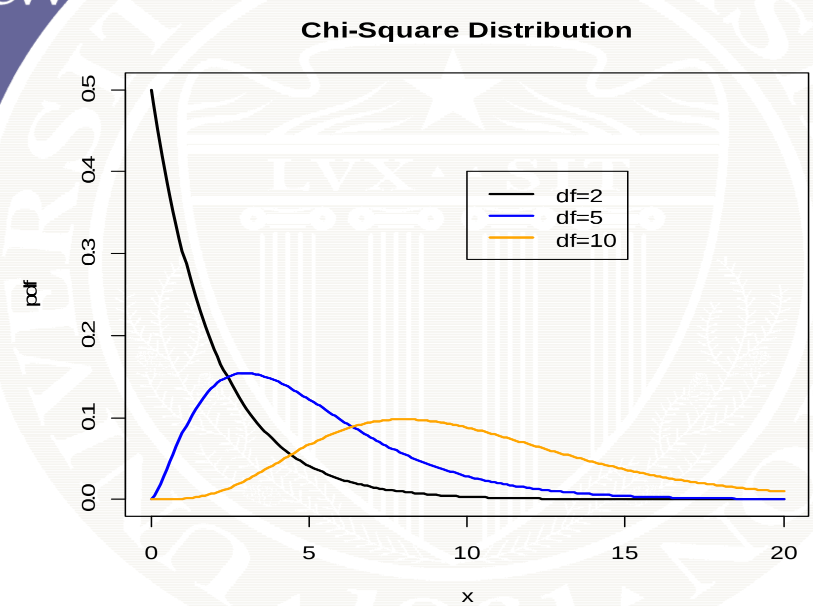
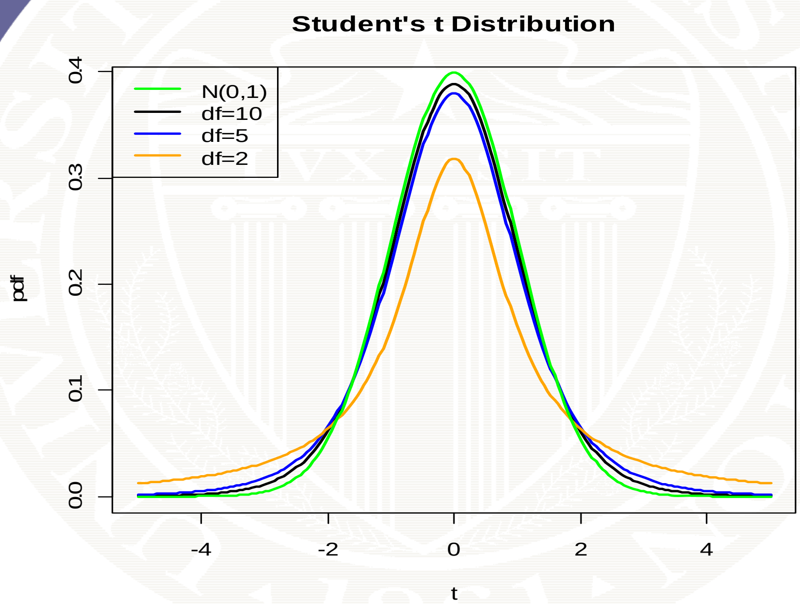
Hypothesis Testing in the CER Model

