Stock-Market Return Predictability: An Unwise Data Transformation, and Saved by the Bootstrap

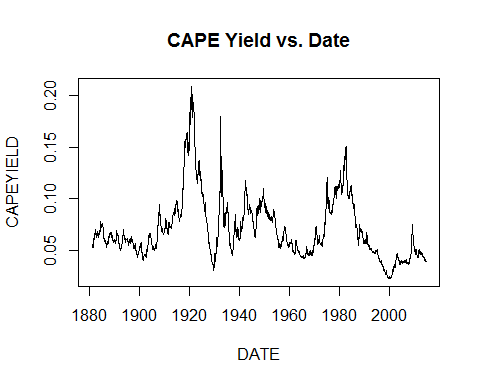
J. Bradford DeLong

#### September 7, 2014 :: UC Berkeley

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-β 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:

# 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":  
Shiller <- read.csv("../../Data/20140824\_Shiller\_Data.csv", header  
 = TRUE, sep=",", na.strings="NA", dec=".")  
  
# 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  
out  
}  
# Pull variables out of the dataframe:  
DATE <- Shiller$DATE[121:1724] # Date  
REAL\_PRICE <- Shiller$REAL\_PRICE[121:1724]   
# Real stock index price-- Cowles and then S&P Composite  
REAL\_DIVIDENDS <- Shiller$REAL\_DIVIDENDS[121:1724]   
# Real dividends on the index at an annualized rate  
REAL\_EARNINGS <- Shiller$REAL\_EARNINGS[121:1724]   
# Real earnings on the index at an annualized rate  
MA10\_EARNINGS <- Shiller$MA.10.\_OF\_EARNINGS[121:1724]   
# The Campbell-Shiller trailing ten-year moving average of earnings  
#--what Shiller calls "cyclically adjusted" earnings, but they aren't really  
CUMULATIVE\_RETURN <- Shiller$CUMULATIVE\_RETURN[121:1724]   
# Cumulative returns on a reinvested portfolio since 1871  
CAPE <- REAL\_PRICE/MA10\_EARNINGS   
# The Campbell-Shiller "cyclically adjusted price-earnings ratio"  
YIELD <- REAL\_EARNINGS/REAL\_PRICE # The simple earnings yield  
CAPEYIELD <- 1/CAPE # The attached permanent earnings yield  
# Calculate monthly and annual forward returns:  
LEAD1RETURN <- (lead(CUMULATIVE\_RETURN,12)/CUMULATIVE\_RETURN)-1   
# The forward one-year return on a reinvested portfoloi  
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:  
plot(DATE,CAPEYIELD,type='l',main="CAPE Yield vs. Date")

 ##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:###

cape\_regression\_1mo <- lm(LEAD1MORETURN ~ CAPEYIELD)  
summary(cape\_regression\_1mo)

##   
## Call:  
## lm(formula = LEAD1MORETURN ~ CAPEYIELD)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.35 -0.47 -0.11 0.23 154.91   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.581 0.251 -2.31 0.02075 \*   
## CAPEYIELD 12.216 3.266 3.74 0.00019 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.96 on 1601 degrees of freedom  
## (1 observation deleted due to missingness)  
## Multiple R-squared: 0.00867, Adjusted R-squared: 0.00805   
## F-statistic: 14 on 1 and 1601 DF, p-value: 0.00019

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 computere 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:

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:

## [1] 12.22

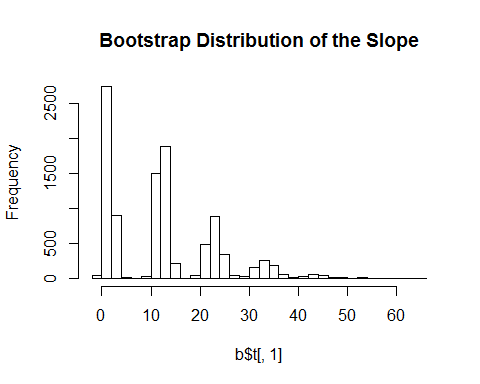
#### The standard deviation of the estimated slope:

## [1] 10.72

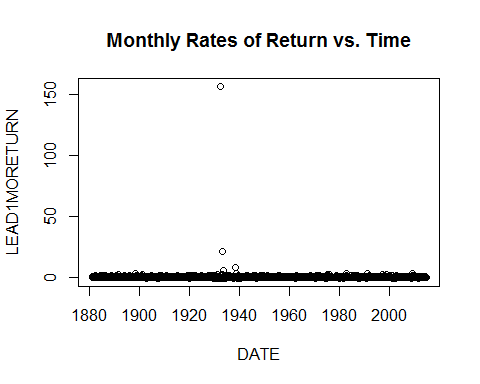
The bootstrap standard error of the slope is more than three times as large as what the central limit theorem-trusting computer spat out... Let's look at the distribution of the bootstrap replication estimates:

