Data Science Methods for Clean Energy Research

Week 7 L2: Multiple regression, nonlinear regression and resampling

Feb 13, 2017

You will need the HCEPD_100K.csv file (or the path to it) for python example time today

UNIVERSITY of WASHINGTON

Outline

- Quick review from last time
- Multiple regression
- Python example of simple and multiple regression
- Nonlinear regression
  - Python implementation
- Resampling methods (will do this first, depending on time)
  - Cross-validation
  - Bootstrapping
  - Python examples (if time)

Topics last time

- Simple linear regression and the origin of our fit coefficients
- Assessing the quality of our fit coefficients
- Assessing the error in our model

Accuracy of the model – how are we doing overall?

> **Residual standard error**

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \qquad (3.15)$$

– **A measure of the** lack of fit **of your model (**<u>in units of Y!</u>**)**

> **R^2 statistic**

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} \qquad (3.17) \qquad \mathrm{TSS} = \sum (y_i - \bar{y})^2$$

– **A scale invariant measure (0-1 range) that explains "***the proportion of the variability of Y that is explained by X***"**

– **Lets chat about TSS and what it means…**

The correlation

- **Recall the basic descriptor – correlation coefficient or simply** correlation, **which we use to describe trends in our data and relationship between variables**

$$\mathrm{Cor}(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}, \qquad (3.18)$$

Multiple regression

> Concept: independently assess the variation in Y with different values of X:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon, \qquad (3.19)$$

> As with SLR, the coefficients are determined by setting the analytical partial derivatives to zero and solving the resultant *p+1* linear equations

> As with SLR there is an exact solution

– Will not show math, but easily found online

Python break /examples

- <<instructions>>

Key questions with multi-parameter fits

- Section 3.2.2 in ISL does an excellent job of discussing the following four key questions in the context of an MLR fit for marketing/sales data
  1. Is at least one of the predictors X1 , X2 , . . . , Xp useful in predicting the response?
  2. Do all the predictors help to explain Y, or is only a subset of the predictors useful?
  3. How well does the model fit the data?
  4. Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

4 Q directly copied from ISL page 75!

Big picture concepts

- **Temping to simply add one term for each feature in X and see how good of a fit we obtain, but that doesn't give us much inference**
- **There is a huge risk of** overfitting **with using a lot of parameters!**
- **We can use a new type of hypothesis test to find out if** any **of the parameters are significant**
- **We can use some algorithms (**selection algorithms**) to try and reduce the number of parameters**

Multiple regression and the F-score

- With large number of parameters (p) it is not useful to individually hypothesis on the individual β_i
  - Consider p=100, and the following hypothesis is true:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

  - By random chance, you can expect 5% of the individual P-values to be below 0.05!
  - Instead we can evaluate the entire hypothesis in one go using the F-statistic

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n - p - 1)}, \qquad (3.23)$$

Multiple regression and the F-score

$$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n - p - 1)}, \qquad (3.23)$$

- **The F-statistic has a** penalty **for increasing the number of parameters (this should make sense!)**
- **Recall from the t-statistic, we had a specific recipe to test the hypothesis at a certain significance level (e.g., α=0.05)**
- **Similar test is beyond the scope of this class, but you should look for F values:** at minimum > 1 **as n <u>decreases</u> and p <u>increases</u> the F values to show significance can take values >> 1**

Which of the variables are important?

- **Once we have some idea that at least one of the variables are important, how might we figure out what variables matter?**
- **5-10 min discussion (partner/table/group). Think about** algorithm!
- **Two basic concepts**
  - **Forward selection:** start with a null model ($\boldsymbol{y} = \boldsymbol{\beta_0}$) and add to it and find the min RSS
  - **Backward selection:** start with complete model (max p) and remove, in order, variables w/largest P-values
  - **The algorithm continues until a stopping rule is reached**
- **Selection algorithms foreshadow a need for more sophisticated methods (**subset selection and regularization**)**

Residual squared error (RSE) in MLR

- Our error metric for the goodness of fit of the model also includes a penalty for increasing p compared to n

$$\text{RSE} = \sqrt{\frac{1}{n-p-1}\text{RSS}}, \tag{3.25}$$

- With an multiple linear regression model in hand, you can make predictions and also trivially add confidence intervals, just as with simple linear regression

