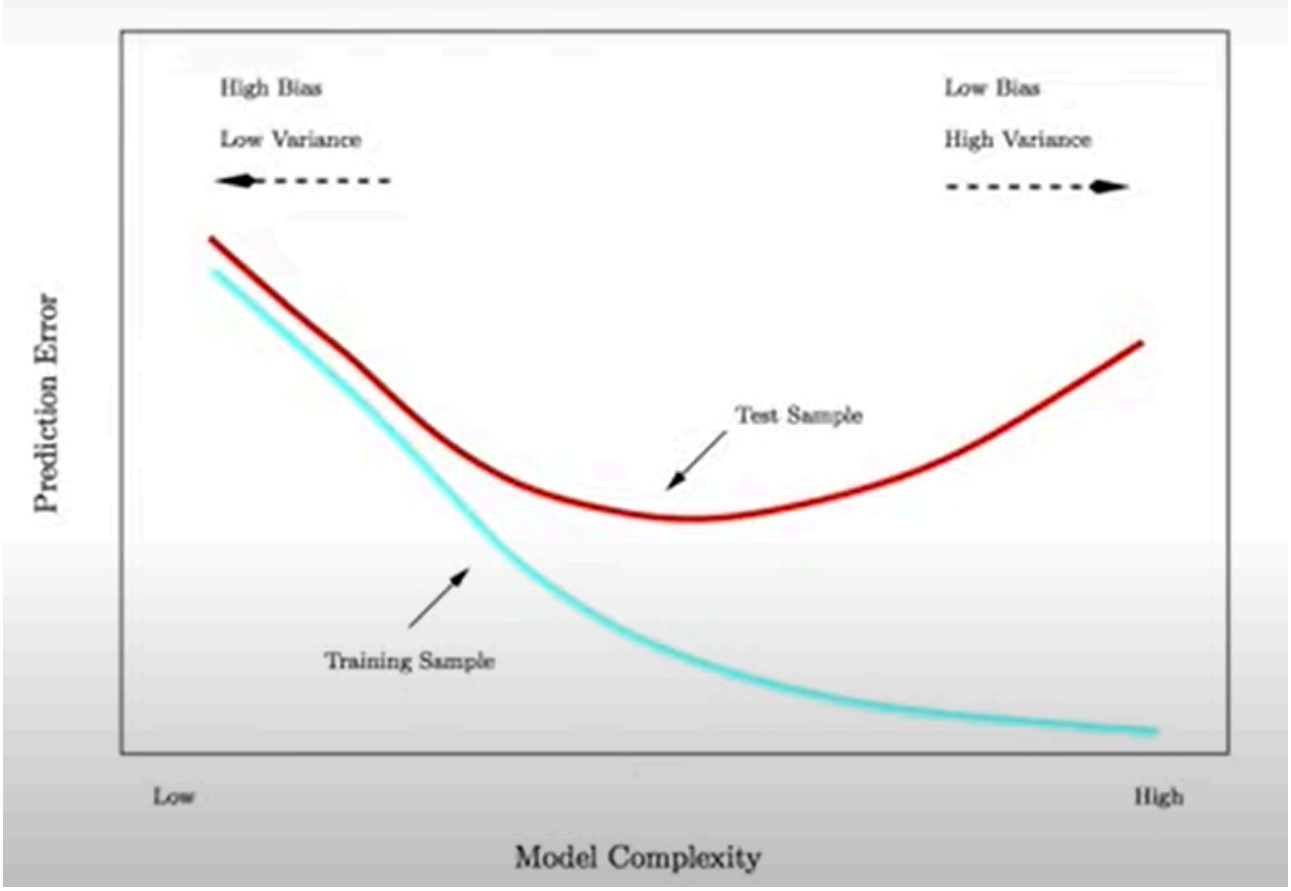
# Chapter_5_Resampling_Methods

- In this chapter we discuss two resampling methods: Cross-validation and Bootstrap
- These methods refit a model of interest to samples formed from the training set , in order to obtain additional information about the fitted model.
- For example, they provide estimates of test-set prediction error, and the standard deviation and bias of our parameter estimates.

## Training Error versus Test Error

- The **test error** is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
- The **training error** can be easily calculated by applying the statistical learning method to the observations used in its training.
- But training error and test error are very different and in particular the former can **dramatically underestimate** the latter.




**Bias:** how far off on the average the model is from the truth.
**Variance:** how much that estimate varies around its average.

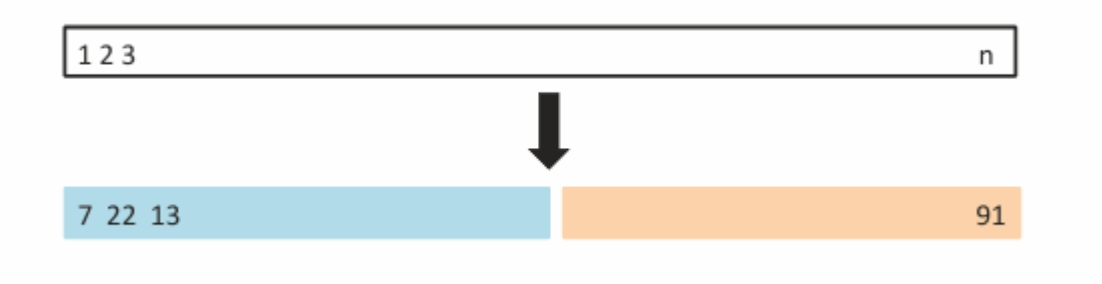
So what do we do?