#### The Bootstrap Histogram

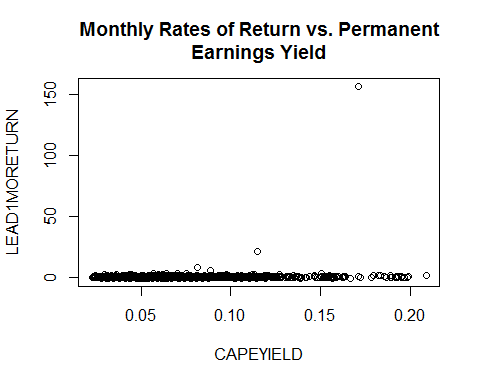
hist(b$t[,1], main="Bootstrap Distribution of the Slope", breaks="fd")

 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:

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])

## [1] 3.976

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 x 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?

LEAD1MORETURN <- (lead(CUMULATIVE\_RETURN,1)/CUMULATIVE\_RETURN)-1  
cape\_regression\_2mo <- lm(LEAD1MORETURN ~ CAPEYIELD)  
summary(cape\_regression\_2mo)

##   
## Call:  
## lm(formula = LEAD1MORETURN ~ CAPEYIELD)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.2651 -0.0197 0.0029 0.0238 0.5105   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.000517 0.002652 0.19 0.846   
## CAPEYIELD 0.077849 0.034515 2.26 0.024 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0418 on 1601 degrees of freedom  
## (1 observation deleted due to missingness)  
## Multiple R-squared: 0.00317, Adjusted R-squared: 0.00254   
## F-statistic: 5.09 on 1 and 1601 DF, p-value: 0.0242

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?

LEAD1MORETURN[620] <- NA  
cape\_regression\_3mo <- lm(LEAD1MORETURN ~ CAPEYIELD)  
summary(cape\_regression\_3mo)

##   
## Call:  
## lm(formula = LEAD1MORETURN ~ CAPEYIELD)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.2651 -0.0197 0.0029 0.0238 0.5109   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.000662 0.002649 0.25 0.80   
## CAPEYIELD 0.074890 0.034483 2.17 0.03 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0418 on 1600 degrees of freedom  
## (2 observations deleted due to missingness)  
## Multiple R-squared: 0.00294, Adjusted R-squared: 0.00232   
## F-statistic: 4.72 on 1 and 1600 DF, p-value: 0.03

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:

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

## [1] 0.0899

print(sd(b$t[,1])) # the standard error of the slope

## [1] 0.0527

hist(b$t[,1], main="Bootstrap Distribution of the Slope", breaks="fd")

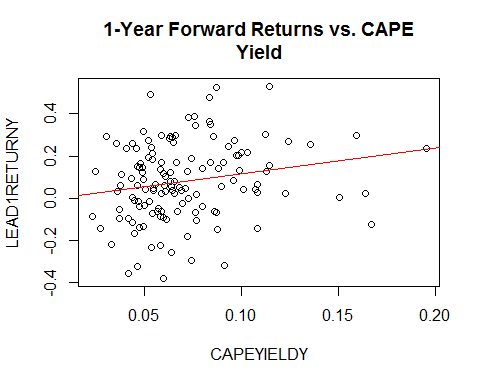
## 

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:###

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)

##   
## Call:  
## lm(formula = LEAD1RETURNY ~ CAPEYIELDY)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.4468 -0.1260 -0.0041 0.1056 0.4351   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.00656 0.04007 -0.16 0.870   
## CAPEYIELDY 1.22122 0.52347 2.33 0.021 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.18 on 131 degrees of freedom  
## (1 observation deleted due to missingness)  
## Multiple R-squared: 0.0399, Adjusted R-squared: 0.0326   
## F-statistic: 5.44 on 1 and 131 DF, p-value: 0.0212

plot(CAPEYIELDY,LEAD1RETURNY,main="1-Year Forward Returns vs. CAPE  
Yield")  
abline(lm(LEAD1RETURNY ~ CAPEYIELDY), col="red")



RETRESY <- residuals(cape\_regression\_1yr)

There is no loss in estimated significance at all. The annual slope coefficient is somewhat larger: 1.22 > .075 x 12 = 0.90.

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

## [1] 1.247

print(sd(b$t[,1])) # the standard error of the slope

## [1] 0.5206

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 righthand 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:

cape\_regression\_2yr <- lm(LEAD1RETURNY ~ YIELDY)  
summary(cape\_regression\_2yr)

##   
## Call:  
## lm(formula = LEAD1RETURNY ~ YIELDY)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.4535 -0.1274 -0.0108 0.1271 0.4579   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.0475 0.0453 1.05 0.30  
## YIELDY 0.4373 0.5787 0.76 0.45  
##   
## Residual standard error: 0.184 on 131 degrees of freedom  
## (1 observation deleted due to missingness)  
## Multiple R-squared: 0.00434, Adjusted R-squared: -0.00326   
## F-statistic: 0.571 on 1 and 131 DF, p-value: 0.451

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

