Cochrane (RFS 2008) Replication

Alexander Ober

4/19/2022

Code is posted at the following site: https://github.com/alexanderober/BUSI525/upload/main/Cochrane

Replication of Table 3 and Figure 1

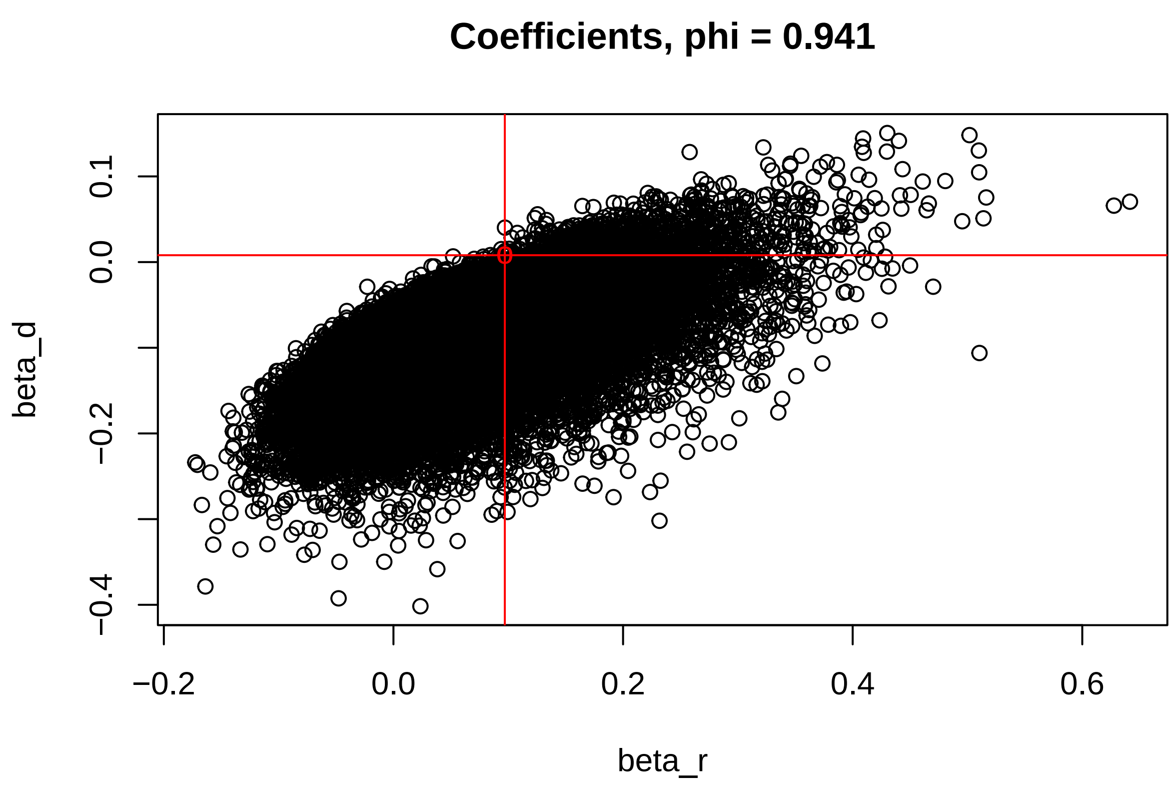
I follow the steps Cochrane provides in his paper. In order to simulate the systems, I use values reported in the tables. Presumably using these values gives some rounding errors, although these don't change the results qualitatively. In general, I use $\phi = 0.941$.

```
## inputs
rho = 0.9638
T = 2004 - 1927+1
N = 50000
sigma_d = 0.14
sigma_dp = 0.153
rho_d_dp = 0.075
sigma_d_dp = rho_d_dp*sigma_d*sigma_dp
phi = 0.941

## initialize variables
r = rep(0, T+1)
d_p = rep(0, T+1)
d_growth = rep(0, T+1)
beta_d = rep(0, N)
beta_r = rep(0, N)
phis = rep(0, N)
t_d = rep(0, N)
t_r = rep(0, N)
## matrix to hold simulations of disturbances
eps = matrix(rep(0, 2*T), T, 2)
library(MASS)

for (n in 1:N){
  ## simulate the disturbances forwards
  eps = mvrnorm(n = T,
                rep(0, 2),
                matrix(c(sigma_d^2, sigma_d_dp, sigma_d_dp, sigma_dp^2),2,2))
  ## initialize the dividend yield
  d_p[1] = mvrnorm(n = 1, 0, sigma_dp^2/(1 - phi^2))

  ## simulate the system forward
  for (t in 1:T){
    d_p[t+1] = phi*d_p[t] + eps[t, 2]
```

```
    d_growth[t+1] = (rho*phi - 1)*d_p[t] + eps[t, 1]
    r[t+1] = eps[t, 1] - rho*eps[t, 2]
  }

  ## compute and hold regression statistics
  mod1 <- lm(r[2:(T+1)]~d_p[1:T])
  mod2 <- lm(d_growth[2:(T+1)] ~ d_p[1:T])
  mod3 <- lm(d_p[2:(T+1)] ~ d_p[1:T])
  summ1 <- summary(mod1)
  summ2 <- summary(mod2)
  summ3 <- summary(mod3)
  beta_r[n] <- summ1$coefficients[2, 1]
  beta_d[n] <- summ2$coefficients[2, 1]
  phis[n] <- summ3$coefficients[2, 1]
  t_r[n] <- summ1$coefficients[2, 3]
  t_d[n] <- summ2$coefficients[2, 3]
}
```

