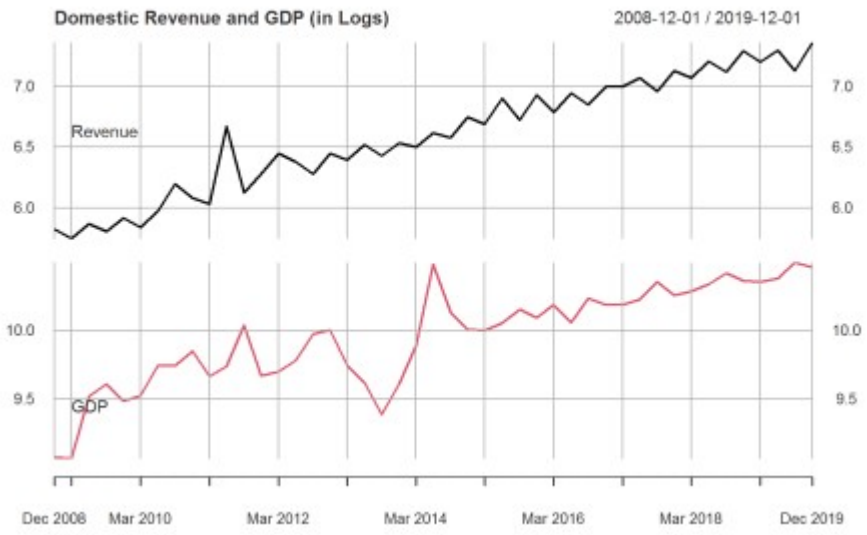
There are several ways to forecast tax revenue. The IMF *Financial Programming Manual* reviews 3 of them: (i) the effective tax rate approach; (ii) the elasticity approach; and (iii) the regression approach. Approach (iii) typically results in the most accurate short-term forecasts. The simple regression approach regresses tax revenue on its own lags and GDP with some lags.

In the absence of large abrupt shifts in the tax base, domestic revenue can be assumed to have a linear relationship with GDP. Since however both revenue and GDP are typically non-stationary series, this relationship often takes the form of cointegration. The correct way to deal with cointegrated variables is to specify and Error Correction Model (ECM). This blog post will briefly demonstrate the specification of an ECM to forecast the tax revenue of a developing economy[1]. First we examine the data, which is in local currency and was transformed using the natural logarithm.

```
library(haven)       # Import from STATA
library(collapse)    # Data transformation
library(magrittr)    # Pipe operators %>%
library(tseries)     # Time series tests
library(lmtest)      # Linear model tests
library(sandwich)    # Robust standard errors
library(dynlm)       # Dynamic linear models
library(jtools)      # Enhanced regression summary
library(xts)         # Extensible time-series + pretty plots

# Loading the data from STATA
data <- read_dta("data.dta") %>% as_factor

# Generating a date variable
settfm(data, Date = as.Date(paste(Year, unattrib(Quarter) * 3L, "1",
sep = "/")))

# Creating time series matrix X
X <- data %$% xts(cbind(lrev, lgdp), order.by = Date, frequency = 4L)

# (Optional) seasonal adjustment using X-13 ARIMA SEATS
  # library(seasonal)
  # X <- dapply(X, function(x) predict(seas(ts(x, start = c(1997L, 3L),
frequency = 4L))))
  # # X <- X["2015/", ] # Optionally restricting the sample to after
2014

# Plotting the raw data
plot(na_omit(X)[-1L, ] %>% setColnames(.c(Revenue, GDP)),
     multi.panel = TRUE, yaxis.same = FALSE,
     main = "Domestic Revenue and GDP (in Logs)",
     major.ticks = "years", grid.ticks.on = "years")
```




```
# Plotting the log-differenced data
plot(na_omit(D(X)), legend.loc = "topleft",
     main = "Revenue and GDP in Quarterly Log-Differences",
     major.ticks = "years", grid.ticks.on = "years")
```

The data was not seasonally adjusted as revenue and GDP exhibit similar seasonal patterns. Summarizing the log-differenced using a function designed for panel data allows us to assess the extent of seasonality relative to overall variation.

```
# Summarize between and within quarters
tfmv(data, 3:4, D) %>% qsu(pid = lrev + lgdp ~ Quarter)
## , , lrev
##
##             N/T    Mean      SD      Min     Max
## Overall      91  0.0316  0.1545  -0.5456  0.6351
## Between       4  0.0302  0.1275  -0.0997  0.1428
## Within    22.75  0.0316  0.1077  -0.4144  0.5239
##
## , , lgdp
##
##             N/T    Mean      SD      Min     Max
## Overall      45  0.0271   0.183  -0.3702  0.5888
```

```
## Between      4  0.0291  0.0767  -0.0593  0.1208
## Within   11.25  0.0271    0.17  -0.3771  0.4951
```

For log revenue, the standard deviation between quarters is actually slightly higher than the within-quarter standard deviation, indicating a strong seasonal component. The summary also shows that we have 23 years of quarterly revenue data but only 11 years of quarterly GDP data.

