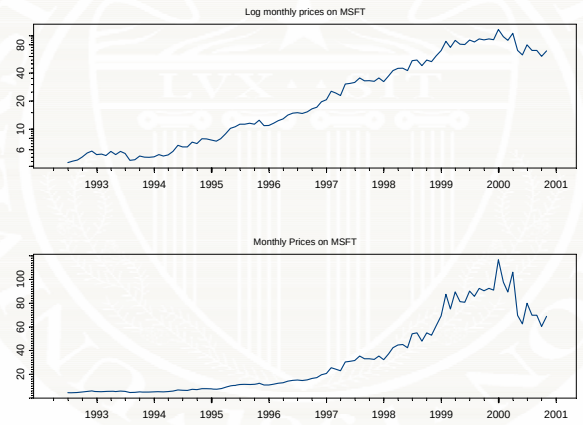
Constant Expected Return Model

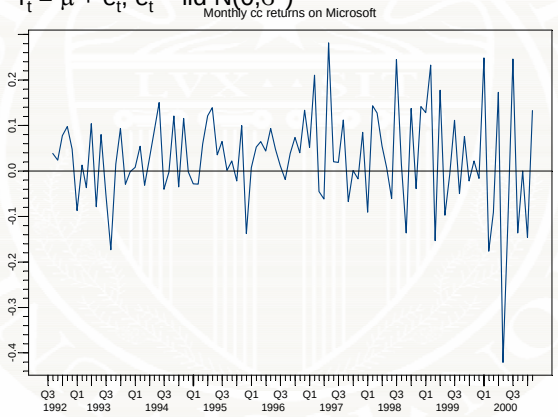
Econ 424
Eric Zivot
Fall 2006
Updated: October 24, 2006

Monthly Closing Prices of Microsoft Stock

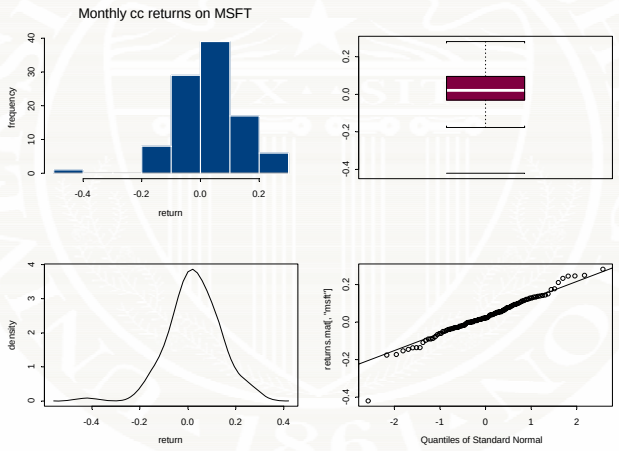


Conjecture: Monthly CC Returns follow CER Model

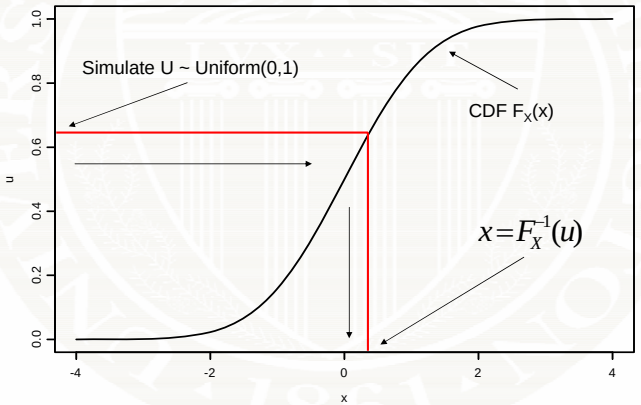
$r_t = \mu + e_t$, $e_t \sim$ iid $N(0,\sigma^2)$



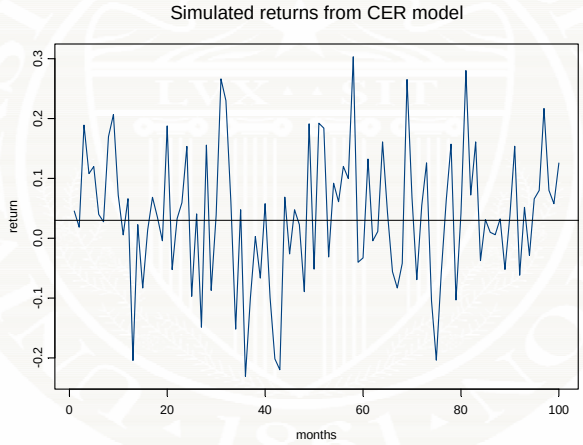
Distribution Summary for Monthly CC Returns on MSFT