- Best solution: a large designated test set, often not available.
- Some methods make a mathematical adjustment to the training error rate in order to estimate the test error rate. These include the $C_p$ $statistic$ , $AIC$ , and $BIC$ .They are discussed elsewhere.
- Here we instead consider a class of methods that estimate the test error by **holding out** a subset of training observations from the fitting process, and then applying the statistical learning method to those held out observations.

# Validation-set approach

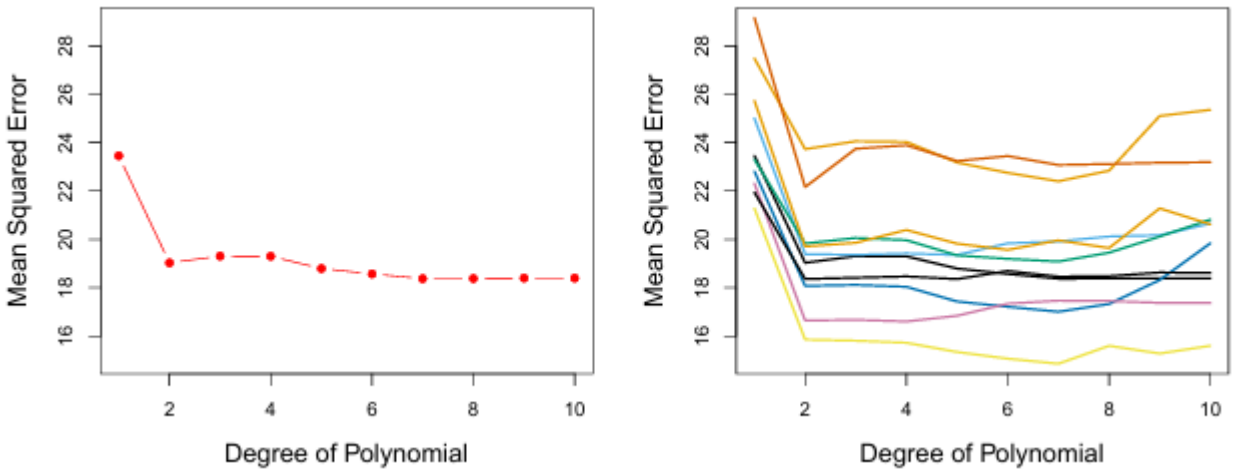
- Here we randomly divide the available set of samples into two parts: a **training set** and a **validation** or **hold-out set**.
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate in the case of a qualitative (discrete) response.

**Validation Process:**
A random splitting into two halves : left part is training set and right part is validation set.




Now using the example of **automobile data**:

- Want to compare linear vs higher-order polynomial term in a linear regression.
- We randomly split the 392 observations into two sets , a training set containing 196 of the data points ,and a validation set containing the remaining 196 observations.




- Left: Validation error estimates for a single split into training and validation data sets.
- Right: The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set.

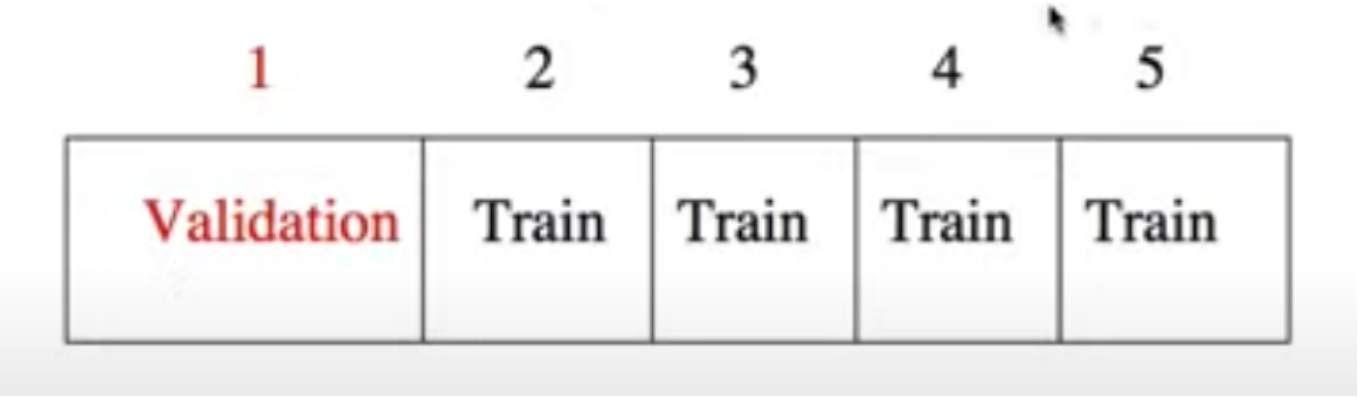
This illustrates the variability in the estimated test MSE that results from this approach. even though the shape remains more or less the same.

Some of the drawbacks of validation-set approach are:

- validation estimate of the test error can be highly variable
- in the validation approach , only a subset of observations are used to fit the model.

# K-fold Cross Validation

- Widely used approach for estimating test error.
- Estimates can be used to select best model, and to give an idea of the test error of the final chosen model.
- Idea is to randomly divide the data into $K$ equal-sized parts. We leave out part $k$ , fit the model to the $K-1$ parts(combined), and then obtain the prediction for the left-out $k$ th part.
- This is done in turn for each part $k=1,2,\ldots,K$ , and then the results are combined.