Econon 424
Eric Zivot
Fall 2008
Updated: October 28, 2008

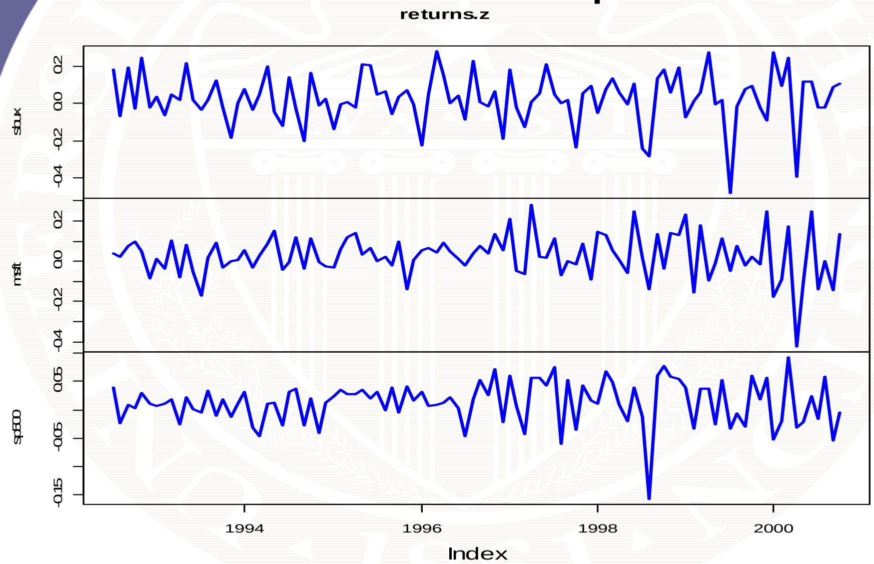




Quantiles of Student’s t

```
# N(0,1) = Student’s t with infinite df
> qnorm(c(0.01,0.05))
[1] -2.326348 -1.644854

# Student’s t with 10 df
> qt(c(0.01,0.05), df=10)
[1] -2.763769 -1.812461

# Student’s t with 5 df
> qt(c(0.01,0.05), df=5)
[1] -3.364930 -2.015048

# Student’s t with 2 df
> qt(c(0.01,0.05), df=2)
[1] -6.964557 -2.919986
```

Data for Examples



T = 100 months

Basic Significance Tests

```
# construct test by brute force
> nobs = dim(returns.z)[1]
> muhat.vals = apply(returns.z,2,mean)
> muhat.vals
    sbux    msft   sp500
 0.02777 0.02756 0.01253

> sigmahat.vals = apply(returns.z,2,sd)
> se.muhat = sigmahat.vals/sqrt(nobs)
> se.muhat
    sbux    msft    sp500
 0.01359 0.01068 0.003785
> t.stats = muhat.vals/se.muhat
> abs(t.stats)
  sbux msft sp500
 2.044 2.58 3.312
```

|t-stats| > 2 => we should reject H0: $\mu = 0$

Basic Significance Test

```
# compute 2-sided 5% critical values
> cv.2sided = qt(0.975, df=nobs-1)
> cv.2sided
[1] 1.984
> abs(t.stats) > cv.2sided
 sbux msft sp500
    T    T     T

# compute 2-sided p-values
> 2*(1-pt(abs(t.stats),df=nobs-1))
    sbux     msft     sp500
 0.04363 0.01134 0.001295
```


Basic Significance Test

```
# Test H0: mu = 0 vs H1: mu ≠ 0 for sbux
> t.test(returns.z[,"sbux"],
+        alternative="two.sided",
+        mu=0, conf.level=0.95)
     One-sample t-Test
data:  returns.z[, "sbux"]
 = 2.044, df = 99, p-value = 0.0436
alternative hypothesis:  mean is not equal to 0
95 percent confidence interval:
 0.0008099 0.0547264
sample estimates:
 mean of x
   0.02777
```

Here, the |t-stat| = 2.044 > 2 so we reject H0: $\mu = 0$ at 5% level. Also, p-value < 0.05.

Test for Specific Value

```
# Test H0: mu = 0.03 vs H1: mu ≠ 0.03
> t.test(returns.z[,"msft"],
+        alternative="two.sided",
+        mu = 0.03, conf.level=0.95)
     One-sample t-Test
data:  returns.z[, "msft"]
 = -0.2281, df = 99, p-value = 0.8201
alternative hypothesis:  mean is not equal to
0.03
95 percent confidence interval:
 0.006368 0.048760
sample estimates:
 mean of x
   0.02756
```

Here, the |t-stat| =.228 < 2 so we do not reject H0: μ = 0.03 at 5% level. Also, p-value > 0.05.