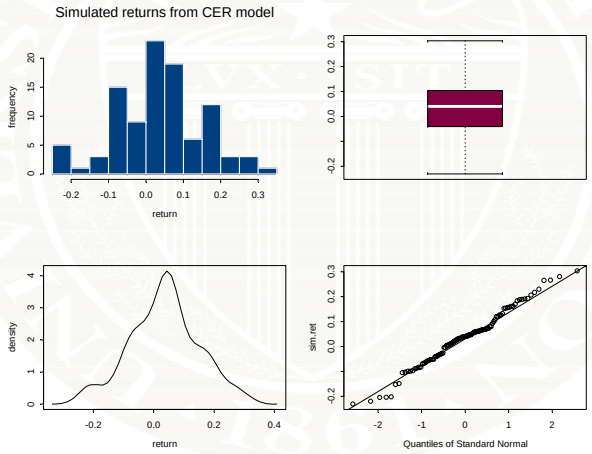
Simulating Random Data for X with CDF F_X



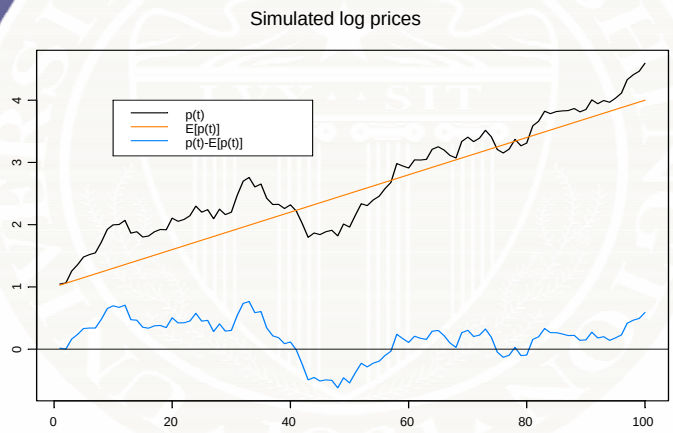
Simulated Returns from CER Model with Same Mean and SD as Microsoft



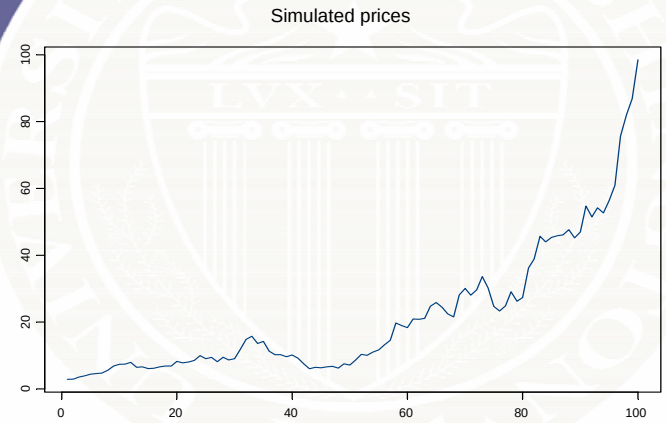
Distribution Summary for Simulated Data from CER Model



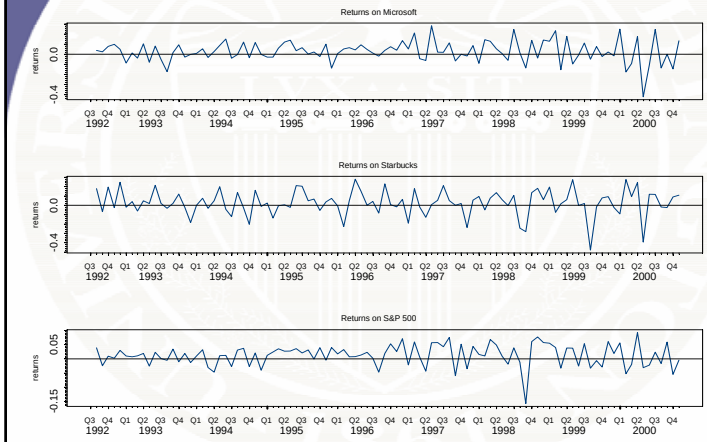
Simulated Data from Random Walk Model for Log Prices



Simulated Prices from Random Walk Model



Monthly Returns on MSFT, SBUX and S&P 500



CER Model Estimates

```
> muhat.vals
    sbux    msft   sp500
 0.02777 0.02756 0.01253

> sigma2hat.vals = colVars(returns.mat)
> sigma2hat.vals
    sbux    msft    sp500
 0.01846 0.01141 0.001432

> sigmahat.vals = colStdevs(returns.mat)
> sigmahat.vals
   sbux   msft   sp500
 0.1359 0.1068 0.03785
```

