We can repeat this procedure for increasingly complex polynomial fits.

```
> loocv_error <- rep(0, 10)
> for (i in 1:10) {
+          glm_fit <- glm(mpg ~ poly(horsepower, i, raw = TRUE),
                data = Auto)
+          loocv_error[i] <- cv.glm(Auto, glm_fit)$delta[1]
+         }
> loocv_error
[1] 24.23151 19.24821 19.33498 19.42443 19.03321 18.97864 18.83305
[8] 18.96115 19.06863 19.49093
```

we see a sharp drop in the estimated test MSE between the linear and the quadratic fits, but then no clear improvement from using higher-order polynomials.

The cv.glm() function can also be used to implement k-fold CV. Below we use $k = 10$, a common choice for k, on the Auto data set.

```
> set.seed(5)
> cv_error_10 <- rep(0, 10)
> for (i in 1:10) {
+         glm_fit <- glm(mpg ~ poly(horsepower, i, raw = TRUE),
                data = Auto)
+         cv_error_10[i] <- cv.glm(Auto, glm_fit, K=10)$delta[1]
+ }
> cv_error_10
[1] 24.14736 19.38899 19.26022 19.57029 19.10580 18.66693 18.57784
[8] 18.70916 19.26092 19.64803
```

we still see little evidence that using cubic or higher order polynomial terms leads to lower test error than simply using a quadratic term.

Cross-Validation on Classification Problems

- In the classification setting, cross-validation works just as described earlier, except that rather than using MSE to quantify test error, we instead use the number of misclassified observations.

- We divide the data into k roughly equal-sized parts $C_1, C_2, ..., C_k$. C_i denotes the indices of the observations in part i. There are n_i observations in part i. Then, the k-fold CV error rate takes the form:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \text{Err}_i$$

where $\text{Err}_i = \frac{1}{n_i}\sum_{j\in C_i} I(y_j \neq \hat{y}_j)$.

The Bootstrap

- The *bootstrap* is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.

- As a simple example, it can be used to estimate the standard errors of the coefficients from a linear regression fit.

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 α of our money in X, and will invest the remaining $1-\alpha$ in Y. Since there is variability associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance, of our investment. In other words, we want to minimize $\text{Var}(\alpha X + (1-\alpha)Y)$.

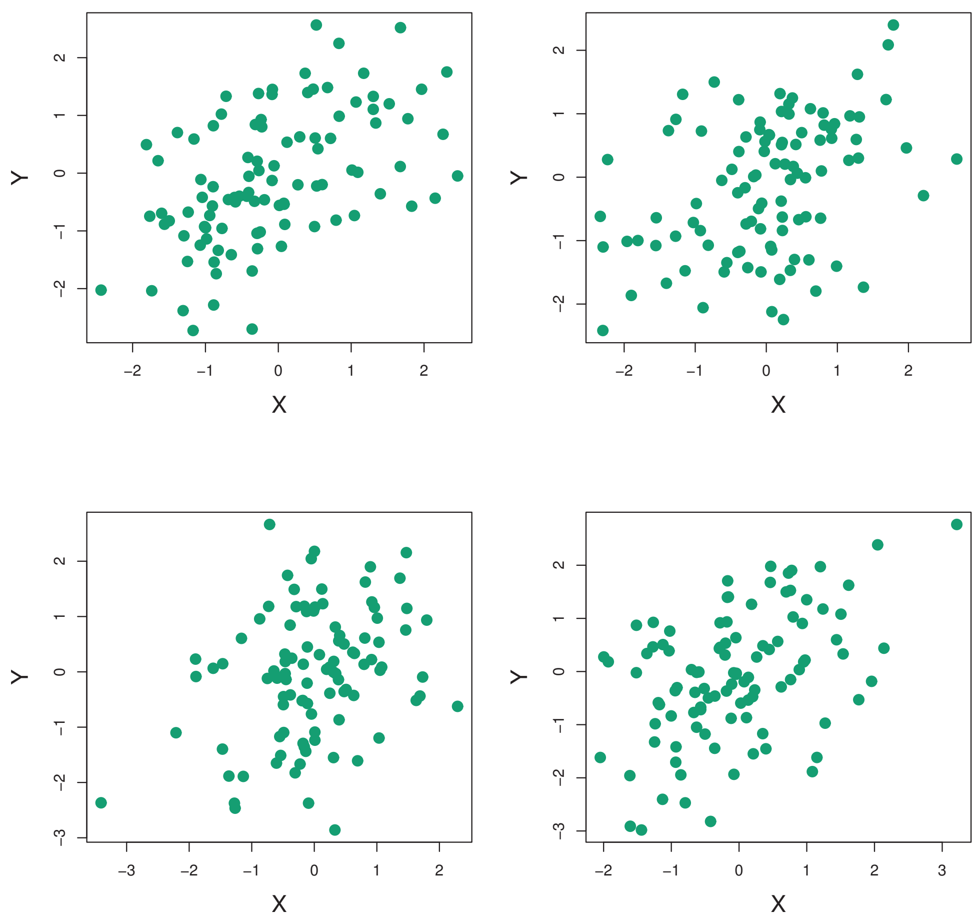
- One can show that the value that minimizes the risk is given by