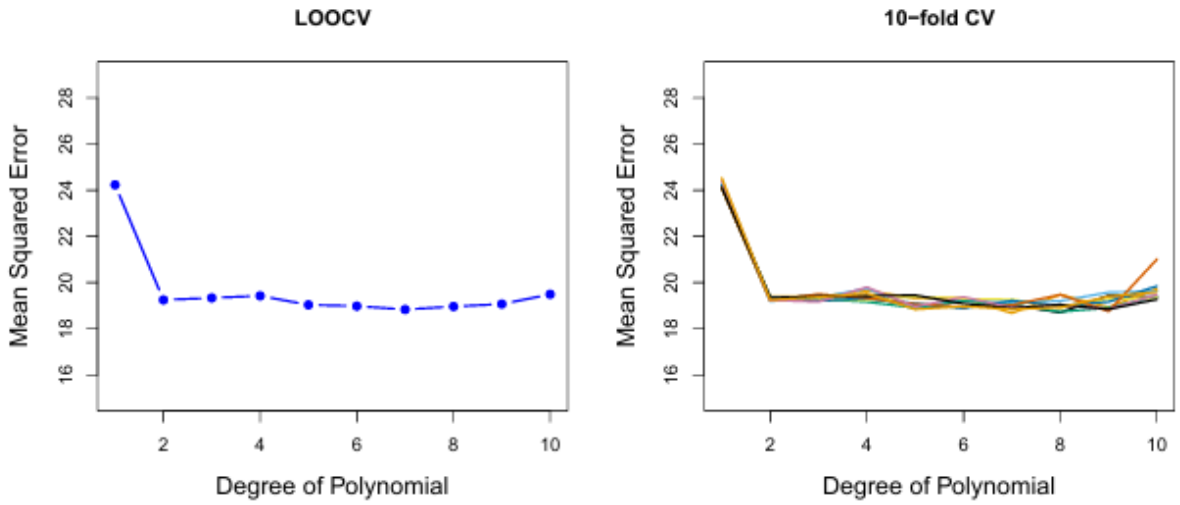

- Let the $K$ parts be $C_1, C_2, \ldots, C_k$ , where $C_k$ denotes the indices of the observations in part $k$ . There are $n_k$ observations in part $k$ : if $n$ is a multiple of $K$ , then $n_k = n/K$ .
- Compute

$$CV_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k$$

  where $MSE_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ , and $\hat{y}_i$ is the fit for observation $i$ , obtained from the data with part $k$ removed.
- Setting $K = n$ yields $n$-fold or **leave-one out cross validation** (LOOCV).
- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single fit ! . The following formula holds:

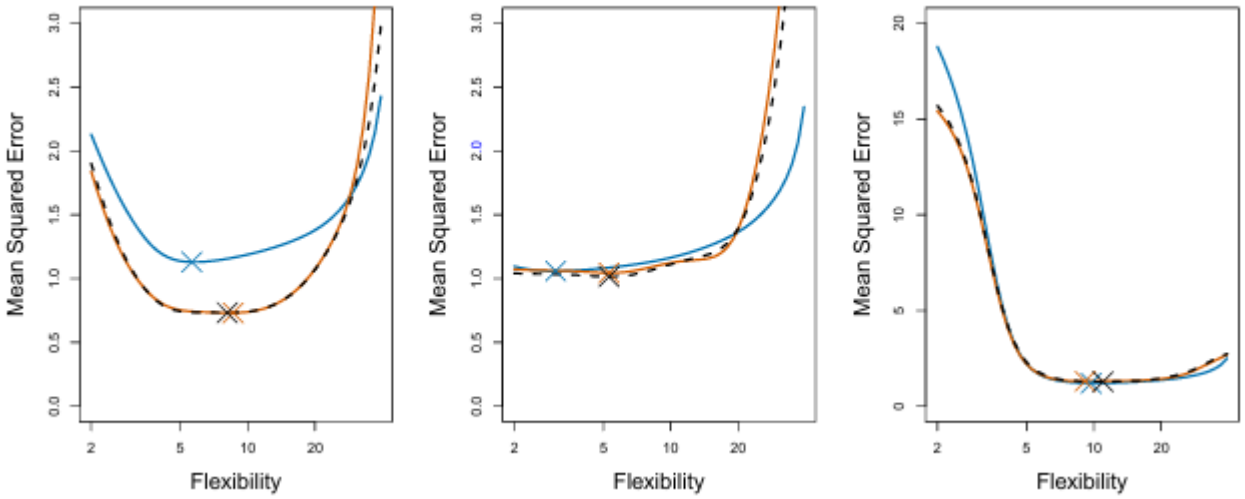
$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} (\frac{y_i - \hat{y}_i}{1 - h_i})^2,$$

  where $\hat{y}_i$ is the $i$th fitted value from the original least squares fit, and $h_i$ is the leverage (diagonal of the hat matrix). This is like the ordinary MSE, except the $i$th residual is divided $(1 - h_i)$ .
- LOOCV sometimes useful, but typically doesn't **shake up** the data enough. The estimates from each fold are highly correlated and hence their average can have high variance.
- a better choice is $K = 5$ or $K = 10$ . Why? Because LOOCV is trying to estimate the error rate with almost same number of observations.