CER Model Estimates

```
> cov.mat = var(returns.ts)
> cor.mat = cor(returns.ts)

> covhat.vals = cov.mat[lower.tri(cov.mat)]
> rhohat.vals = cor.mat[lower.tri(cor.mat)]
> names(covhat.vals) = names(rhohat.vals) =
+ c("sbux,msft","sbux,sp500","msft,sp500")

> covhat.vals
 sbux,msft sbux,sp500 msft,sp500
 0.00403   0.00215    0.00224

> rhohat.vals
 sbux,msft sbux,sp500 msft,sp500
   0.2777  0.4197     0.5551
```

Estimated Standard Errors

```
> se.muhat = sigmahat.vals/sqrt(nobs)
> rbind(muhat.vals,se.muhat)
              sbux     msft     sp500
muhat.vals    0.0277   0.0275   0.01253
  se.muhat    0.0135   0.0106   0.00378

> se.sigma2hat = sigma2hat.vals/sqrt(nobs/2)
> rbind(sigma2hat.vals,se.sigma2hat)
                   sbux      msft      sp500
sigma2hat.vals     0.01845   0.01141   0.00143
  se.sigma2hat     0.00261   0.00161   0.00020

> se.sigmahat = sigmahat.vals/sqrt(2*nobs)
> rbind(sigmahat.vals,se.sigmahat)
                 sbux     msft     sp500
sigmahat.vals    0.1358   0.1068   0.0378
  se.sigmahat    0.0096   0.0075   0.0026
```

Estimated Standard Errors

```
> se.rhohat = (1-rhohat.vals^2)/sqrt(nobs)
> rbind(rhohat.vals,se.rhohat)
            sbux,msft sbux,sp500 msft,sp500
rhohat.vals 0.2777    0.4197     0.5551
  se.rhohat 0.0922    0.0823     0.0691
```

Stylized Facts for the Estimation of CER Model Parameters

- The mean is not estimated very precisely
  - Large standard errors relative to size of mean estimates
- Standard deviations and correlations are estimated more precisely than the mean

95% Confidence Intervals

```
> mu.lower = muhat.vals - 2*se.muhat
> mu.upper = muhat.vals + 2*se.muhat
> mu.width = mu.upper - mu.lower
> cbind(mu.lower,mu.upper,mu.width)
       mu.lower mu.upper mu.width
 sbux    0.0006   0.0549   0.0543
 msft    0.0061   0.0489   0.0427
sp500    0.0049   0.0201   0.0151

> sigma.lower = sigmahat.vals - 2*se.sigmahat
> sigma.upper = sigmahat.vals + 2*se.sigmahat
> sigma.width = sigma.upper - sigma.lower
> cbind(sigma.lower,sigma.upper,sigma.width)
      sigma.lower sigma.upper sigma.width
 sbux      0.1166      0.1550      0.0384
 msft      0.0917      0.1219      0.0302
sp500      0.0324      0.0431      0.0107
```

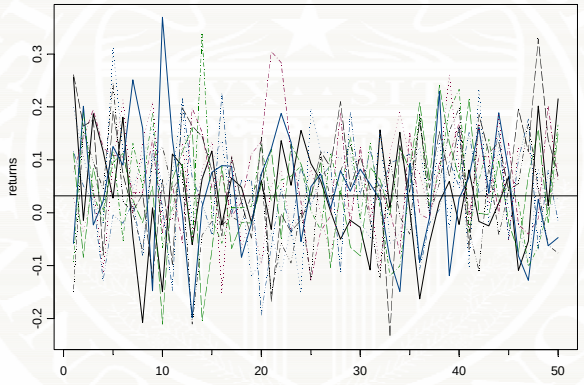
95% Confidence Intervals

```
> rho.lower = rhohat.vals - 2*se.rhohat
> rho.upper = rhohat.vals + 2*se.rhohat
> rho.width = rho.upper - rho.lower
> cbind(rho.lower,rho.upper,rho.width)
             rho.lower rho.upper rho.width
 sbux,msft      0.0931    0.4622     0.3691
sbux,sp500      0.2549    0.5845     0.3295
msft,sp500      0.4167    0.6934     0.2767
```

