(Stock) Liquidity and the business cycle

We consider the analysis in Næs, Skjeltorp, and Ødegaard (2011). Show how to do some of the estimations (those related to forecasting).

Forecasting exercises

We show the calculation of the forecasting tests.
Investigate a situation where

- x_t is the variable to forecast
- y_t some other variable.

Proposed forecasting model:

$$x_{t+1} = a + b_x x_t + b_y y_t + \varepsilon_{t+1}$$

(linear ADL(1,1) model.)
Look at the case where

- x_t is growth in US GDP, and
- y_t is a measure of stock market liquidity, specifically the Amihud Illiquidity Ratio (ILR).

Data

GDP is available on a quarterly basis for the US.
Download the variable GDPC96 from FRED.
(Note that this is not exactly the variable used in the paper, but is should be close enough).
Liquidity variable *ILR*: available on the homepage.

Data ctd

Reading in of the data

```
> liq  <- read.csv("../../data_us/US_illiquidity_series_mar
                    sep=",", header=TRUE)
> ILR  <- zooreg(liq$ILR_NYSE, frequency=4,
                  order.by=as.yearqtr(liq$YearQuarter,format
> head(ILR)
1946 Q1 1946 Q2 1946 Q3 1946 Q4 1947 Q1 1947 Q2
1.55019 1.42891 2.57527 3.12041 2.82972 3.99412
>
> RGDP <- read.zoo("../../data_us/GDPC96.csv",
                    sep=",",format="%Y-%m-%d",header=TRUE)
> RGDP <- zooreg(coredata(RGDP), frequency=4,
                  order.by=as.yearqtr(index(RGDP)))
> head(RGDP)
 1947 Q1  1947 Q2  1947 Q3  1947 Q4  1948 Q1  1948 Q2
1934.471 1932.281 1930.315 1960.705 1989.535 2021.851
> dRGDP <- diff(log(RGDP))
> dILR <- diff(ILR)
```

In sample model estimation

Estimating the ADL(1,1) model.
Test whether the coefficient on *dILR* is significant.
(In sample model specification testing.)
R-usage: load the `dyn` library, allows for lag structures in the models

```
> library(dyn)
> dGDP <- diff(log(RGDP))
> dILR <- diff(ILR)
> data <- merge(dGDP,dILR,all=FALSE)
> dGDP <- data$dGDP
> dILR <- data$dILR
> regr <- dyn$lm(dGDP~lag(dGDP)+lag(dILR))
```

In sample model estimation

The output of this regression is

```
> summary(regr)

Call:
lm(formula = dyn(dGDP ~ lag(dGDP) + lag(dILR)))

Residuals:
       Min         1Q     Median         3Q        Max
-0.0306094 -0.0048075  0.0000893  0.0049269  0.0292395

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0053457  0.0007416   7.209 7.11e-12 ***
lag(dGDP)   0.3733990  0.0585509   6.377 9.01e-10 ***
lag(dILR)   0.0074651  0.0025429   2.936  0.00365 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
```

In sample model estimation

Which is much better viewed as

```
> stargazer(regr)
```

	Dependent variable:
	dGDP ~lag(dGDP) + lag(dILR)
lag(dGDP)	0.373***
	(0.059)
lag(dILR)	0.007***
	(0.003)
Constant	0.005***
	(0.001)
Observations	246
Adjusted R^2	0.149

In sample forecast evaluation

Suppose we use a mean square criterion for comparing forecasts.
The easiest way to do this comparison is to compare R^2.
To do so estimate the restricted model:

```
> regr1 <- dyn$lm(dGDP~lag(dGDP))
```

And compare R^2:

In sample forecast evaluation

	Dependent variable:	
	dGDP ~lag(dGDP) + lag(dILR)	dGDP ~lag(dGDP)
	(1)	(2)
lag(dGDP)	0.373***	0.349***
	(0.059)	(0.059)
lag(dILR)	0.007***	
	(0.003)	
Constant	0.005***	0.005***
	(0.001)	(0.001)
Observations	246	246
Adjusted R^2	0.149	0.122
Note:		*p<0.1; **p<0.05; ***p<0.01

Here the $\overline{R}^2$ increases from 0.122 to 0.149.

So, an improvement in the in-sample forecasting power.