When good assumptions go bad

- **We haven't discussed it in detail but there are three key assumptions that have been used to build our linear regression model:**
  - **Errors are uncorrelated and normally distributed**
  - **The variance of the error (in Y) is independent of where we are in X**
  - **Liner relationship between X and Y (**the predictor-response relationship**)**
  - **Individual contributions of your X's are piecewise additive to the response**
- **These assumptions underlie many methods we commonly use!**

Understanding correlation in error

> The most common way that error becomes correlated is with time series data

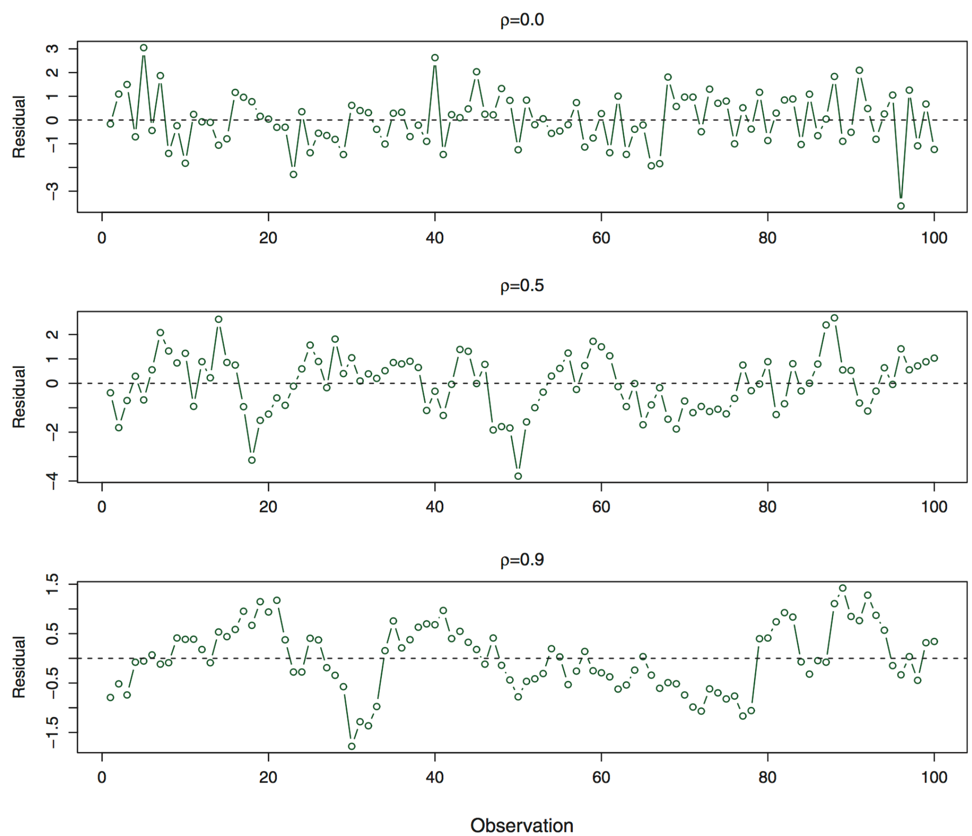
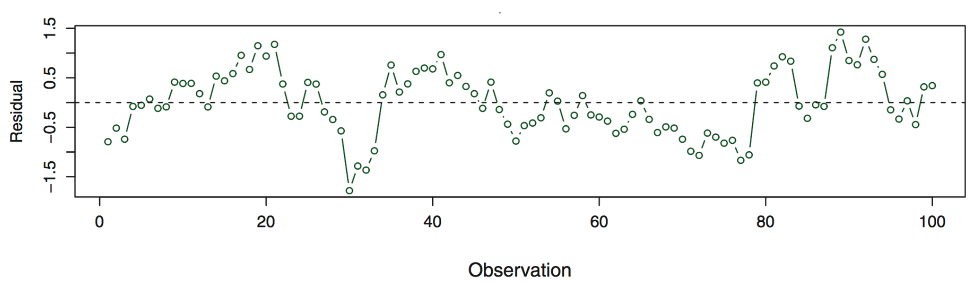


FIGURE 3.10. *Plots of residuals from simulated time series data sets generated with differing levels of correlation ρ between error terms for adjacent time points.*