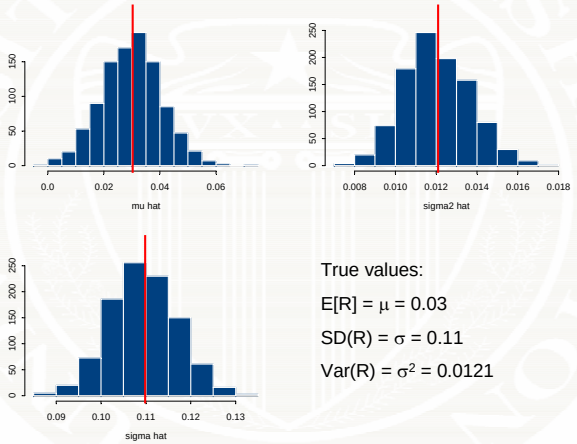
Monte Carlo Simulation Loop

```
mu = 0.03
sd = 0.11
n.obs = 100
n.sim = 1000
set.seed(111)
sim.means = rep(0,n.sim)      # initialize vectors
sim.vars = rep(0,n.sim)
sim.sds = rep(0,n.sim)
for (sim in 1:n.sim) {
      sim.ret = rnorm(n.obs,mean=mu,sd=sd)
      sim.means[sim] = mean(sim.ret)
      sim.vars[sim] = var(sim.ret)
      sim.sds[sim] = sqrt(sim.vars[sim])
}
```

10 simulated samples from CER model



Histograms of 1000 Monte Carlo Estimates



True values:

E[R] = μ = 0.03

SD(R) = σ = 0.11

Var(R) = σ^2 = 0.0121

Monte Carlo Evaluation of Bias

```
> mean(sim.means)       # true mean = 0.03
[1] 0.0298
> mean(sim.means) – mu  # estimate of bias
[1] -0.0001

> mean(sim.vars)        # true variance = 0.012
[1] 0.0120
> mean(sim.vars) - sd^2 # estimate of bias
[1] 0.0020

> mean(sim.sds)         # true SD = 0.11
[1] 0.1093
> mean(sim.sds) – sd    # estimate of bias
[1] 0.0093
```

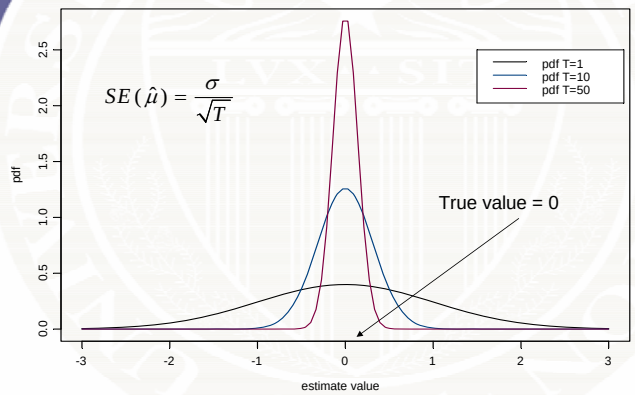
Monte Carlo Evaluation of Estimated Standard Error

```
> stdev(sim.means)  # SD of mu estimates across 1000
[1] 0.0106          # Monte Carlo experiments
> se.muhat["msft"]
     msft
 0.0107             # SE estimate from formula

> stdev(sim.vars)   # SD of sigma^2 estimates across 1000
[1] 0.0016          # Monte Carlo experiments
> se.sigma2hat["msft"]
     msft
 0.0016             # SE estimate from formula

> stdev(sim.sds)    # SD of sigma estimates across 1000
[1] 0.0075          # Monte Carlo experiments
> se.sigmahat["msft"]
     msft
 0.0075             # SE estimate from formula
```

Monte Carlo Evaluateion of 95% Confidence Interval Coverage

```
mu = 0.03
sd = 0.11
n.sim = 1000
set.seed(111)
mu.lower = rep(0,n.sim)    # initialize vectors
mu.upper = rep(0,n.sim)
for (sim in 1:n.sim) {
       sim.ret = rnorm(n.obs,mean=mu,sd=sd)
       mu.hat = mean(sim.ret)
       se.muhat = stdev(sim.ret)/sqrt(n.obs)
       mu.lower[sim] = mu.hat - 2*se.muhat
       mu.upper[sim] = mu.hat + 2*se.muhat
       in.interval = (mu >= mu.lower) & (mu <= mu.upper)
}
in.interval = (mu >= mu.lower) & (mu <= mu.upper)
sum(in.interval)/n.sim

>  0.934
```