An ECM is only well specified if both series are integrated of the same order and cointegrated. This requires a battery of tests to examine the properties of the data before specifying a model[2]. For simplicity I will follow the 2-Step approach of Engele & Granger here, although I note that the more sophisticated Johannsen procedure is available in the *urca* package.

```
# Testing log-transformed series for stationarity: Revenue is clearly
non-stationary
adf.test(X[, "lrev"])
##
##  Augmented Dickey-Fuller Test
##
## data:  X[, "lrev"]
## Dickey-Fuller = -0.90116, Lag order = 4, p-value = 0.949
## alternative hypothesis: stationary

kpss.test(X[, "lrev"], null = "Trend")
##
##  KPSS Test for Trend Stationarity
##
## data:  X[, "lrev"]
## KPSS Trend = 0.24371, Truncation lag parameter = 3, p-value = 0.01

# ADF test fails to reject the null of non-stationarity at 5% level
adf.test(na_omit(X[, "lgdp"]))
##
##  Augmented Dickey-Fuller Test
##
## data:  na_omit(X[, "lgdp"])
## Dickey-Fuller = -3.4532, Lag order = 3, p-value = 0.06018
## alternative hypothesis: stationary

kpss.test(na_omit(X[, "lgdp"]), null = "Trend")
##
##  KPSS Test for Trend Stationarity
##
## data:  na_omit(X[, "lgdp"])
## KPSS Trend = 0.065567, Truncation lag parameter = 3, p-value = 0.1

# Cointegrated: We reject the null of no cointegration
po.test(X[, .c(lrev, lgdp)])
##
##  Phillips-Ouliaris Cointegration Test
##
## data:  X[, .c(lrev, lgdp)]
## Phillips-Ouliaris demeaned = -33.219, Truncation lag parameter = 0,
```

```
p-value = 0.01
```

The differenced revenue and GDP series are stationary (tests not shown), so both series are I(1), and GDP is possibly trend-stationary. The Phillips-Ouliaris test rejected the null that both series are not cointegrated.

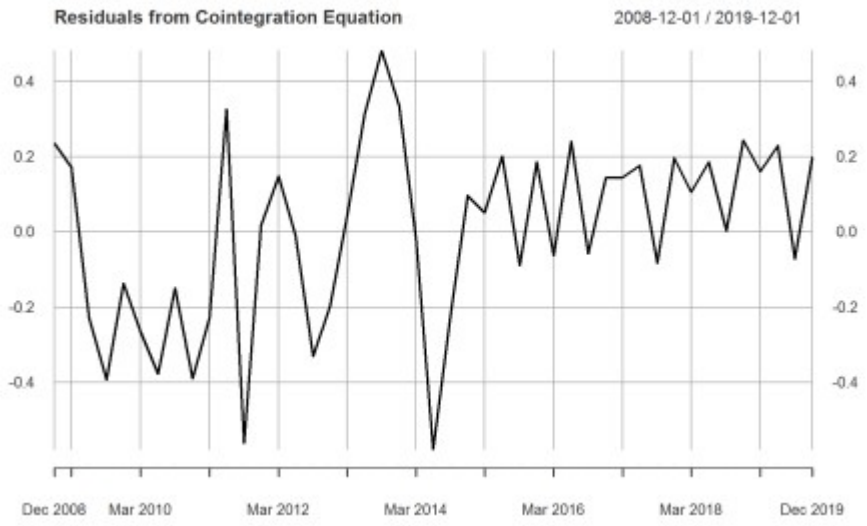
Below the cointegration relationship is estimated. A dummy is included for extreme GDP fluctuations between Q3 2013 and Q3 2014, which may also be related to a GDP rebasing. Since the nature of these events is an increase in volatility rather than the level of GDP, the dummy is not a very effective way of dealing with this irregularity in the data, but for simplicity we will go with it.

```
# Adding extreme GDP events dummy
X <- cbind(X, GDPdum = 0)
X["2013-09/2014-09", "GDPdum"] <- 1

# This estimates the cointegration equation
cieq <- dynlm(lrev ~ lgdp + GDPdum, as.zoo(X))

# Summarizing the model with heteroskedasticity and autocorrelation
consistent (HAC) errors
summ(cieq, digits = 4L, vcov = vcovHAC(cieq))
```