One-Sided Test for Sign

```
# Test H0: mu = 0 vs H1: mu > 0 for msft
> t.test(returns.z[,"msft"],
+        alternative="greater",
+        mu = 0, conf.level=0.95)

     One-sample t-Test
data:  returns.z[, "msft"]
 = 2.58, df = 99, p-value = 0.0057
alternative hypothesis:  mean is greater
than 0
95 percent confidence interval:
 0.009827      NA
sample estimates:
 mean of x
   0.02756
```

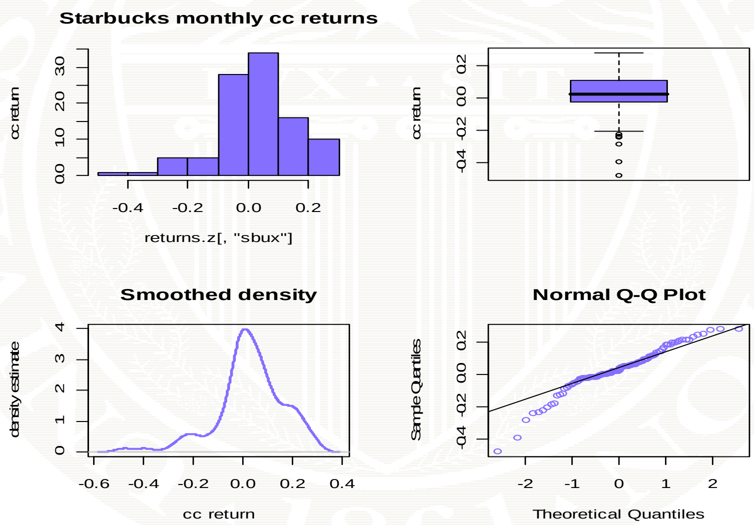
Here, the t-stat = 2.58 > 1.645 so we reject H0: μ = 0 at 5% level. Also, p-value < 0.05.

Paired t-test for Equality of Means

```
# Test H0: mu_msft = mu_sbux vs. H1: mu_msft /=
# mu_sbux
> t.test(x=returns.z[,"msft"],
+        y=returns.z[,"sbux"],
+        paired=T)


         Paired t-Test

 data:   returns.z[, "msft"] and returns.z[, "sbux"]
  = -0.0138, df = 99, p-value = 0.989
 alternative hypothesis:  mean of differences is not
 equal to 0
 95 percent confidence interval:
  -0.02951  0.02910
 sample estimates:
  mean of x - y
     -0.0002042
```

Here, the |t-stat| = .0138 < 2 so we do not reject H0: $\mu = 0$ at 5% level. Also, p-value > 0.05.

Test for Normal Distribution



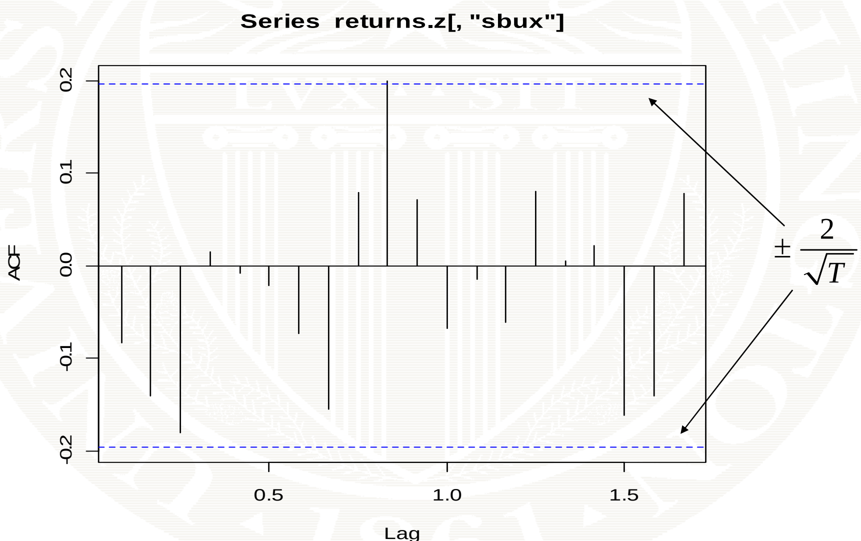
Jarque-Bera Test for Normality

```
> sbux.skew = skewness(returns.z[,"sbux"])
> sbux.ekurt= kurtosis(returns.z[,"sbux"])
> sbux.skew
[1] -0.8272737
> sbux.ekurt
[1] 1.761706
> JB = nobs*(sbux.skew^2 + 0.25*sbux.ekurt^2)/6
> JB
[1] 24.33806
```

