At the core of the old efficient market hypothesis was the claim that you could not forecast stock returns save for the higher average returns of high-beta portfolios--not of individual stocks, not of stock indices, not from past price patterns, not from indicators of fundamental value. This came crashing down when Robert Shiller led a decades-long campaign to demonstrate that at the level of the stock market as a whole returns are, indeed, predictable.

Let us begin with the Shiller stock index database. The preliminaries:

```{r}
# Setup variables...

bootreps <- 10000   # Number of replications for the bootstrap

# Read in the Shiller monthly stock market
# database from the web and assign it to
# the dataframe "Shiller":

`20140824_Shiller_Data` <-

Shiller <- `20140824_Shiller_Data`

# Define the "lead" function to easily
# construct lead and lag variables:

lead<-function(x,shift_by){
    stopifnot(is.numeric(shift_by))
    stopifnot(is.numeric(x))
    if (length(shift_by)>1)
        return(sapply(shift_by,shift, x=x))
    out<-NULL
    abs_shift_by=abs(shift_by)
    if (shift_by > 0 )
        out<-c(tail(x,-abs_shift_by),rep(NA,abs_shift_by))
    else if (shift_by < 0 )
        out<-c(rep(NA,abs_shift_by), head(x,-abs_shift_by))
    else
        out<-x
```
# Pull variables out of the dataframe:

```r
out
```

```r
} # Date
```

```r
REAL_PRICE <- Shiller$REAL_PRICE[121:1724]  # Real stock index price--Cowles and then S&P Composite
```

```r
REAL_DIVIDENDS <- Shiller$REAL_DIVIDENDS[121:1724]  # Real dividends on the index at an annualized rate
```

```r
REAL_EARNINGS <- Shiller$REAL_EARNINGS[121:1724]  # Real earnings on the index at an annualized rate
```

```r
MA10_EARNINGS <- Shiller$MA.10._OF_EARNINGS[121:1724]  # The Campbell-Shiller trailing ten-year moving average of earnings--what Shiller calls "cyclically adjusted" earnings, but they aren't really
```

```r
CUMULATIVE_RETURN <- Shiller$CUMULATIVE_RETURN[121:1724]  # Cumulative returns on a reinvested portfolio since 1871
```

```r
CAPE <- REAL_PRICE/MA10_EARNINGS  # The Campbell-Shiller "cyclically adjusted price-earnings ratio"
```

```r
YIELD <- REAL_EARNINGS/REAL_PRICE  # The simple earnings yield
```

```r
CAPEYIELD <- 1/CAPE  # The attached permanent earnings yield
```

```r
```

# Calculate monthly and annual forward returns:

```r
LEAD1RETURN <- (lead(CUMULATIVE_RETURN,12)/CUMULATIVE_RETURN)-1  # The forward one-year return on a reinvested portfolio
```

```r
LEAD1MORETURN <- (lead(CUMULATIVE_RETURN,1)/CUMULATIVE_RETURN)^12-1  # The forward one-month return, at an annual rate
```

# See if things are what they should be:

```r
plot(DATE,CAPEYIELD,type='l',main="CAPE Yield vs. Date")
```

```r
```

## Return Predictability ##

Now let's start investigating return predictability. The question is whether--at the level of an index of the stock market as a whole--the old efficient market hypothesis is right or wrong. And we investigate this by seeing whether we can forecast stock market returns knowing nothing but the ratio of the stock market's value to the "permanent earnings" of the company that make up the market. We start by running the simplest possible regression: how does the CAPEYIELD--the ratio of the permanent earnings of the companies making up Shiller's stock index portfolio do at predicting whether the next month's stock returns will be high or low?

### The Simplest One-Month Return-Predictability Regression: ###

```r
cape_regression_1mo <- lm(LEAD1MORETURN ~ CAPEYIELD)
```

```
```
The computer tells us—with 1602 observations—that there is a significant relationship here: the higher the permanent earnings yield, that is, the higher the ten-year moving average of lagged earnings is as a proportion of the current price, the higher return. The one-tailed significance of the t-value of 3.74 is 0.00009: the computer seems to be telling us that if we really did live in a world in which there was no relationship between the current CAPE yield and next month’s returns, we have had a very unlikely history: only 9 times in 100,000, only 0.009% of the time, would this regression have been this statistically significant.

However, we want to check. And the natural way to check is to run the bootstrap on our data:

```{r}
boot1modata <- data.frame(DATE, CAPEYIELD, LEAD1MORETURN)
d <- 1:1723
library(boot)

delongbootstrap <- function(boot1modata, d) {
  yd <- as.vector(boot1modata$LEAD1MORETURN)
  xd <- as.vector(boot1modata$CAPEYIELD)
  y <- yd[d]
  x <- xd[d]
  regressionresults.lm <- lm(formula = y ~ x)
  ceoffs <- regressionresults.lm$coefficients
  slope <- ceoffs[2]
  return(slope)
}

slope2 <- delongbootstrap(boot1modata, d)
b = boot(boot1modata, delongbootstrap, R=bootreps)
```

####The mean of the estimated slope:
```{r echo=FALSE}
print(mean(b$t[,1]))
```
####
####The standard deviation of the estimated slope:
```{r echo=FALSE}
print(sd(b$t[,1]))
```
####

The bootstrap standard error of the slope is more than three times as large as what the central limit theorem-trusting computer spat out...