Cross-validation was used on the Auto data set in order to estimate the test error that results from predicting mpg using polynomial functions of horsepower.
Left: The LOOCV error curve.
Right: 10-fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.

True and estimated test MSE for the simulated data sets in the figure. The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.




- When we perform cross-validation, our goal might be to determine how well a given statistical learning procedure can be expected to perform on independent data; in this case, the actual estimate of the test MSE is of interest.
- But at other times we are interested only in the location of the minimum point in the estimated test MSE curve. This is because we might be performing cross-validation on a

number of statistical learning methods, or on a single method using different levels of flexibility, in order to identify the method that results in the lowest test error.

- For this purpose, the location of the minimum point in the estimated test MSE curve is important, but the actual value of the estimated test MSE is not.
- We find in the figure that despite the fact that they sometimes underestimate the true test MSE, all of the CV curves come close to identifying the correct level of flexibility—that is, the flexibility level corresponding to the smallest test MSE

**Other issues with Cross-Validation**

- Since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upwards.
- This bias is minimized when $K = n(LOOCV)$ , but this estimate has high variance , as noted earlier.
- $K = 5$ or $K = 10$ provides a good compromise for this bias-variance tradeoff.

# Cross-Validation for Classification Problems

- It is computed as:

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{n} Err_k$$

  where $Err_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$
- The estimated standard deviation of $CV_K$ is

$$\hat{SE}(CV_K) = \sqrt{\sum_{k=1}^{K} (Err_k - \bar{Err}_k)^2/(K-1)}$$

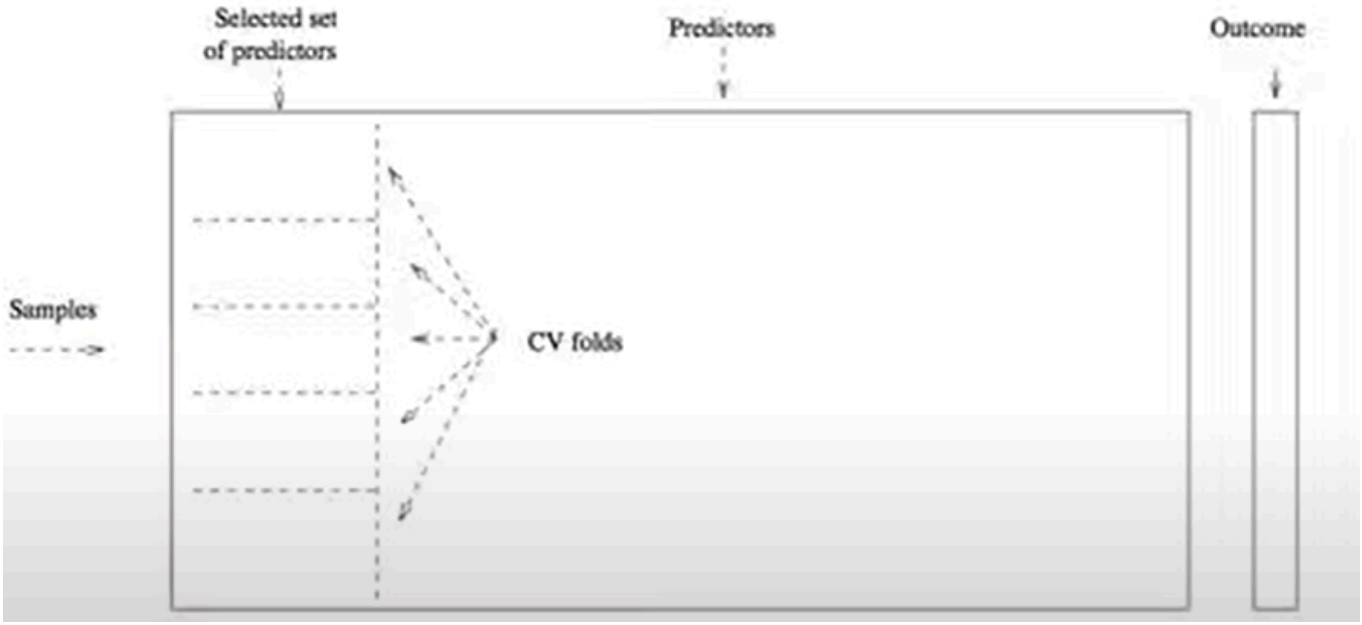
- This is useful estimate, strictly speaking not quite valid.