JB = 24.34 > 6 so we reject H0: returns on sbux are normally distributed at the 5% level

```
> p.value = 1 - pchisq(JB, df = 2)
> p.value
[1] 5.188691e-06
```


Testing for Serial Correlation




```
> acf(returns.ts[,"sbux"]
```

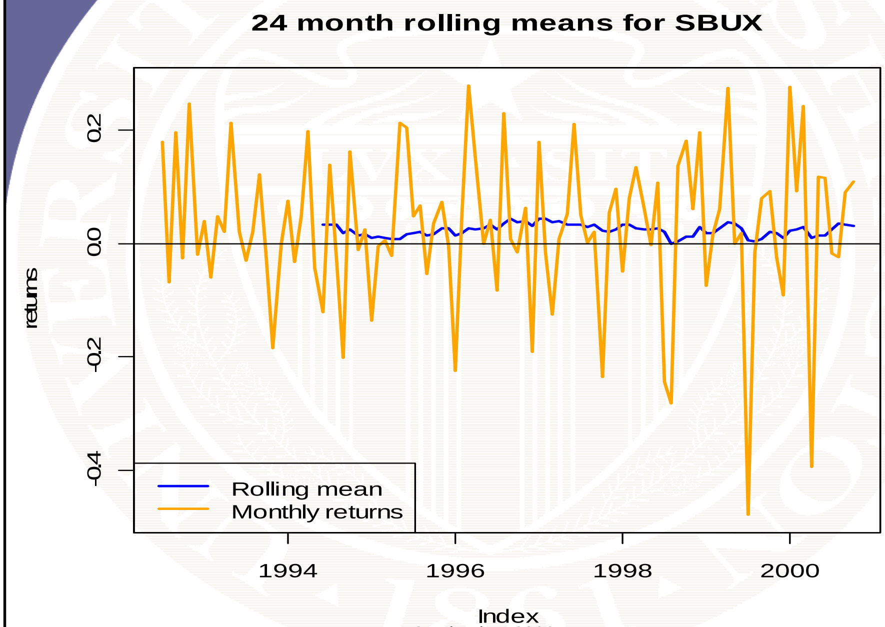
Two Sample t-test for equality of Means

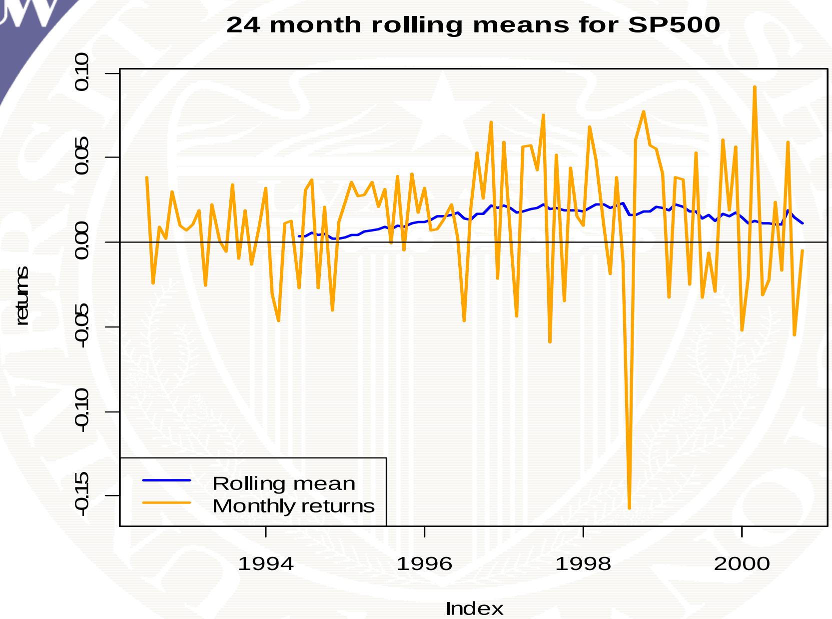
```
> t.test(x=returns.z[1:50,"msft"],
+        y=returns.z[51:100,"msft"],
+        paired=F)

    Pooled-Variance Two-Sample t-Test

data:  returns.z[1:50, "msft"] and
returns.[51:100, "msft"]
t = -0.2337, df = 98, p-value = 0.8157
alternative hypothesis:  difference in means
is not equal to 0
95 percent confidence interval:
 -0.04762  0.03758
sample estimates:
 mean of x mean of y
   0.02506   0.03007
```

|t-stat| = 0.2237 < 2 so do not reject H0: E[dt] = 0 at 5% level. Also, p-value > 0.05.