Observations	46 (44 missing obs. deleted)			
Dependent variable	lrev			
Type	OLS linear regression			
	F(2,43)	64.4122		
	R^2	0.7497		
	Adj. R^2	0.7381		
	Est.	**S.E.**	**t val.**	**p**
(Intercept)	-4.7667	1.2958	-3.6787	0.0006
lgdp	1.1408	0.1293	8.8208	0.0000
GDPdum	0.0033	0.2080	0.0160	0.9873

Standard errors: User-specified

```
# Residuals of cointegration equation
res <- as.xts(cieq$residuals)
plot(res[-1L, ], main = "Residuals from Cointegration Equation",
     major.ticks = "years", grid.ticks.on = "years")
```




```
# Testing residuals: Stationary
adf.test(res)
##
##  Augmented Dickey-Fuller Test
##
## data:  res
## Dickey-Fuller = -4.3828, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary

kpss.test(res, null = "Trend")
##
##  KPSS Test for Trend Stationarity
##
## data:  res
## KPSS Trend = 0.045691, Truncation lag parameter = 3, p-value = 0.1
```

Apart from a cointegration relationship which governs the medium-term relationship of revenue and GDP, revenue may also be affected by past revenue collection and short-term fluctuations in GDP. A sensible and simple specification to forecast revenue in the short to medium term (assuming away shifts in the tax base) is thus provided by the general form of a bivariate ECM:

\[\begin{equation} A(L)\Delta r_t = \gamma + B(L)\Delta y_t + \alpha (r_{t-t} - \beta_0 - \beta_i y_{t-1}) + v_t, \end{equation}\] where \[\begin{align*} A(L) &= 1- \sum_{i=1}^p L^i = 1 - L - L^2 - \dots - L^p, \\ B(L) &= \sum_{i=0}^q L^i= 1 + L + L^2 + \dots + L^q \end{align*}\] are polynomials in the lag operator \(L\) of order \(p\) and \(q\), respectively. Some empirical investigation of the fit of the model for different lag-orders \(p\) and \(q\) established that \(p = 2\) and \(q = 1\) gives a good fit, so that the model estimated is

\[\begin{equation} \Delta r_t = \gamma + \Delta r_{t-1} + \Delta r_{t-2} + \Delta y_t + \Delta y_{t-1} + \alpha (r_{t-t} - \beta_0 - \beta_i y_{t-1}) + v_t. \end{equation}\]

```
# Estimating Error Correction Model (ECM)
ecm <- dynlm(D(lrev) ~ L(D(lrev), 1:2) + L(D(lgdp), 0:1) + L(res) +
GDPdum,
             as.zoo(merge(X, res)))

summ(ecm, digits = 4L, vcov = vcovHAC(ecm))
```

Observations	44 (44 missing obs. deleted)
Dependent variable	D(lrev)
Type	OLS linear regression
F(6,37)	12.9328
R²	0.6771
Adj. R²	0.6248

	Est.	S.E.	t val.	p
(Intercept)	0.0817	0.0197	4.1440	0.0002
L(D(lrev), 1:2)1	-0.9195	0.1198	-7.6747	0.0000
L(D(lrev), 1:2)2	-0.3978	0.1356	-2.9342	0.0057
L(D(lgdp), 0:1)1	0.1716	0.0942	1.8211	0.0767
L(D(lgdp), 0:1)2	-0.2654	0.1128	-2.3532	0.0240
L(res)	-0.2412	0.1096	-2.2008	0.0341
GDPdum	0.0212	0.0207	1.0213	0.3138

Standard errors: User-specified

```
# Regression diagnostic plots
# plot(ecm)

# No heteroskedasticity (null of homoskedasticity not rejected)
bptest(ecm)
##
##  studentized Breusch-Pagan test
##
## data:  ecm
## BP = 9.0161, df = 6, p-value = 0.1727

# Some autocorrelation remainig in the residuals, but negative
cor.test(resid(ecm), L(resid(ecm)))
##
##  Pearson's product-moment correlation
##
## data:  resid(ecm) and L(resid(ecm))
## t = -1.8774, df = 41, p-value = 0.06759
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5363751  0.0207394
## sample estimates:
##       cor
## -0.281357

dwtest(ecm)
##
##  Durbin-Watson test
##
## data:  ecm
## DW = 2.552, p-value = 0.9573
## alternative hypothesis: true autocorrelation is greater than 0
```

```
dwtest(ecm, alternative = "two.sided")
##
##  Durbin-Watson test
##
## data:  ecm
## DW = 2.552, p-value = 0.08548
## alternative hypothesis: true autocorrelation is not 0
```

The regression table shows that the log-difference in revenue strongly responds to its own lags, the lagged log-difference of GDP and the deviation from the previous period equilibrium, with an adjustment speed of $\alpha = -0.24$.