# Cross-Validation: right and wrong

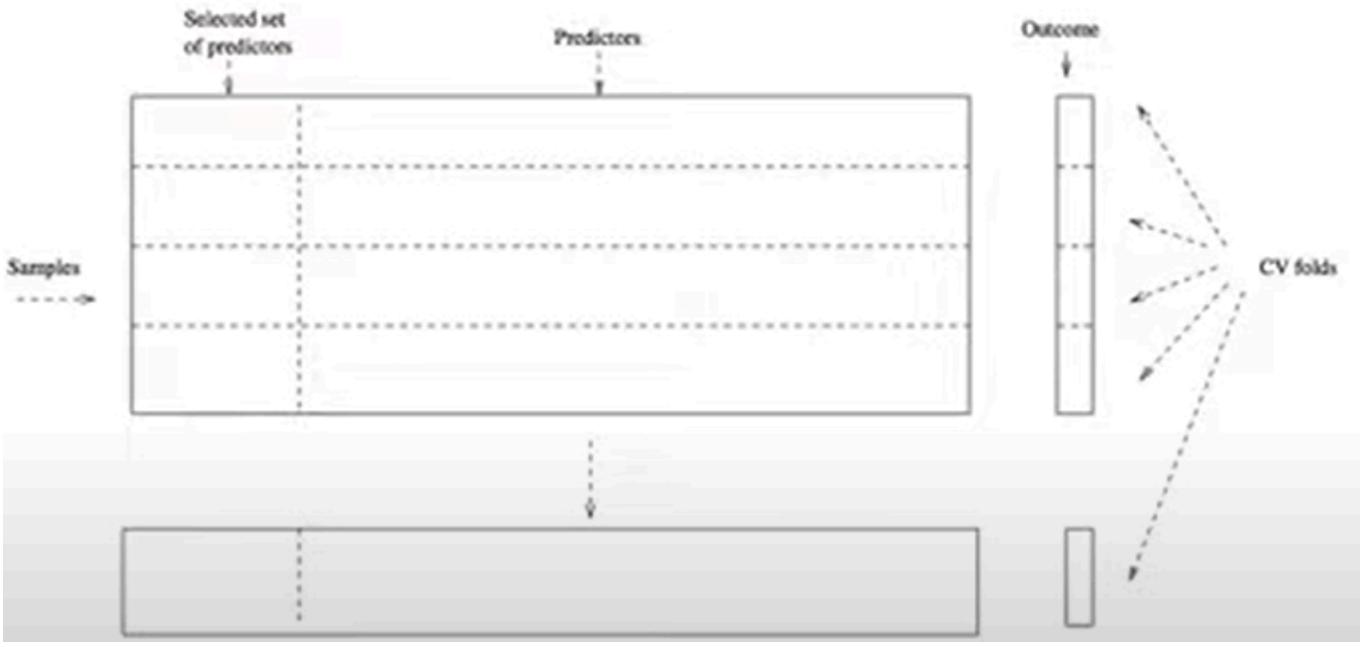
- Consider a simple classifier applied to some two-class data:
  1. Starting with 5000 predictors and 50 samples , find the 100 predictors having the largest correlation with the class labels.
  2. We then apply a classifier such as logistic regression ,using only these 100 predictors.
     How do we estimate the test performance of this classifier?
     Can we apply cross-validation in step 2, forgetting about step 1?

**NO !**

- This would ignore the fact that in step 1, the procedure has already seen the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.
- It is easy to simulate realistic data with the class labels independent of the outcome , so that the true test error = 50%, but the CV error estimate that ignores Step 1 is zero! .So it's a serious bias.
- We have seen this problem in some high profile genomic papers.
- Wrong: Apply cross-validation in step 2.




- Right: Apply cross-validation to steps 1 and 2. You can see that in each iteration you might filter out different set of predictors.




---

# The Bootstrap

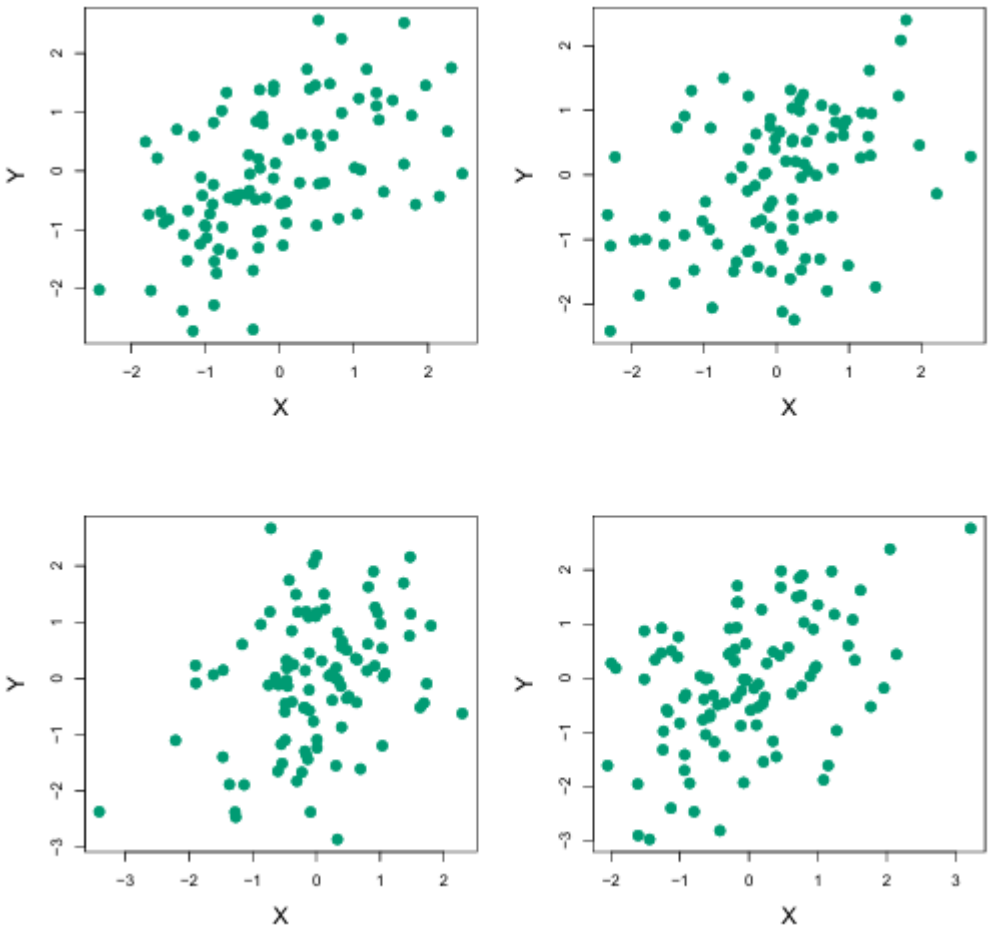
- It is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning methods.
- For example, it can provide an estimate of the standard error of a coefficient , or a confidence interval for that coefficient.
- Lets look at this simple example. Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of $X$ and $Y$, respectively, where $X$ and $Y$ are random quantities.
- We will invest a fraction $\alpha$ of our money in $X$, and will invest the remaining $1-\alpha$ in $Y$.
- We wish to choose $\alpha$ to minimize the total risk, or variance, of our investment. In other words, we want to minimize $\mathrm{Var}(\alpha X + (1-\alpha)Y)$.
- One can show that the value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