Compute Rolling Means using zoo function `rollapply()`

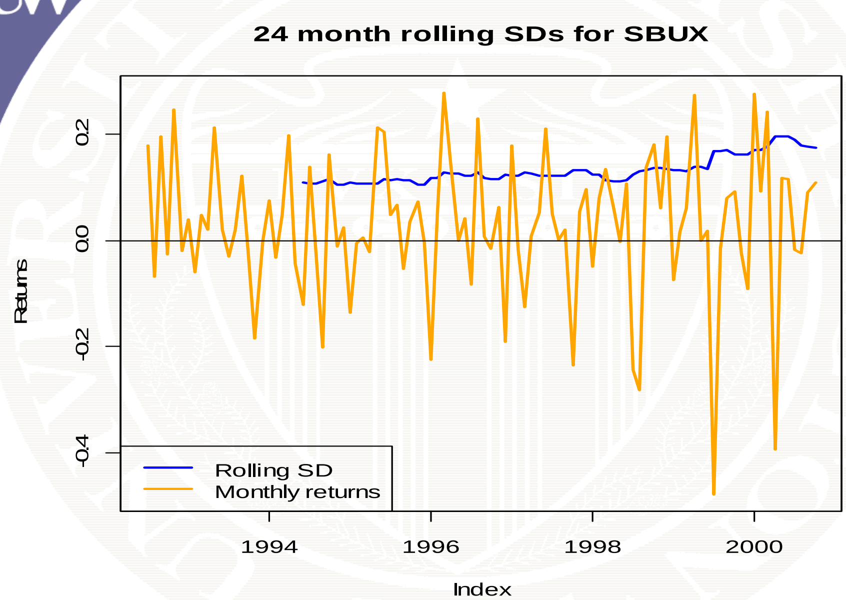
```
> roll.muhat = rollapply(returns.z[,"sbux"], width=24,
+                        FUN=mean, align="right")
> class(roll.muhat)
[1] "zooreg" "zoo"

> roll.muhat[1:5]
   1994(6)    1994(7)    1994(8)    1994(9)   1994(10)
0.03415205 0.03244166 0.03418205 0.01757783 0.02538356
```

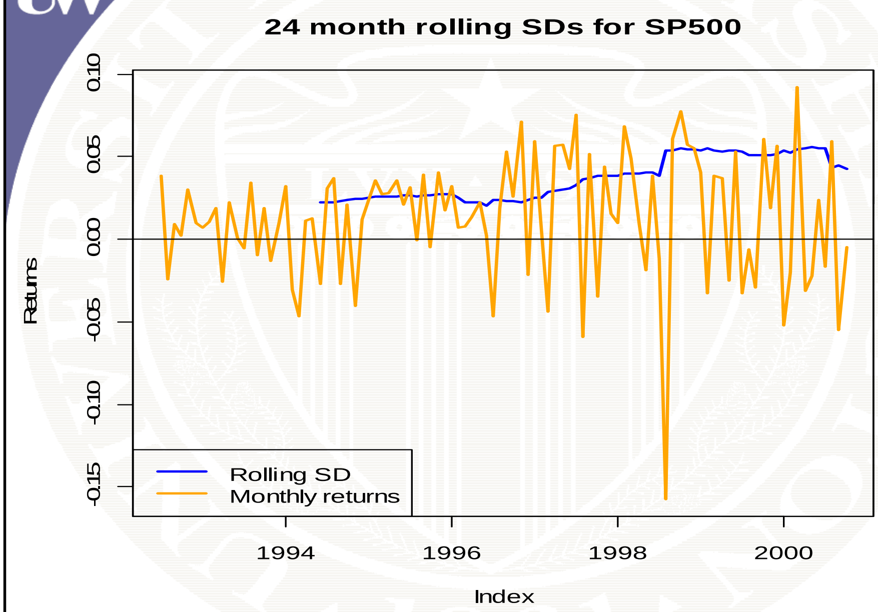

24 month rolling means for SBUX
returns
0.2
0.0
-0.2
-0.4
Rolling mean
Monthly returns
1994
1996
1998
2000
Index