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

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

- In reality, the values of σ_X^2, σ_Y^2 , and σ_{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 past measurements for X and Y.
- We can then estimate the value of α 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 α 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 α, 1,000 times.

- We thereby obtained 1,000 estimates for α, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.

- 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 α is 0.6.

- 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 $\alpha = 0.6$, 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 $\hat{\alpha}$: $\mathrm{SE}(\hat{\alpha}) \approx 0.083$.

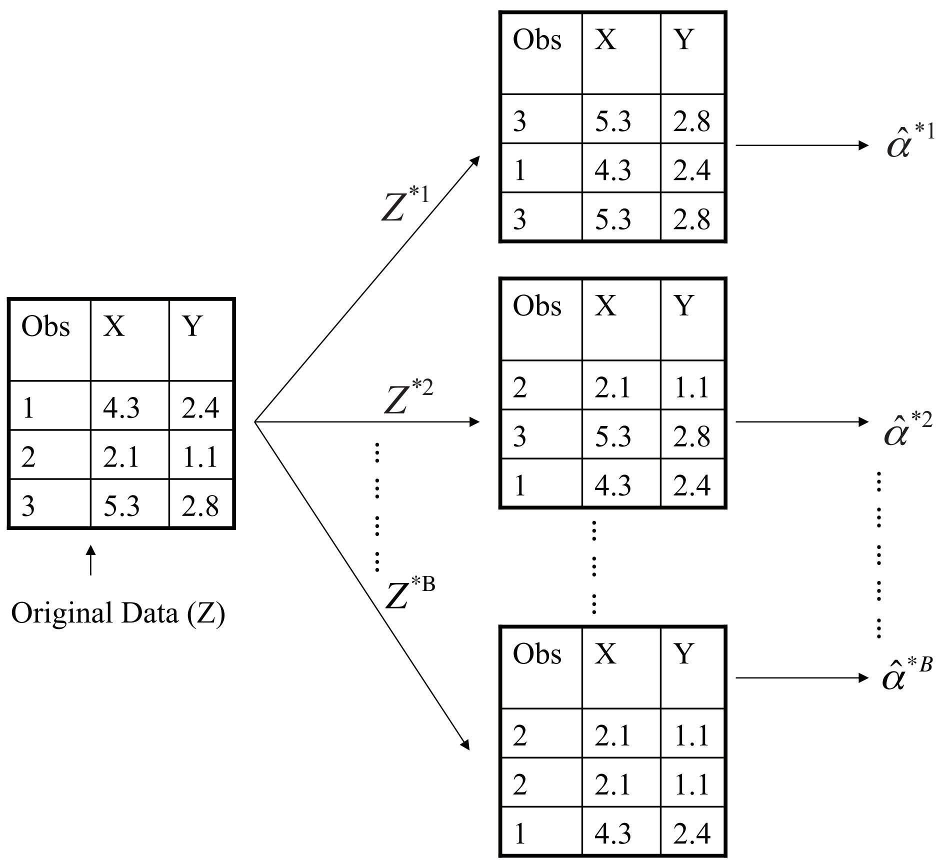
This means that, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08, on average.