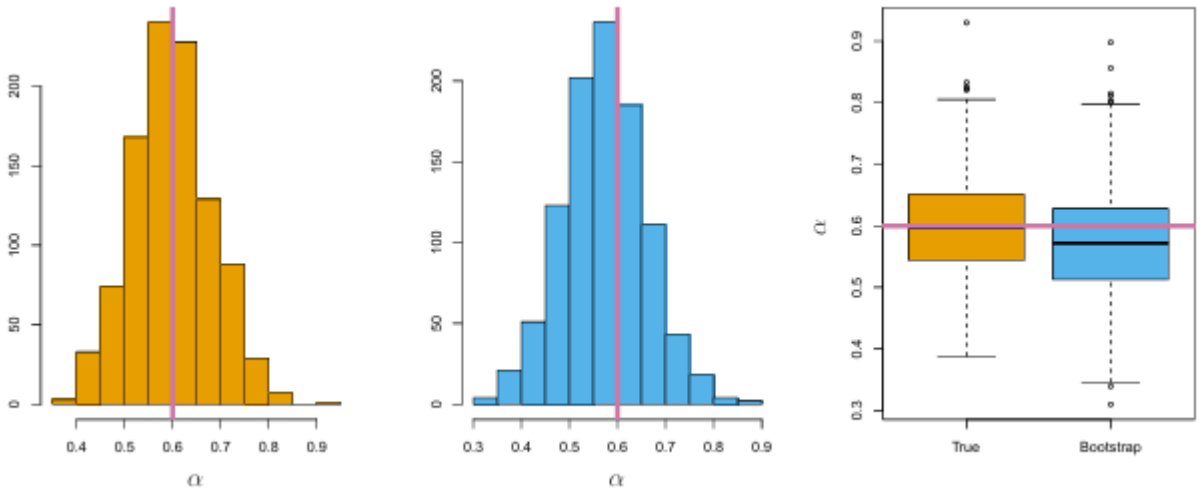
, where $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

- But the values of $\sigma_X^2$, $\sigma_Y^2$, and $\sigma_{XY}$ are unknown.
- We can compute estimates for these quantities, $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$, and $\hat{\sigma}_{XY}$, using a data set that contains measurements for $X$ and $Y$.
- We can then estimate the value of $\alpha$ that minimizes the variance of our investment using

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$




- Each panel displays 100 simulated returns for investments $X$ and $Y$. From left to right and top to bottom, the resulting estimates for $\alpha$ are 0.576, 0.532, 0.657, and 0.651.
- To estimate the standard deviation of $\hat{\alpha}$, we repeated the process of simulating 100 paired observations of $X$ and $Y$, and estimating $\alpha$ 1,000 times.
- We thereby obtained 1,000 estimates for $\alpha$, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.
- The left-hand panel of the figure displays a histogram of the resulting estimates.
- For these simulations the parameters were set to $\sigma_X^2 = 1, \sigma_Y^2 = 1.25$, and $\sigma_{XY} = 0.5$, and so we know that the true value of $\alpha$ is 0.6 (indicated by the red line).