Alternative framework: VAR

Another possibility for in-sample estimation: A VAR estimation, where we use *dGDP* and *dILR* as the elements of the vector. Collect the data into one matrix, and then call the `vars` routine

```
> library(vars)
> va <- VAR(data)
```

Alternative framework: VAR

```
> summary(va)

VAR Estimation Results:
=========================
Endogenous variables: dGDP, dILR
Deterministic variables: const
Sample size: 246
Log Likelihood: 845.376
Roots of the characteristic polynomial:
0.2046 0.03167
Call:
VAR(y = data)
```

Alternative framework: VAR

```
Estimation results for equation dGDP:
=====================================
dGDP = dGDP.l1 + dILR.l1 + const

         Estimate Std. Error t value Pr(>|t|)
dGDP.l1  0.3147518  0.0585512   5.376 1.79e-07 ***
dILR.l1 -0.0123299  0.0023734  -5.195 4.33e-07 ***
const    0.0054640  0.0007373   7.410 2.08e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘

Residual standard error: 0.008746 on 243 degrees of freedom
Multiple R-Squared: 0.2133,Adjusted R-squared: 0.2068
F-statistic: 32.93 on 2 and 243 DF,  p-value: 2.208e-13
```

Alternative framework: VAR

```
Estimation results for equation dILR:
=====================================
dILR = dGDP.l1 + dILR.l1 + const

        Estimate Std. Error t value Pr(>|t|)
dGDP.l1  2.52853    1.50372   1.682   0.0939 .
dILR.l1 -0.07846    0.06095  -1.287   0.1992
const   -0.03745    0.01894  -1.977   0.0491 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘

Residual standard error: 0.2246 on 243 degrees of freedom
Multiple R-Squared: 0.02119,Adjusted R-squared: 0.01314
F-statistic: 2.631 on 2 and 243 DF,  p-value: 0.07407
```

Alternative framework: VAR

We then do causality tests using the estimated VAR:
Of primary interest is whether *dILR* Granger causes *dGDP*.

```
> causality(va,cause="dILR")
$Granger
Granger causality H0: dILR do not Granger-cause dGDP

data:  VAR object va
F-Test = 26.9891, df1 = 1, df2 = 486, p-value = 3.016e-07
n
$Instant
H0: No instantaneous causality between: dILR and dGDP

data:  VAR object va
Chi-squared = 13.32, df = 1, p-value = 0.0002626
```

Here we reject the null of no Granger causality.

Out of Sample comparisons

We use the forecast library when we want to construct forecasts.

```
> library(forecast)
```

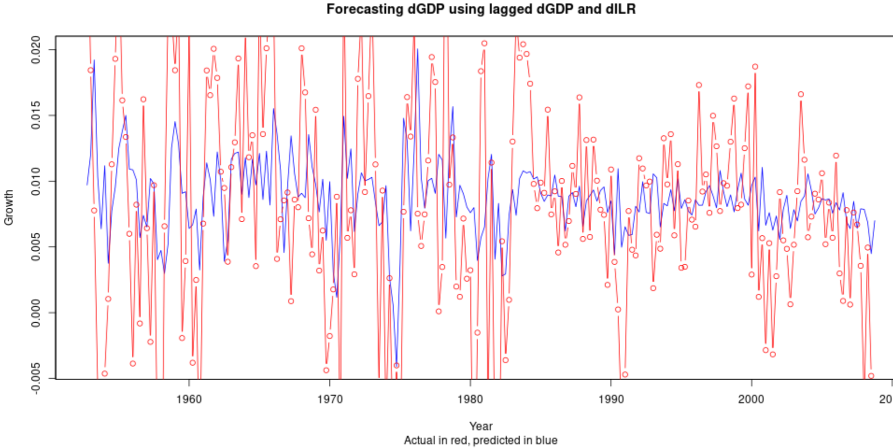
Transform the data into what we need

```
> dILR   <- diff(ILR)
> growth <- diff(log(RGDP))
> ## use two lags of growth since the current lag is not ob
> lgrowth <- lag(growth,-2)
> ldILR    <- lag(dILR,-1)
> data <-   merge(dILR,growth,ldILR,lgrowth,all=FALSE)
> dILR <- data[,1]
> growth <- data[,2]
> ldILR <- data[,3]
> lgrowth <- data[,4]
```

Out of Sample comparisons

```
> n <- length(growth)
> actual <-NULL
> predicted0 <- NULL
> predicted1 <- NULL
> for (t in 21:n){
+   g <- growth[1:(t-1)]
+   lg <-  lgrowth[1:(t-1)]
+   lsp <- ldILR[1:(t-1)]
+   pred0 <- lm( g ~ lg )
+   pred1 <- lm( g ~ lg + lsp)
+   nd0 <- data.frame(lg <- lgrowth[t])
+   fc0 <- forecast.lm(pred0, nd0 ,h=1)
+   nd1 <- data.frame(lsp <-ldILR[t], lg <- lgrowth[t])
+   fc1 <- forecast.lm(pred1, nd1 ,h=1)
+   actual <- c(actual,growth[t])
+   predicted0 <- c(predicted0, fc0$mean)
+   predicted1 <- c(predicted1, fc1$mean)
+ }
```