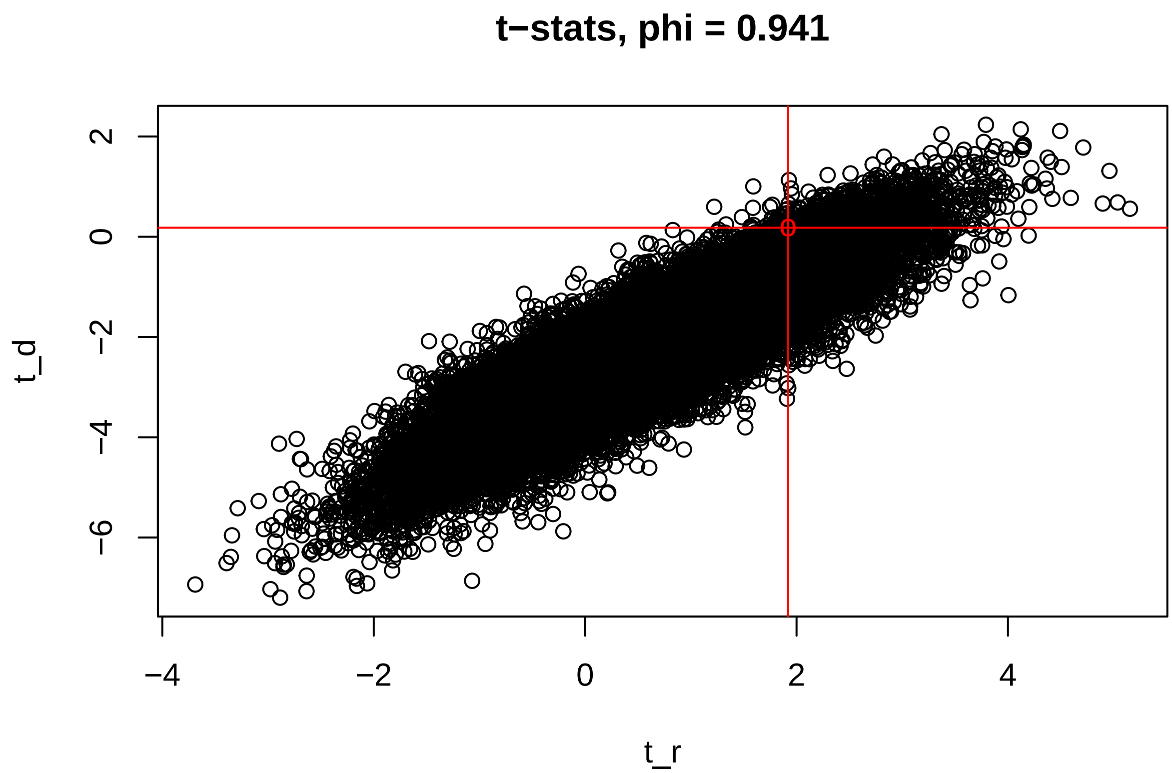
Output the scatter plots for betas and t values as in Figure 1.

```
title = paste0('Coefficients, phi = ', phi)
plot(beta_r, beta_d, main = title)
abline(v = 0.097, h = 0.008, col = 'red')
points(c(0.097), c(0.008), pch = 'o', col = 'red')
```




```
title = paste0('t-stats, phi = ', phi)
plot(t_r, t_d, main = title)
```

```r
abline(v = 1.92, h = 0.18, col = 'red')
points(c(1.92), c(0.18), pch = 'o', col = 'red')
```