- The mean over all 1,000 estimates for α is

$$\bar{\alpha} = \frac{1}{1000}\sum_{r=1}^{1000}\hat{\alpha}_r = 0.5996$$

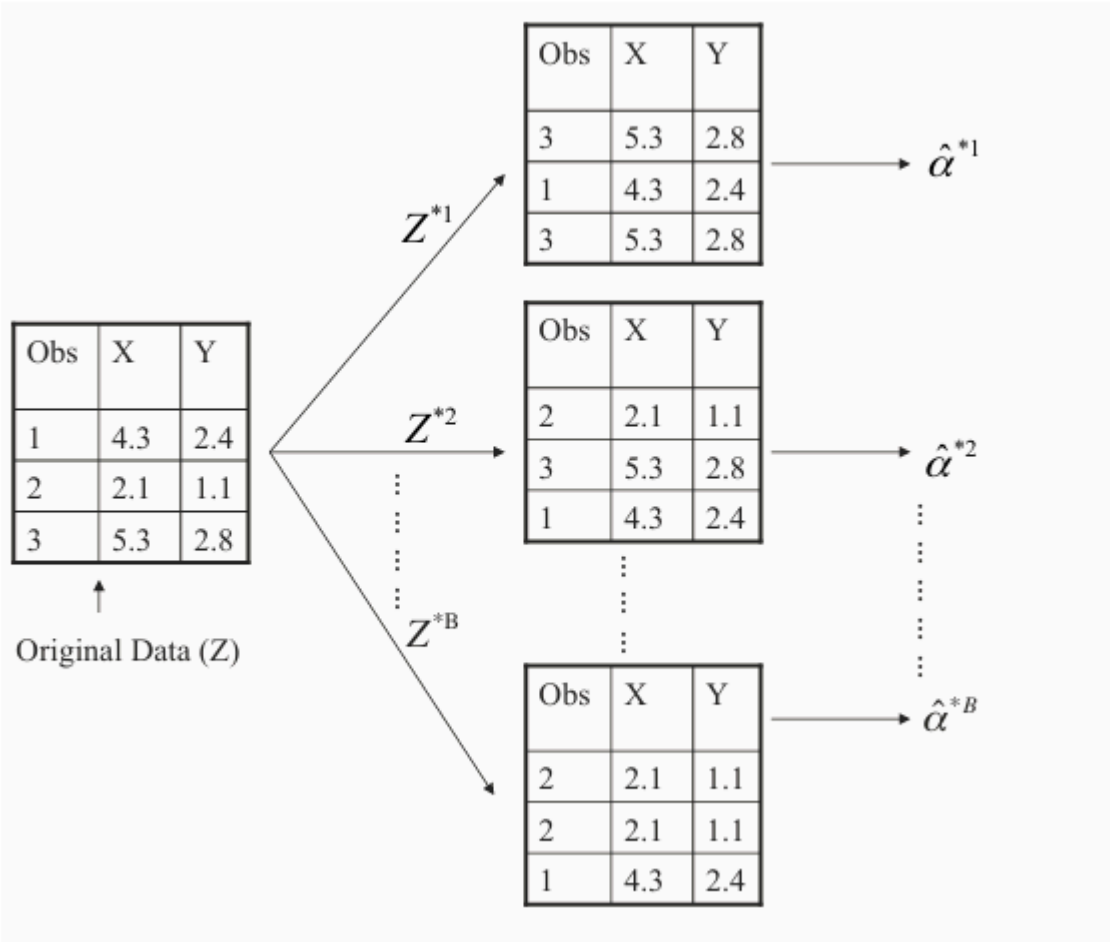
, very close to α = 0.6, and the standard deviation of the estimates is

$$\sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000}(\hat{\alpha}_r - \bar{\alpha})^2} = 0.083$$

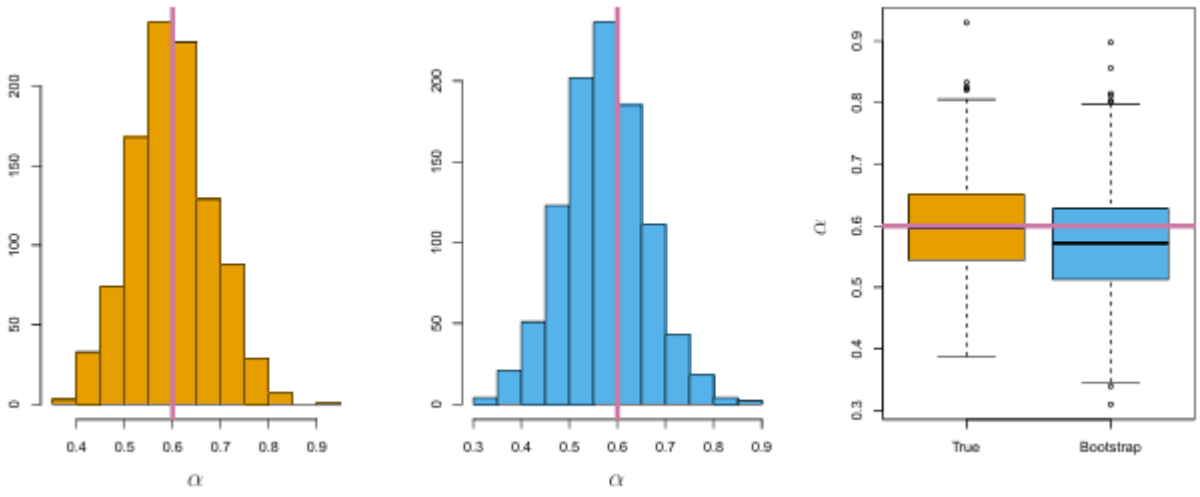
- This gives us a very good idea of the accuracy of $\hat{\alpha}$: SE($\hat{\alpha}$) ≈ 0.083.
- So roughly speaking, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08, on average.

# Now look back at the real world

- The procedure described above can't be applied because for real data we can't generate new samples from the original population.
- However , bootstrap approach allows us to use a computer to mimic the process of obtaining new datasets, so that we can estimate the variability of our estimate without generating additional samples.
- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set with **replacement** .
- Each of these "bootstrap datasets" is created by sampling **with replacement** , and is the **same size** as our original dataset. As a result some observations may appear more than once in a given bootstrap data set and some not at all.




Above is a graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations . Each bootstrap data set contains $n$ observations ,sampled with replacement from the original dataset. Each bootstrap data set is used to obtain an estimate of $\alpha$ .