- The procedure outlined above cannot be applied, because for real data we cannot generate new samples from the original population.

- However, the bootstrap approach allows us to use a computer to mimic the process of obtaining new data sets, 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.

- This approach is illustrated on a simple data set, which we call Z, that contains only $n = 3$ observations.



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 data set. Each bootstrap data set is used to obtain an estimate of α.

- Each of these "bootstrap data sets" 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.

- Denoting the first bootstrap data set by Z^{*1}, we use Z^{*1} to produce a new bootstrap estimate for α, 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 α estimates, $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$.

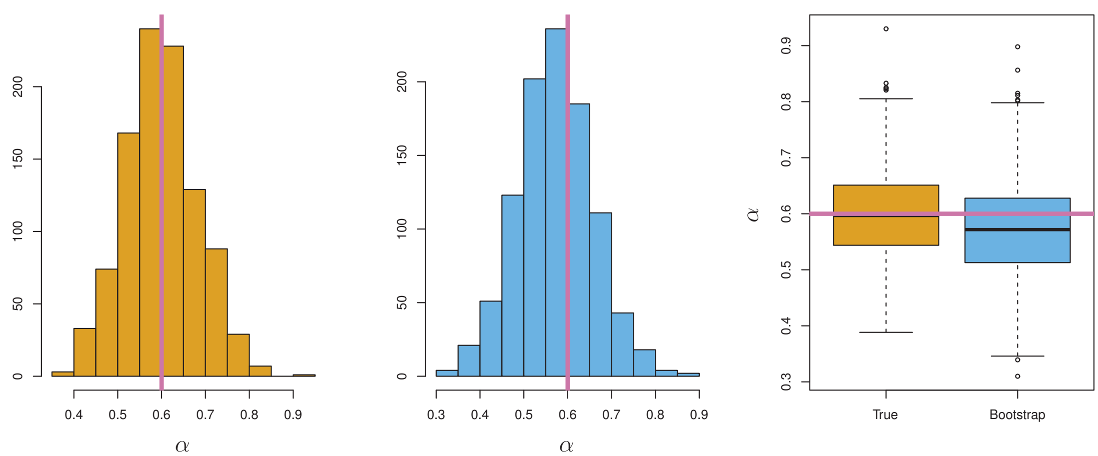
- We estimate the standard error of these bootstrap estimates using the formula

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

where $\bar{\hat{\alpha}}^{*} = \frac{1}{B}\sum_{r=1}^{B}\hat{\alpha}^{*r}$.

$SE_B(\hat{\alpha})$ serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set.

$SE_B(\hat{\alpha}) = 0.087$ when $B = 1000$



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.

Performing a bootstrap analysis in R entails only two steps.

- First, we must create a function that computes the statistic of interest.
- Second, we use the boot() function, which is part of the boot library, to perform the bootstrap by repeatedly sampling observations from the data set with replacement.

Now we use R to perform the bootstrap on our previous example.

The Portfolio data set in the ISLR2 package is simulated data of 100 pairs of returns, generated in the fashion described earlier.

We first create a function, alpha_fn(), which takes as input the (X, Y) data as well as a vector indicating which observations should be used to estimate α. The function then outputs the estimate for α based on the selected observations.

```
> alpha_fn <- function(data, index) {
+         X <- data$X[index]
+         Y <- data$Y[index]
+          (var(Y) - cov(X, Y)) / (var(X) + var(Y) - 2 * cov(X, Y))
+ }
```

This function returns, or outputs, an estimate for α based on applying $\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$ to the observations indexed by the argument index. For instance, the following command tells R to estimate α using all a bootstrap data set.

```
> library(ISLR2)
> set.seed(1)
> alpha_fn(Portfolio, sample(100, 100, replace = T))
[1] 0.5963833
```

Now we produce $R = 1,000$ bootstrap estimates for α using the boot() function.

```
> library(boot)
> boot(Portfolio, alpha_fn, R = 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Portfolio, statistic = alpha_fn, R = 1000)


Bootstrap Statistics :
     original        bias    std. error
t1* 0.5758321 -7.315422e-05  0.08861826
```

$\hat{\alpha}$

$SE_B(\hat{\alpha})$ when $B = 1000$