1}{k}\sum cv_k$ = cv estimate of test error

$\sqrt{\frac{1}{k(k-1)}\sum(cv_k - \overline{cv})^2}$ = standard error of cv estimate

One of the prices paid for more precise estimates from CV (decreased variance) is that CV estimates tend to be biased. Hastie and Tibshirani recommend 5- or 10-fold cross-validation. Witten and Frank recommend 10-fold cross-validation repeated ten times (on different random splits of the data set) with the standard error = std dev of the 10 replications.




CV error
23.2
23.4
23.6
23.8
24.0
24.2
5
10
15
20
Number of folds

Something to be careful about: In CV, the whole model-fitting process has to be applied to the training set in each of the k stages. It's not OK to pre-screen using the entire training set and *then* use cross-validation.

- One reason why subset selection methods have a bad reputation; they can be hard to validate.

- But "unsupervised screening" is OK. Example: out of 2000 variables in gene expression data set, pick the 100 which have the highest variance and use them in the model.

LOOCV

In a data set with n observations, n-fold CV is called leave-one-out cross validation or LOOCV.

For linear regression, the LOOCV error can be computed exactly from a single model fit using the formula

$$\frac{1}{n}\sum \frac{(y_i - \hat{y}_i)^2}{(1 - h_{ii})^2}$$

Where h_{ii} is the ith diagonal element of the hat matrix.

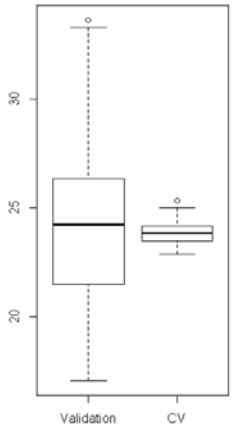
In R:

```
model <- lm(medv~., data=Boston)
PRESS <- sum(resid(model)^2/(1-influence(model)$hat)^2)
(1/nrow(Boston))*PRESS
```

In SAS:

```
proc reg data = boston;
 model MEDV = CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO B LSTAT /
influence;
run;
```

- Look through the output until you find PRESS.
- Divide PRESS by the number of observations.



Answer: 23.7

Cross-validation is great because:

- It directly measures the thing you are really trying to measure (predictive performance on unseen data).
- It often actually works.
- You can use it to select tuning parameters for more complex models.
- It allows you to compare any two models (provided that you can obtain predictions from these models. *[But that's crazy! What kind of model would not allow you to make a prediction? Stay tuned.]*) Compare ANOVA, which we looked at earlier in the course, but only gives you a way to compare nested models.

Example:

Data analysis tip: Generally if some variable is measured in dollars, you should take its log.

In the Boston data, do we get better performance by predicting log(MEDV) or just MEDV?

Answer: from 10x10 fold cross-validation,

Model MEDV ~.
CV estimate of MSE on unseen data: 24 +/- 0.5

Model log(MEDV) ~.
CV estimate of MSE on unseen data: 19 +/- 0.4

So yes, it looks like it's better to use log(MEDV) as the dependent variable if we are set on using linear regression.

Thanks, Cross-validation!