Now let's output the probability values corresponding to table 3 in the paper. Note I output the values for the given input of ϕ above, corresponding to a given row of Table 3.

```r
## probability beta_r exceeds its sample value
sum(beta_r > 0.097)/50000
```

```
## [1] 0.22356
```

```r
## probability beta_d exceeds its sample value
sum(beta_d > 0.008)/50000
```

```
## [1] 0.01824
```

```r
## probability t_r exceeds its sample value
sum(t_r > 1.92)/50000
```

```
## [1] 0.0888
```

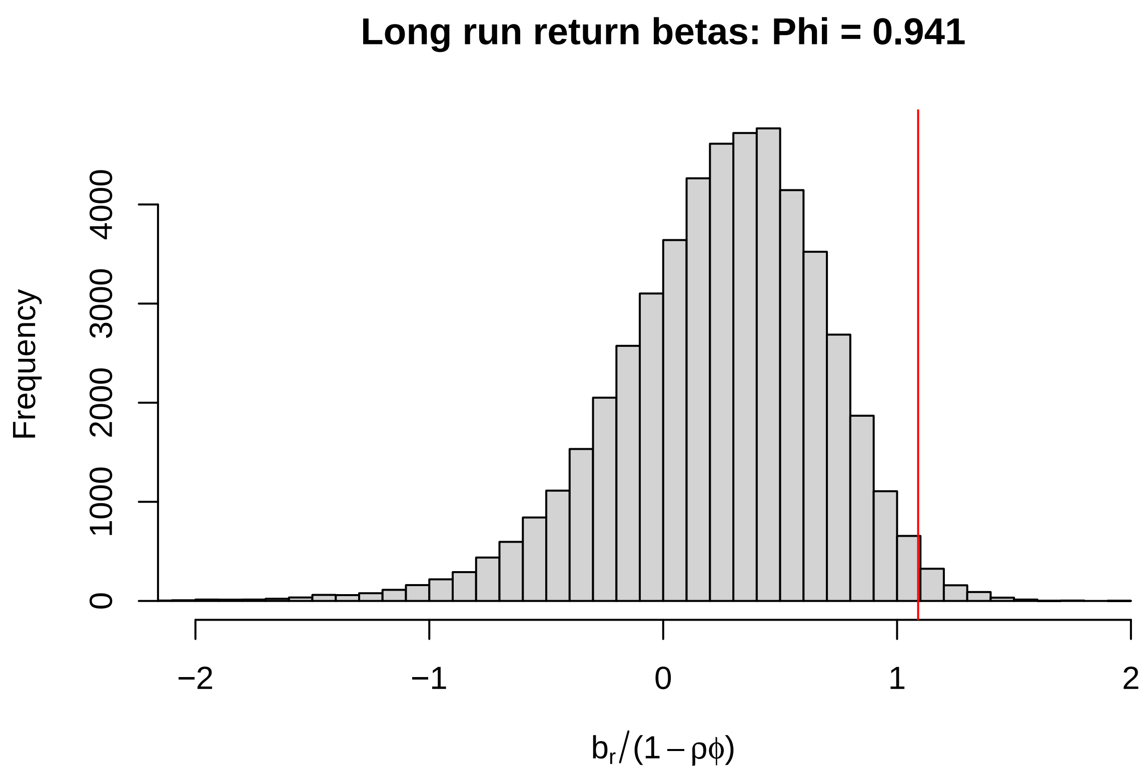
```r
## probability t_d exceeds its sample value
sum(t_d > 0.18)/50000
```

```
## [1] 0.01682
```