Identifying correlation in error

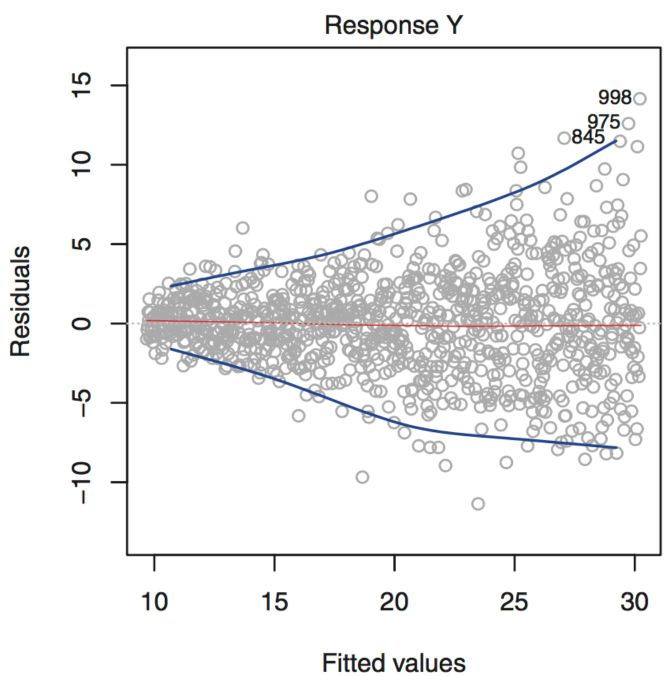
> **Note that the plot in the last slide was residual vs. observation (**always a good idea to plot this in addition to a histogram of your residuals – both whether our assumptions are in line**)**

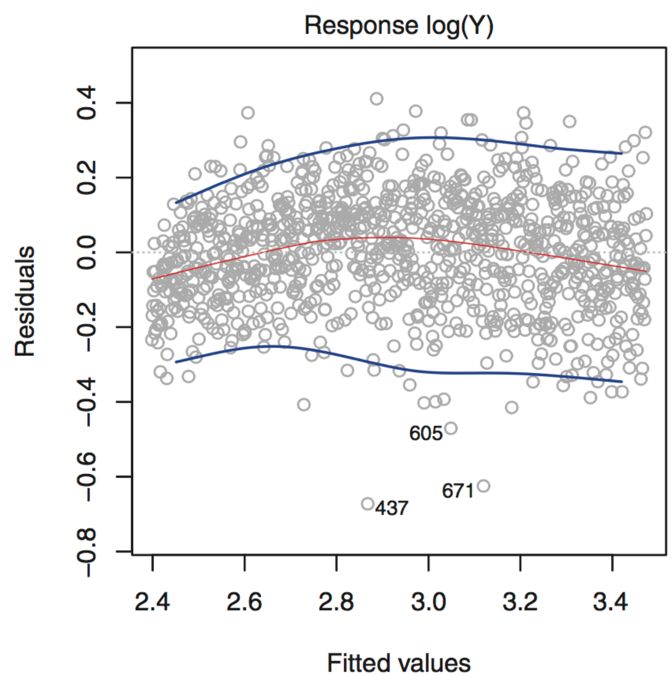
> **Introduction and practical implementation of methods to deal w/correlated errors is beyond scope of this class but we can discuss a few principles**
  – **The** autocorrelation **time**
  – **Getting the most out of your data**



Nonlinear variance

- **This phenomena is known as** heteroscedasticity
- **Transform data (as in Fig 3.11)**
- **Weigh the observables by their variance (e.g., $y_{i,new} = y_i / \sigma$)**