Graphical Illustration of Consistency

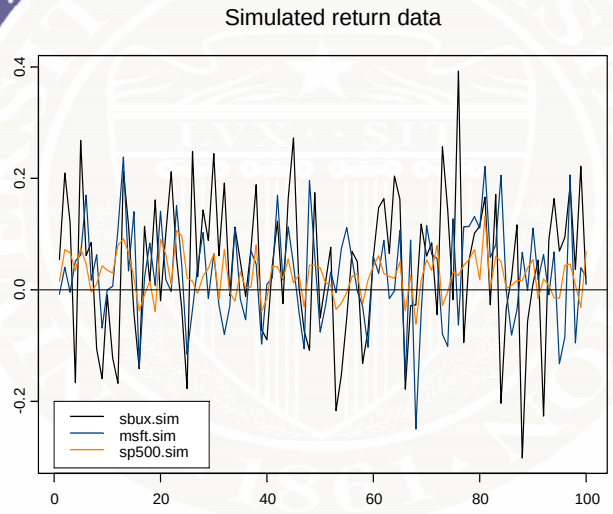


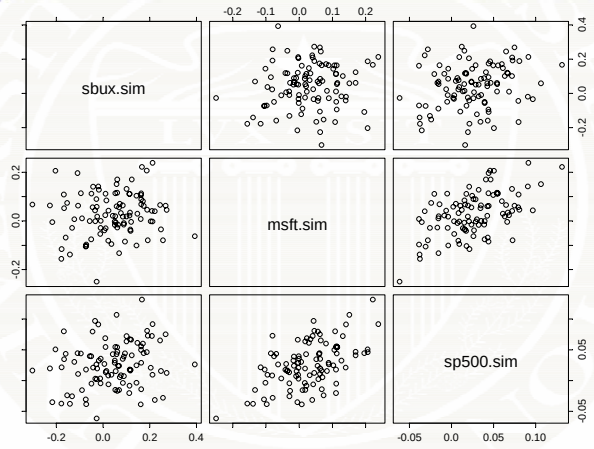
Estimating VaR in CER Model

```
# estimate quantiles from CER model
> qhat.05 = muhat.vals + sigmahat.vals*qnorm(0.05)
> qhat.05
     sbux      msft     sp500
 -0.19571 -0.14815 -0.049717

# estimate 5% VaR
> W0 = 100000
> VaRhat.05 = (exp(qhat.05)-1)*W0
> VaRhat.05
   sbux   msft   sp500
 -17775 -13769 -4850.1
```

Monte Carlo Simulation: 3 assets

```
# simulate data to match SBUX, MSFT, SP500
> muhat.vals
     sbux     msft    sp500
 0.027768 0.027564 0.012534
> cov.mat
            sbux      msft     sp500
 sbux 0.0184589 0.0040306 0.0021583
 msft 0.0040306 0.0114112 0.0022442
sp500 0.0021583 0.0022442 0.0014323

> nobs = 100
> set.seed(123)
> sim.e = rmvnorm(nobs,mean=rep(0,3),cov=cov.mat)
> sim.ret = muhat.vals + sim.e
> colIds(sim.ret) =
+ paste(colIds(returns.mat),".sim",sep="")
```

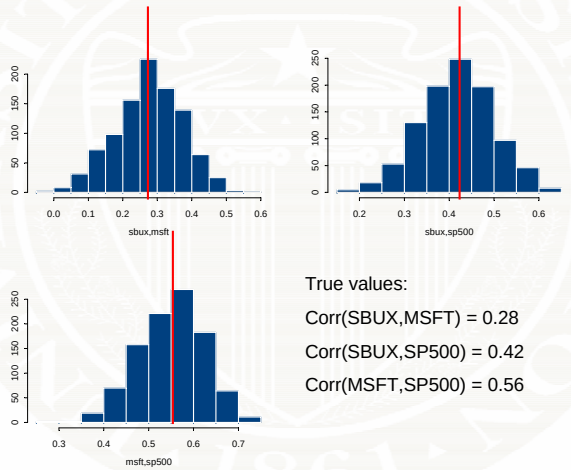




Monte Carlo Simulation Loop to Evaluate Distribution of correlation estimate

```
# generate 1000 samples from CER and compute correlations
n.obs = 100
n.sim = 1000
set.seed(111)
sim.corrs = matrix(0,n.sim,3)     # initialize vectors
colIds(sim.corrs) =
c("sbux,msft","sbux,sp500","msft,sp500")

for (sim in 1:n.sim) {
      sim.ret = rmvnorm(n.obs,mean=muhat.vals,cov=cov.mat)
      cor.mat = cor(sim.ret)
      sim.corrs[sim,] = cor.mat[lower.tri(cor.mat)]
}
```

Histograms of 1000 Monte Carlo Estimates of Correlation



True values:

Corr(SBUX,MSFT) = 0.28

Corr(SBUX,SP500) = 0.42

Corr(MSFT,SP500) = 0.56