Out of Sample comparisons



Out of Sample comparisons

We next calculate the various measures of accuracy of the forecasts.
Note: R is restricted U is unrestricted

```
> resR <- actual-predicted0
> resU <- actual-predicted1
> p=length(resR)
> h=1
> MSEr <- sum(resR^2)/p
[1] 6.881447e-05
> MSEu <- sum(resU^2)/p
[1] 6.202594e-05
> ENCNEW <- (p-h+1)* (sum(resR^2-resU*resR)/p)/MSEu
[1] 18.78918
> MSEF <- (p-h+1)*(MSEr-MSEu)/MSEu
[1] 20.79485
> MSEu/MSEr
[1] 0.9013503
```

But these numbers are very different from what you have in table

Out of Sample comparisons

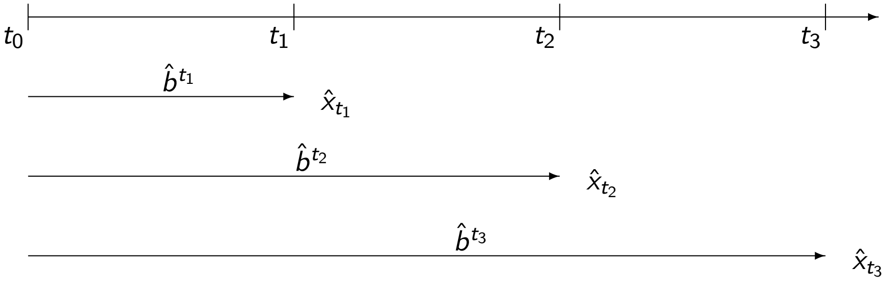
This is to illustrate another issue in forecasting:
Do you use the whole sample in the forecasts?
To repeat part of the loop:

```
+    g <- growth[1:(t-1)]
+    lg <-  lgrowth[1:(t-1)]
+    lsp <- ldILR[1:(t-1)]
+    pred0 <- lm( g ~ lg )
+    pred1 <- lm( g ~ lg + lsp)
```

As t grows, the sample expands.

Out of Sample comparisons

Let us show with a picture:

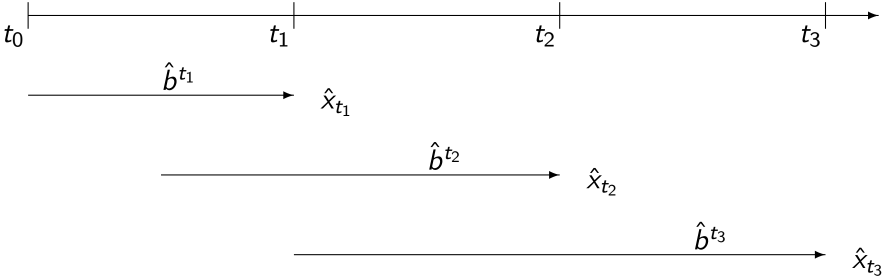


Out of Sample comparisons

If you dig into the paper, you will see that we are using

> *... a rolling estimation scheme with a fixed width of 20 quarters...*

So we are using the following rolling type of estimation:



Out of Sample comparisons

The loop is instead specified as

```
> wlength <- 20
> for (t in (wlength+1):n) {
+   g <- growth[(t-wlength):(t-1)]
+   lg <-  lgrowth[(t-wlength):(t-1)]
+   lsp <- ldILR[(t-wlength):(t-1)]
+   pred0 <- lm( g ~ lg )
+   pred1 <- lm( g ~ lg + lsp)
+   nd0 <- data.frame(lg <- lgrowth[t])
+   fc0 <- forecast.lm(pred0, nd0 ,h=1)
+   nd1 <- data.frame(lsp <-ldILR[t],
+                     lg <- lgrowth[t])
+   fc1 <- forecast.lm(pred1, nd1 ,h=1)
+   actual <- c(actual,growth[t])
+   predicted0 <- c(predicted0, fc0$mean)
+   predicted1 <- c(predicted1, fc1$mean)
+ }
```

Out of Sample comparisons

With that specification we get much closer to the paper:

```
> resR <- actual-predicted0
> resU <- actual-predicted1
> p=length(resR)
> MSEr <- sum(resR^2)/p
[1] 9.348539e-05
> MSEu <- sum(resU^2)/p
[1] 8.07512e-05
> ENCNEW <- (p-h+1)* (sum(resR^2-resU*resR)/p)/MSEu
[1] 48.24702
> MSEF <- (p-h+1)*(MSEr-MSEu)/MSEu
[1] 35.48175
> MSEu/MSEr
[1] 0.8637841
```

Still not completely the same: Slightly different macro series / different computer package

Randi Næs, Johannes A Skjeltorp, and Bernt Arne Ødegaard. Stock market liquidity and the Business Cycle. *Journal of Finance*, LXVI:139–176, February 2011.