24 month rolling means for SP500
returns
0.10
0.05
0.00
-0.05
-0.10
-0.15
Rolling mean
Monthly returns
1994
1996
1998
2000
Index

Compute Rolling SDs Using zoo Function `rollapply()`

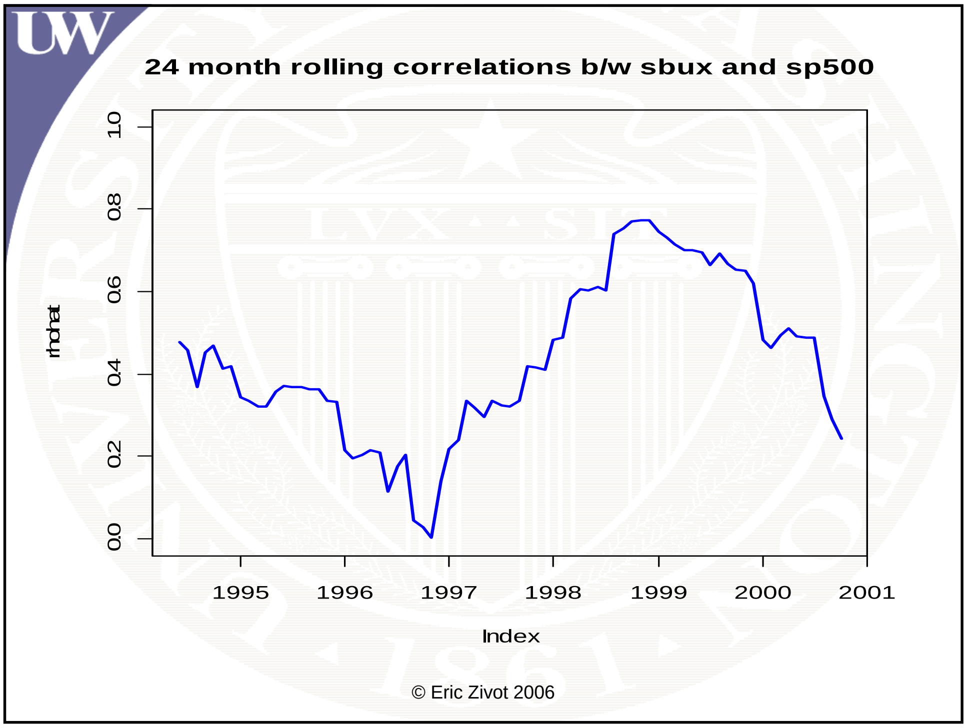
```
> roll.sigmahat = rollapply(returns.z[,"sbux"], width=24,
+                           FUN=sd, align="right")
> class(roll.sigmahat)
[1] "zooreg" "zoo"

> roll.sigmahat[1:5]
  1994(6)   1994(7)   1994(8)   1994(9)  1994(10)
0.1100916 0.1080346 0.1066745 0.1113659 0.1147587
```



Compute Rolling Correlations Using zoo Function `rollapply()`

```
# compute rolling estimates b/w sp500 and sbux

rhohat = function(x) {
      cor(x)[1,2]
}

> roll.rhohat = rollapply(returns.z[,c("sp500","sbux")],
+                         width=24,FUN=rhohat,
+                         by.column=FALSE, align="right")
> class(roll.rhohat)
[1] "zooreg" "zoo"

> roll.rhohat[1:5]
  1994(6)   1994(7)   1994(8)   1994(9)   1994(10)
0.4786364 0.4570107 0.3694040 0.4515018 0.4682775
```


UW
24 month rolling correlations b/w sbux and sp500
rhohat
0.0
0.2
0.4
0.6
0.8
1.0
1995
1996
1997
1998
1999
2000
2001
Index
© Eric Zivot 2006