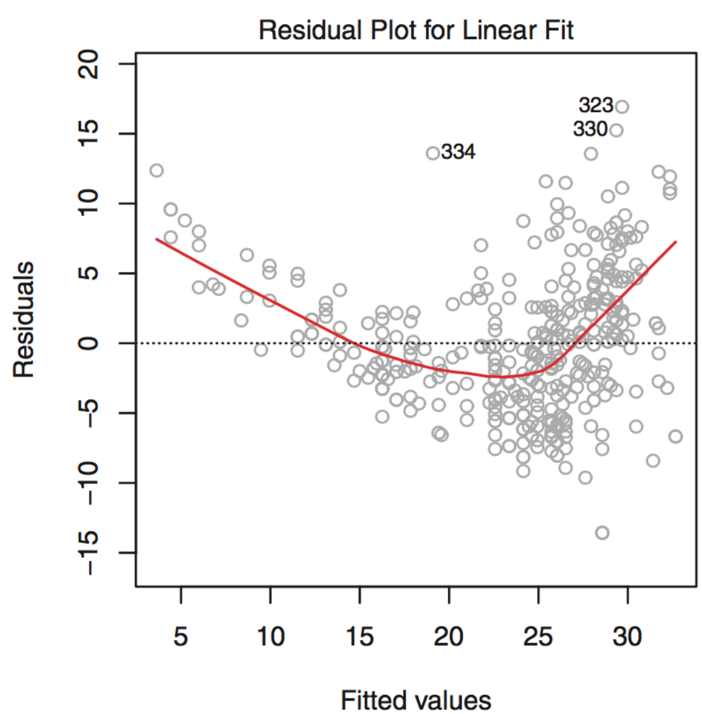


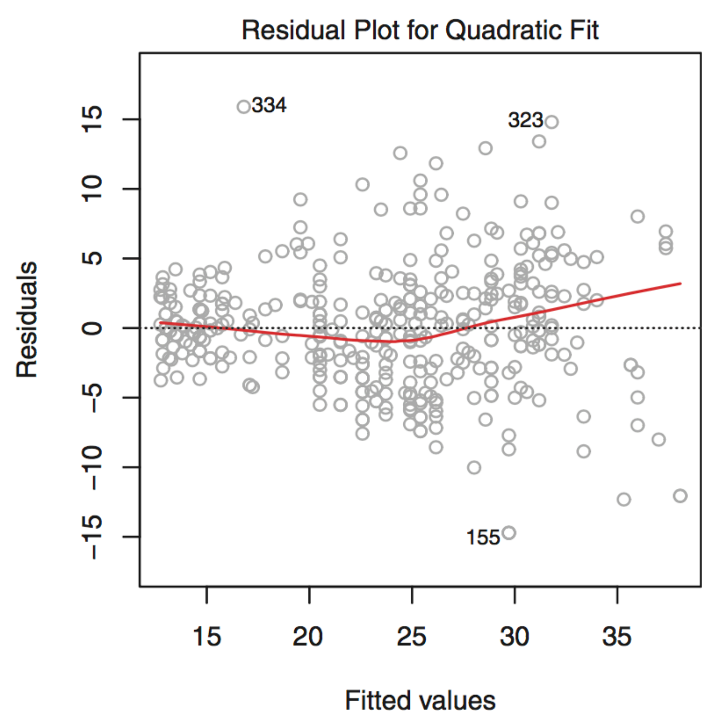


Intro to nonlinear regression

> Sometimes your variables have a clear non-linear dependence on the response

> This is especially clear when you look at the residuals (ISL Fig 3.9)





Simple types nonlinear regression

> **Sometimes a simple variable transformation can take care of nonlinear terms in our regression (e.g., $X_{i,new} = \sqrt{X_i}$)**
  - **In these cases we can use the same pipeline/framework for large scale MLR approaches**
  - **This would be performed** prior **to training your model**

> **Sometimes there is a clear co-dependence** (interaction) **on several of your features/variables**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon.$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon. \tag{3.31}$$

Where to go when you have a true nonlinear model to fit/train?

- **In general, the packages in scikit-learn work for general linear (single or multiple) or very specific types of nonlinear (e.g., splines) models of Y**
  - **The reason why has to do with speed of model solution and large training data set sizes**
- **Other Python packages can also perform regression (**See W4L1L2.ipynb in my notes!**)**

scipy.optimize.curve_fit

scipy.optimize.curve_fit(*f*, *xdata*, *ydata*, *p0=None*, *sigma=None*, *absolute_sigma=False*, *check_finite=True*, *bounds=(-inf, inf)*, *method=None*, *jac=None*, ***kwargs***)** [source]

Moving from exact solution to finding a minimum MSE

- When we leave the world of ”exact solutions”, we are then confined to using numerical solutions to find the minimum value of MSE(β_i)
- Do we need an initial guess? (yes!)
  - What should it be?
- A lot of our machinery for assessing the models still works just fine!

$$\mathrm{RSE} = \sqrt{\frac{1}{n-p-1}\mathrm{RSS}}, \tag{3.25}$$

Other topics / suggestions

- Chapter 3 of ISL is strongly suggested to read carefully (maybe multiple times)
- Additional topics we didn't cover
  - Outliers and high leverage points in your training set
  - Collinearity
  - More about nonlinear regression
  - AND MANY MORE!