Replicating Figures 2 and 3

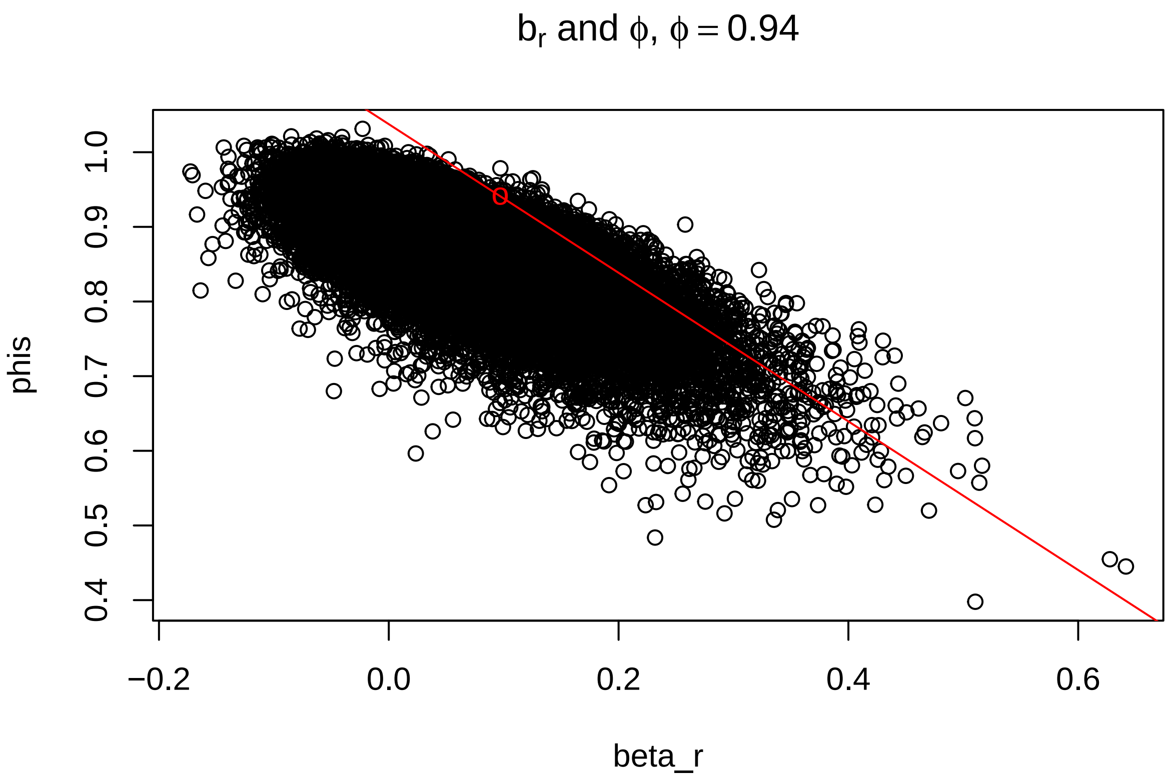
Replication of Figure 2. Note we use the outputs produced above.

```r
library(latex2exp)
title = paste0('Long run return betas: Phi = ', phi)
hist(beta_r/(1 - rho*phis), breaks = 100,
     xlim = c(-2, 2),
     xlab = TeX(r'($b_r/(1 - \rho \phi)$)'),
     main = title)
abline(v = 1.09, col = 'red')
```



Replication of Figure 3. Note we again use the outputs produced above. The red line denotes values of the long run return coefficient equal to the value in the data.

```r
title = TeX(r'($b_r$ and $\phi$, $\phi = 0.94$)')
plot(beta_r, phis, main  = title)
abline(a = 1/rho, b = -1/(0.097/(1 - rho*phi))/rho, col = 'red')
points(c(0.097), c(phi), pch = 'o', col = 'red')
```