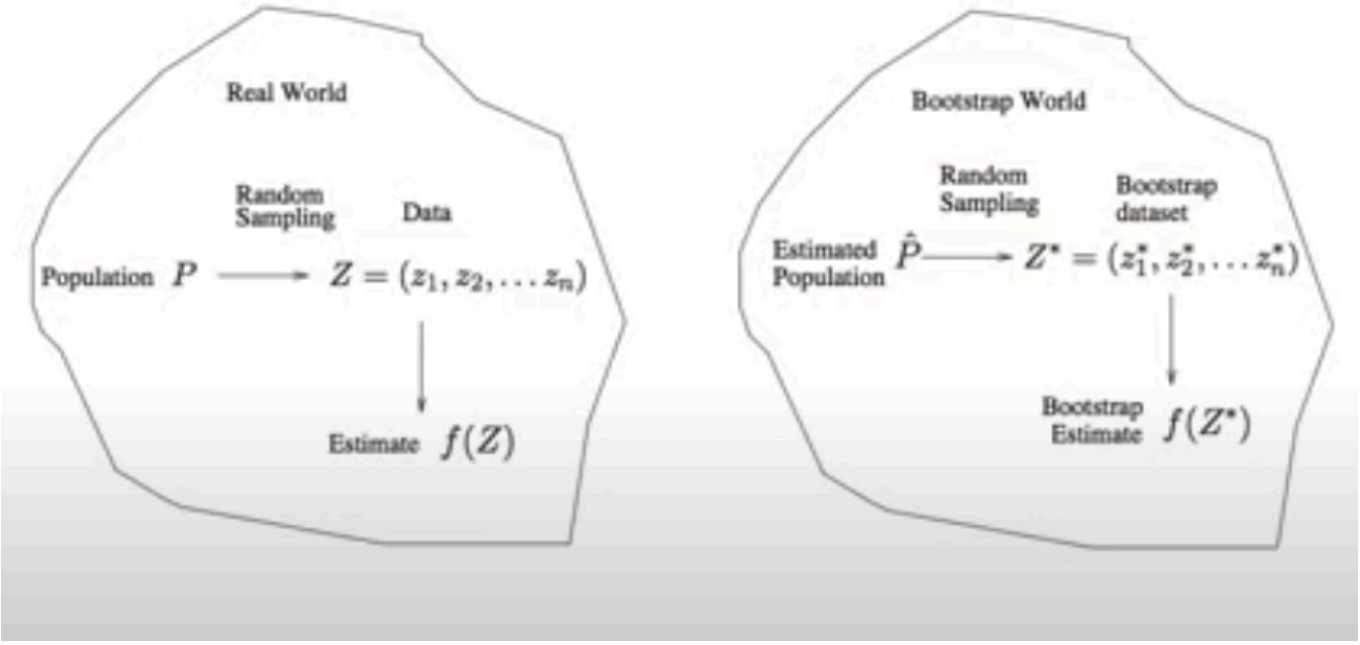

In the center there's a histogram of the estimates of obtained from 1,000 bootstrap samples from a single data set. On the right we have the estimates of displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of $\alpha$ .

- Denoting the first bootstrap data set by $Z^{*1}$, we use $Z^{*1}$ to produce a new bootstrap estimate for $\alpha$, which we call $\hat{\alpha}^{*1}$.
- This procedure is repeated $B$ times for some large value of $B$ (say 100 or 1000), in order to produce $B$ different bootstrap data sets, $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$, and $B$ corresponding $\alpha$ estimates, $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$.
- We estimate the standard error of these bootstrap estimates using the formula

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{*r} - \bar{\hat{\alpha}^*})^2}$$

- This serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set. See center and right panels of Figure . Bootstrap results are in blue. For this example $SE_B(\hat{\alpha}) = 0.087$.

# A general picture of the bootstrap




- In more complex situations, figuring out the appropriate way to generate bootstrap samples can require some thought.
- For example, if the data is time series, we can't simply sample the observations with replacement. Why? because for bootstrap when we were sampling from the population, we assume they were identically independent data. Not the case in timeseries. Solution: Block Bootstrap.

# Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For eg, if you look at the histogram in the middle in above figure, the 5% and 95% quantiles of the 1000 values is (.43,.72)
- This represents an approximate 90% confidence interval for true $\alpha$ . **How do we interpret this confidence interval?** If we were to repeat this experiment from the population many times, the confidence interval will contain the true value of $\alpha$ 90% of the time.
- The above interval is called a **Bootstrap Percentile** confidence interval. It is the simplest method( among many approaches) for obtaining a confidence interval from the bootstrap.

# Can the bootstrap estimate the prediction error?

- In cross-validation , each of the $K$ validation folds is distinct from the other $K-1$ folds used for training: **there is no overlap** . This is crucial for it success.
- To estimate prediction error using bootstrap, we could think about using each bootstrap dataset as our training sample and the original sample as our validation sample.
- But each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample.
- This will cause bootstrap to seriously underestimate the true prediction error.
- The other way around - with original sample = training sample, bootstrap dataset = validation sample is worse!

# Removing the overlap

- Can partly fix the problem by only using predictions for those observations that did not occur in the current bootstrap sample.
- But the methods get complicated, and in the end cross-validation provides a simpler, more attractive approach for estimating prediction error.