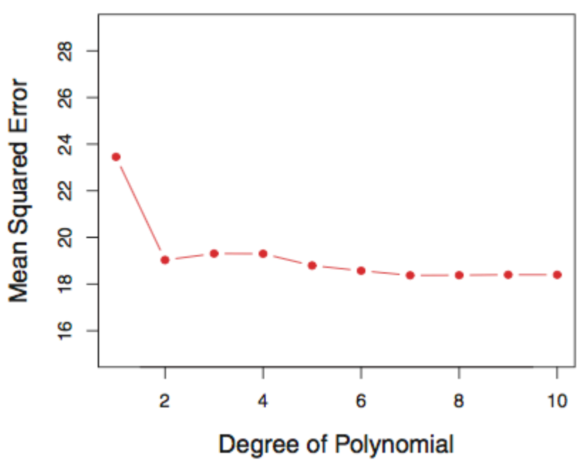
Resampling methods (CH5, ISL)

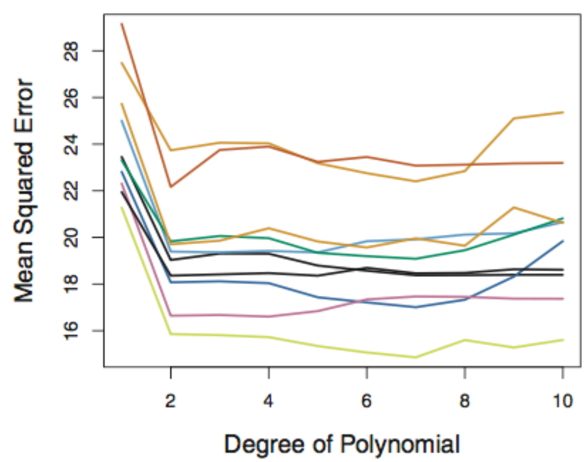
- **Resampling concept**
- **Doing more with your data**

- **A big warning:** as introduced today, the resampling schemes are not to be used to generate independent predictions (**of Y**) for averaging later.
- This is a concept related to ‘ensemble’ methods, which we will discuss soon

Cross Validation (k-fold)

- **Suppose you have <u>one</u> set of data and you have to decide how to break it into pieces for** training **and** validation
- **Simplest approach is the "validation set" , just break it into two pieces**
- **Example (Fig 5.2) looking at variations on** $Y = \beta_0 + \beta_0 + X^n$





Left: <u>training</u> MSE vs n for one data set
Right: <u>training</u> MSE vs n for 10 validation sets

Riddle me this, how many ways are there to choose two 500 data sets from 1000?

Cross Validation (k-fold)

- **Since you only use a portion of your data in the training, the "validation set" approach will tend to** overestimate **your error!**
- **Leave One Out Cross Validation approach**