Let’s look at the distribution of the bootstrap replication estimates:
```
```{r}
hist(b$t[,1], main="Bootstrap Distribution of the Slope", breaks="fd")
```

Something has gone horribly wrong. The distribution of bootstrap slope estimates looks nothing at all like a bell curve. It looks, instead, like a normal distribution with a mean of 3 or so and a standard deviation of 1 or so plus ten times a Poisson—a distribution that has the greatest chance of being zero, a lesser chance of being 10, a lesser chance of being 20, a still lesser chance of being 30, and so forth.

This is a sign that there is an influential observation in the data, inclusion of which in the regression kicks the estimated slope up by 10. And what each bootstrap slope estimate is depends on whether we drew this influential observation 0, 1, 2, 3, 4, or however times in each bootstrap replication sample.

Let's look for this influential observation:
```
```{r}
plot(DATE,LEAD1MORETURN,main="Monthly Rates of Return vs. Time")
plot(CAPEYIELD,LEAD1MORETURN,main="Monthly Rates of Return vs. Permanent Earnings Yield")
sd(LEAD1MORETURN[1:1602])
```

Yes: there is an astonishing outlier: From July-August 1932, with a permanent earnings yield of 17%, we get a 51% jump in the real value of the stock market and thus a 52.4% monthly return. Annualizing that gives us a 15611% annual rate of return for that month, 40 times the standard deviation of the return *even including it in the standard deviation*—and that is driving the results. This is an overwhelming Black Swan: I had to torment Wolfram Alpha for five minutes before it would tell me that the cumulative normal distribution at that value is $2 \times 10^{-349}$ rather than simply zero.

Perhaps if we expressed our monthly returns not in exponentiated-by-12 terms but as monthly returns things would look more reasonable?
```
```{r}
LEAD1MORETURN <- (lead(CUMULATIVE_RETURN,1)/CUMULATIVE_RETURN)-1
cape_regression_2mo <- lm(LEAD1MORETURN ~ CAPEYIELD)
summary(cape_regression_2mo)
```

It is still an influential observation—but with a t-statistic of only 11 rather than 40 a less influential one. What if we dummy it out?
```
```{r}
LEAD1MORETURN[620] <- NA
cape_regression_3mo <- lm(LEAD1MORETURN ~ CAPEYIELD)
summary(cape_regression_3mo)
We get a slope coefficient of 0.074890 with a reported standard deviation of 0.034483, compared to a slope coefficient of 0.077849 with a standard deviation of 0.034515 with the month included: important but not dominant.

And the bootstrap produces a result distribution that looks bell-curved:

```r
LEAD1MORETURN[620] <- 0.51
boot1modata <- data.frame(DATE,CAPEYIELD,LEAD1MORETURN)
d <- 1:1723
library(boot)
delongbootstrap <- function(boot1modata, d) {
yd <- as.vector(boot1modata$LEAD1MORETURN)
xd <- as.vector(boot1modata$CAPEYIELD)
y <- yd[d]
x <- xd[d]
regressionresults.lm <- lm(formula = y ~ x)
ceoffs <- regressionresults.lm$coefficients
slope <- ceoffs[2]
return(slope)
}
slope2 <- delongbootstrap(boot1modata, d)
b = boot(boot1modata, delongbootstrap, R=bootreps)
print(mean(b$t[,1]))   # the mean slope
print(sd(b$t[,1]))   # the standard error of the slope
hist(b$t[,1], main="Bootstrap Distribution of the Slope", breaks="fd")
```

Our right-hand side variable—the CAPEYIELD—is a slowly-moving one. Perhaps we are introducing noise and error by sampling it too often in some sense? After all, the information in the data is the conditional mean of the return given the permanent-earnings yield. Suppose we sample the data each January?

###The Simplest One-Year Return-Predictability Regression:###

```r
yearscreen <- seq(from=1,to=1603, by=12)
DATEY <- DATE[yearscreen]
CAPEYIELDY <- CAPEYIELD[yearscreen]
YIELDY <- YIELD[yearscreen]
LEAD1RETURNY <- LEAD1RETURN[yearscreen]

cape_regression_1yr <- lm(LEAD1RETURNY ~ CAPEYIELDY)
summary(cape_regression_1yr)
plot(CAPEYIELDY,LEAD1RETURNY,main="1-Year Forward Returns vs. CAPE Yield")
```
There is no loss in estimated significance at all. The annual slope coefficient is somewhat larger: $1.22 > 0.075 \times 12 = 0.90$.

```r
boot1yrdata <- data.frame(DATEY, CAPEYIELDY, LEAD1RETURNY)
d <- 1:134

delongbootstrap <- function(boot1yrdata, d) {
  yd <- as.vector(boot1yrdata$LEAD1RETURNY)
  xd <- as.vector(boot1yrdata$CAPEYIELDY)
  y <- yd[d]
  x <- xd[d]
  regressionresults.lm <- lm(formula = y ~ x)
  ceoffs <- regressionresults.lm$coefficients
  slope <- ceoffs[2]
  return(slope)
}
slope2 <- delongbootstrap(boot1yrdata, d)
b = boot(boot1yrdata, delongbootstrap, R=bootreps)
print(mean(b$t[,1]))   # the mean slope
print(sd(b$t[,1]))   # the standard error of the slope
hist(b$t[,1], main="Bootstrap Distribution of the Slope", breaks="fd")
```

And the bootstrap estimates are well-behaved and consonant with the OLS ones...

Note that the key to Shiller's result is the sharpening of the right-hand side variable by taking the *permanent* earnings yield—*the ratio of *permanent* earnings to the price: the raw market index earnings yield produces nothing but mush:

```r
cape_regression_2yr <- lm(LEAD1RETURNY ~ YIELDY)
summary(cape_regression_2yr)
plot(YIELDY, LEAD1RETURNY, main="1-Year Forward Returns vs. Earnings Yield")
abline(lm(LEAD1RETURNY ~ YIELDY), col="red")
```