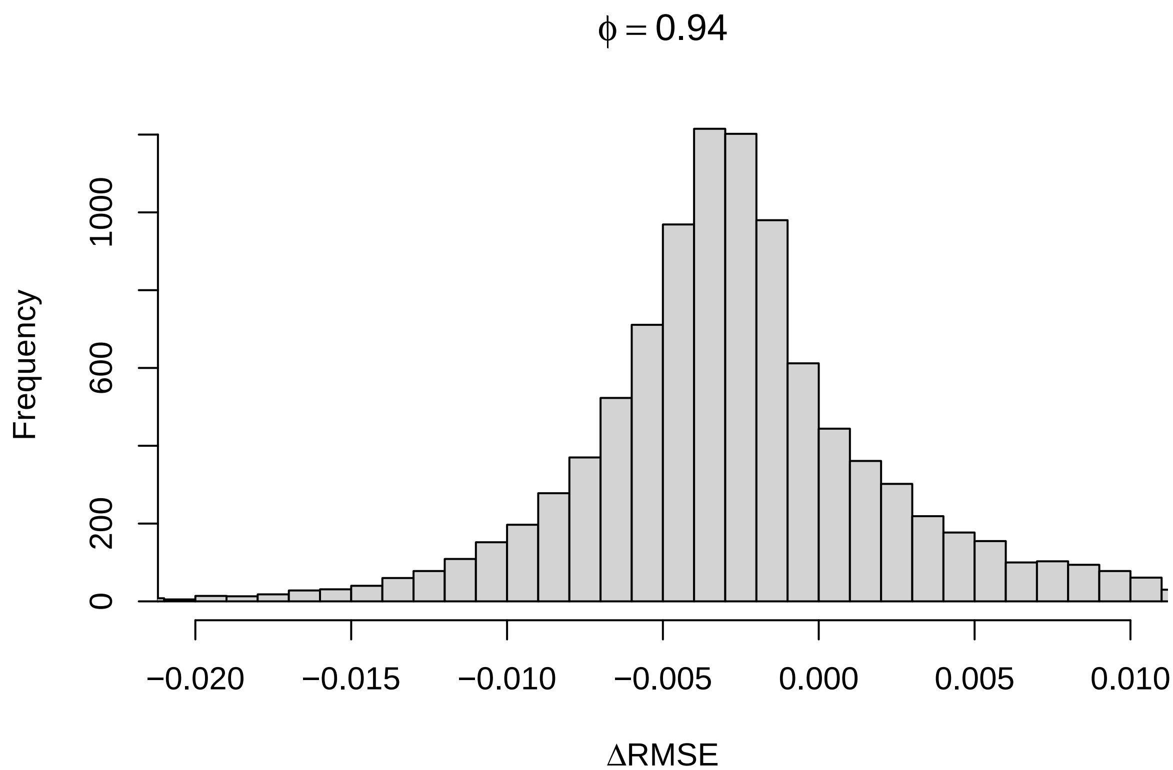


Goyal-Welch results: Figure 6 Replication

Here we replicate the Goyal-Welch statistics on a Monte Carlo simulation of simulated samples under the null of predictability. Running this is quite time-intensive, so I only consider 10,000 simulations instead of the 50,000 in the paper.

```
N = 10000
MSE_ha = 0 ## historical average MSE
MSE_pred = 0 ## predictor MSE
RMSE = rep(0, N)
for (n in 1:N){
  ## simulate the disturbances forward
  eps = mvrnorm(n = T,
                rep(0, 2),
                matrix(c(sigma_d^2, sigma_d_dp, sigma_d_dp,
                         sigma_dp^2),2,2))

  ## initialize the dividend yield
  d_p[1] = mvrnorm(n = 1, 0, sigma_dp^2/(1 - phi^2))

  ## simulate the system forward, assuming return predictability and
  ## no div growth predictability
  for (t in 1:T){
    d_p[t+1] = phi*d_p[t] + eps[t, 2]
    d_growth[t+1] = eps[t, 1]
    ## note there seems to be a typo here in the paper
```

```r
    ## there should be a negative sign in front of (rho phi - 1).
    r[t+1] = (rho*phi - 1)*d_p[t] + eps[t, 1] - rho*eps[t, 2]
  }

  ## compute rolling regressions and historical average/return
  ## regression errors
  for (t in 20:T){
    mod <- lm(r[2:t]~d_p[1:(t-1)])
    summ <- summary(mod)
    MSE_pred = (summ$coefficients[2, 1]*d_p[t] +
                  summ$coefficients[1, 1] - r[t+1])^2 +MSE_pred;
    MSE_ha = (r[t+1] - mean(r[1:t]))^2 + MSE_ha;
  }

  ## compute RMSE and reset MSEs to 0 for the next iteration
  RMSE[n] = sqrt(MSE_ha/(T-20+1)) - sqrt(MSE_pred/(T-20+1));
  MSE_pred = 0;
  MSE_ha = 0;
}

## plot histogram corresponding to Figure 6
xlab = TeX(r'($\Delta RMSE$)')
title = TeX(r'($\phi = 0.94$)')
hist(RMSE, xlim = c(-0.02, 0.01),
     xlab = xlab,
     main = title,
     breaks = 100)
```



The main point is that most of the statistics are negative here.