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i. \qquad (5.1)$$

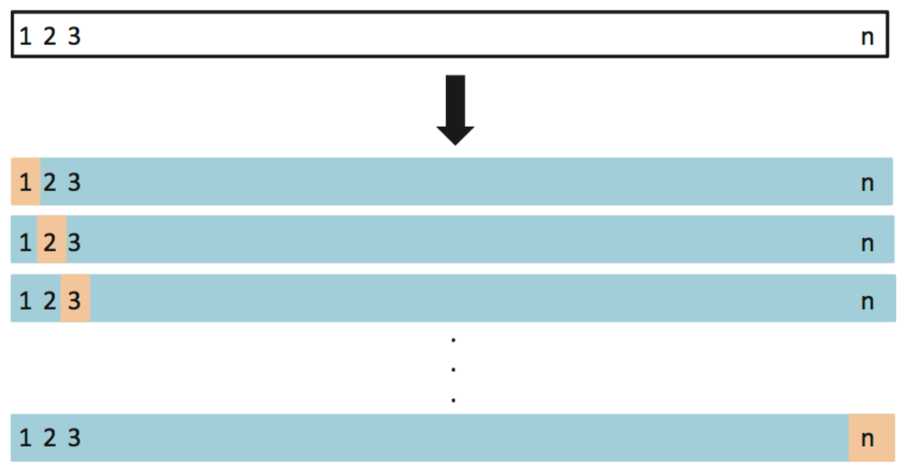


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

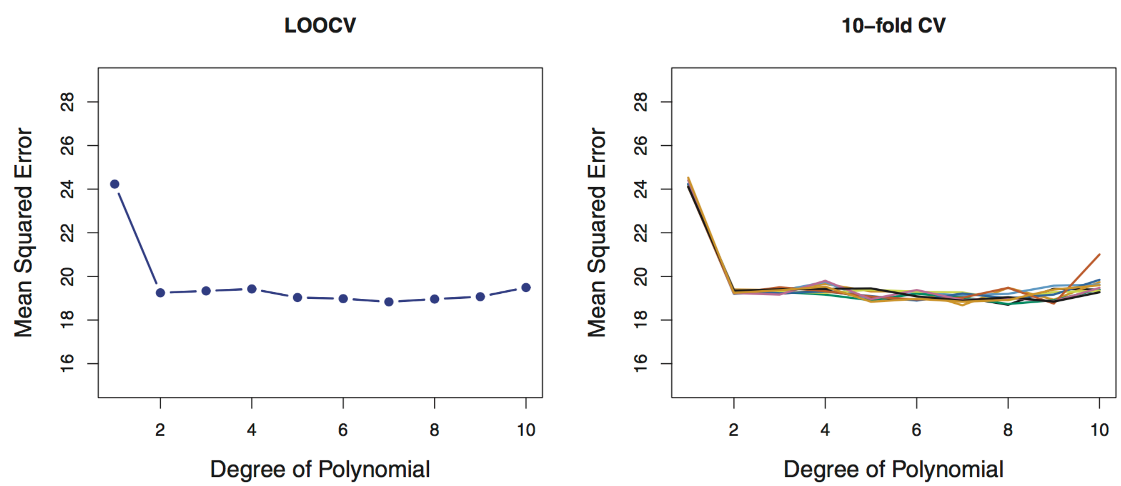
Cross Validation (k-fold)

> **LOOCV is way more accurate (Fig 5.4), but more computationally expensive!**

> **An alternate is to break the data into larger pieces than n and n-1**

> **We break it into "k" folds of data, e.g. 5-fold. The 1st set is saved for** validation, **remaining k-1 sets are used for** training

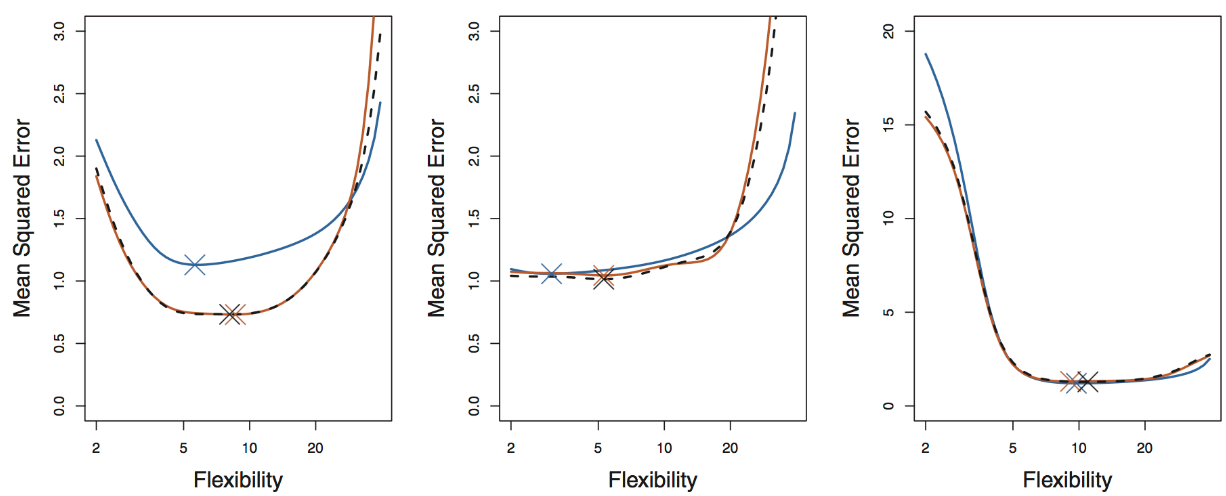
$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i. \quad (5.3)$$



Bias/variance tradeoff: use 5 or 10 folds

- Can anyone recall what we meant by the bias/variance tradeoff?
- Empirically people usually use 5 or 10 folds to avoid too much bias or variance in their resampling algorithm
- This is a great way to get a true estimate of your model's MSE

Fig 5.6 revisits Fig 2.9 in the context of k-fold cross validation



Bootstrap

- **The bootstrap is one of the most versatile tools you will use in statistical analysis of data sets**
- **It involves resampling** with replacement **from your data set**
- **Algorithm:**
  - **Randomly draw, with replacement, some subset from your training data**
  - **Train your model and make an estimate of your coefficient and MSE**
  - **Rinse and repeat until the errors converge**

The power of the bootstrap in one figure

> Fig 5.10, estimates of some parameter, α

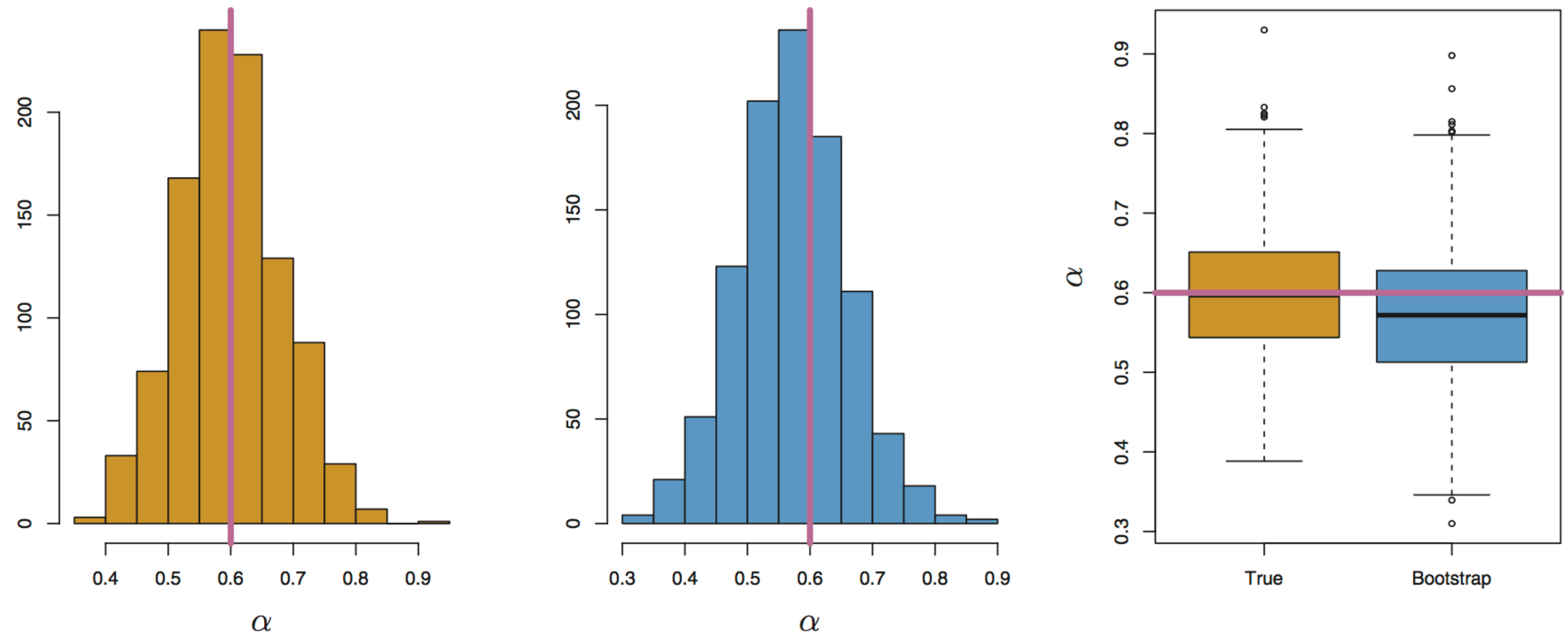


FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

Take care

- **Depending on how large your bootstrap sample data set is, I recommend you avoid using the standard error formula (Eq 5.8) and instead you should use simply the standard deviation of the bootstrap estimates.**
  - Can anyone explain why?
- **In this context α, could be any quantity from your training procedure (MSE, β, etc..)**

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}. \tag{5.8}$$

What’s next?

> HW 4!