The statistical properties of the equation are also acceptable. Errors are homoskedastic and serially uncorrelated at the 5% level. The model is nevertheless reported with heteroskedasticity and autocorrelation consistent (HAC) standard errors.

Curiously, changes in revenue in the current quarter do not seem to be very strongly related to changes in GDP in the current quarter, which could also be accounted for by data being published with a lag. For forecasting this is advantageous since if a specification without the difference of GDP can be estimated that fits the data well, then it may not be necessary to first forecast quarterly GDP and include it in the model in order to get a decent forecasts of the revenue number for the next quarter. Below a specification without the difference in GDP is estimated.

```
# Same using only lagged differences in GDP
ecm2 <- dynlm(D(lrev) ~ L(D(lrev), 1:2) + L(D(lgdp)) + L(res) + GDPdum,
              as.zoo(merge(X, res)))

summ(ecm2, digits = 4L, vcov = vcovHAC(ecm2))
```

Observations	45 (44 missing obs. deleted)			
Dependent variable	D(lrev)			
Type	OLS linear regression			
	F(5,39)	15.1630		
	R²	0.6603		
	Adj. R²	0.6168		
	Est.	**S.E.**	**t val.**	**p**
(Intercept)	0.0839	0.0206	4.0653	0.0002
L(D(lrev), 1:2)1	-0.9111	0.1162	-7.8424	0.0000
L(D(lrev), 1:2)2	-0.3910	0.1305	-2.9950	0.0047
L(D(lgdp))	-0.2345	0.0995	-2.3574	0.0235
L(res)	-0.1740	0.0939	-1.8524	0.0716
GDPdum	0.0244	0.0328	0.7428	0.4621

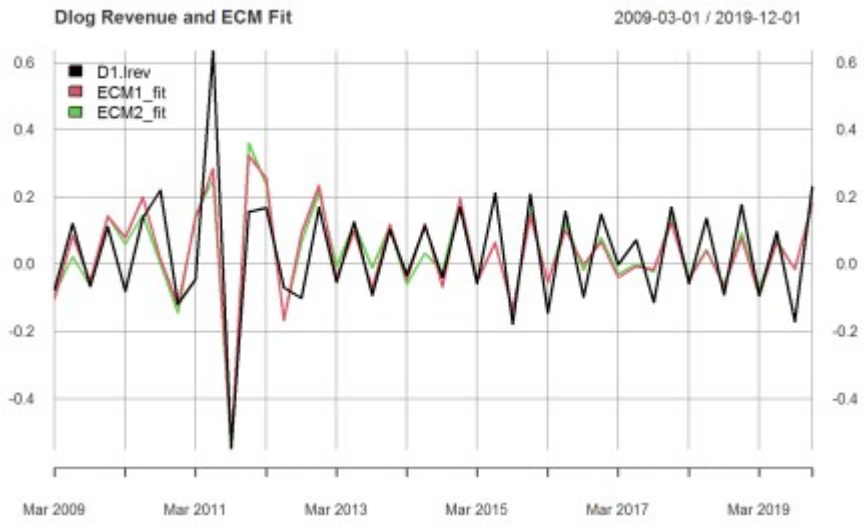
Standard errors: User-specified

```
# plot(ecm2)

bptest(ecm2)
##
##  studentized Breusch-Pagan test
```

```
##
## data:  ecm2
## BP = 7.0511, df = 5, p-value = 0.2169

cor.test(resid(ecm2), L(resid(ecm2)))
##
##  Pearson's product-moment correlation
##
## data:  resid(ecm2) and L(resid(ecm2))
## t = -1.701, df = 42, p-value = 0.09634
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.51214976  0.04651674
## sample estimates:
##        cor
## -0.2538695

dwtest(ecm2)
##
##  Durbin-Watson test
##
## data:  ecm2
## DW = 2.4973, p-value = 0.942
## alternative hypothesis: true autocorrelation is greater than 0

dwtest(ecm2, alternative = "two.sided")
##
##  Durbin-Watson test
##
## data:  ecm2
## DW = 2.4973, p-value = 0.1161
## alternative hypothesis: true autocorrelation is not 0
```

We can also compare the fitted values of the two models:

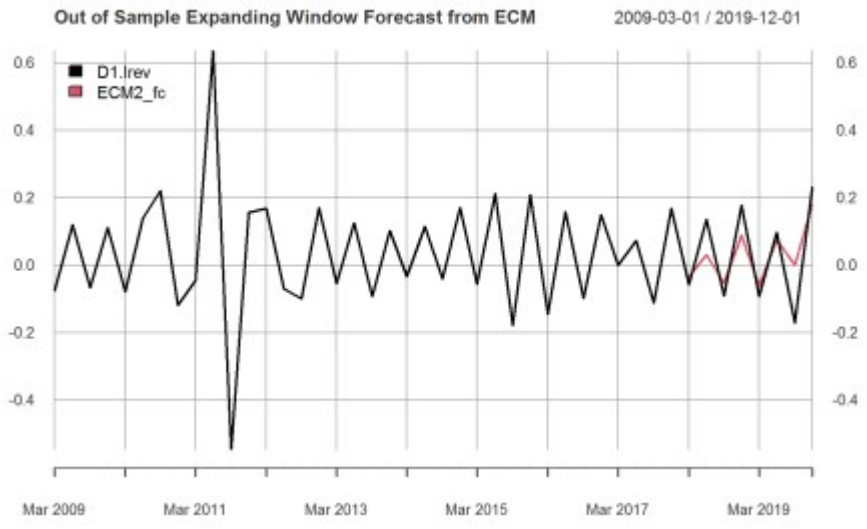
```
# Get ECM fitted values
ECM1_fit <- fitted(ecm)
ECM2_fit <- fitted(ecm2)

# Plot together with revenue
plot(merge(D(X[, "lrev"]), ECM1_fit, ECM2_fit) %>% na_omit,
     main = "Dlog Revenue and ECM Fit",
     legend.loc = "topleft", major.ticks = "years", grid.ticks.on =
"years")
```



Both the fit statistics and fitted values suggest that ECM2 is a feasible forecasting specification that avoids the need to first forecast quarterly GDP.

The true forecasting performance of the model can only be estimated through out of sample forecasts. Below I compute 1 quarter ahead forecasts for quarters 2018Q1 through 2019Q4 using an expanding window where both the cointegration equation and the ECM are re-estimated for each new period.

```
# Function to forecast with expanding window from start year (using
ECM2 specification)
forecast_oos <- function(x, start = 2018) {
  n <- nrow(x[paste0("/", start - 1), ])
  xzoo <- as.zoo(x)
  fc <- numeric(0L)
  # Forecasting with expanding window
  for(i in n:(nrow(x)-1L)) {
    samp <- xzoo[1:i, ]
    ci <- dynlm(lrev ~ lgdp + GDPdum, samp)
    samp <- cbind(samp, res = resid(ci))
    mod <- dynlm(D(lrev) ~ L(D(lrev)) + L(D(lrev), 2L) + L(D(lgdp)) +
L(res) + GDPdum, samp)
    fc <- c(fc, flast(predict(mod, newdata = samp))) # predict does not
re-estimate
  }
  xfc <- cbind(D(x[, "lrev"]), ECM2_fc = NA)
  xfc[(n+1L):nrow(x), "ECM2_fc"] <- unattrib(fc)
  return(xfc)
}

# Forecasting
ECM_oos_fc <- forecast_oos(na_omit(X))

# Plotting
plot(ECM_oos_fc["2009/", ],
     main = "Out of Sample Expanding Window Forecast from ECM",
     legend.loc = "topleft", major.ticks = "years", grid.ticks.on =
"years")
```



The graph suggests that the forecasting performance is quite acceptable. When seasonally adjusting GDP and revenue beforehand, the forecast becomes less accurate, so a part of this fit is accounted for by seasonal patterns in the two series. Finally, we could formally evaluate the forecast computing a sophisticated set of forecast evaluation metrics and also comparing the forecast to a naive forecast provided by the value of revenue in the previous quarter.

```
eval_forecasts <- function(y, fc, add.naive = TRUE, n.ahead = 1) {
  mfc <- eval(substitute(qDF(fc))) # eval substitute to get the name of
the forecast if only a vector is passed
  lagy <- flag(y, n.ahead)
  if (add.naive) mfc <- c(list(Naive = lagy), mfc)
  if (!all(length(y) == lengths(mfc))) stop("All supplied quantities
must be of equal length")
  res <- vapply(mfc, function(fcy) {
    # Preparation
    cc <- complete.cases(y, fcy)
    y <- y[cc]
    fcy <- fcy[cc]
    lagycc <- lagy[cc]
    n <- sum(cc)
    nobessel <- sqrt((n - 1) / n) # Undo bessel correction (n-1)
instead of n in denominator
    sdy <- sd(y) * nobessel
    sdfcy <- sd(fcy) * nobessel
    diff <- fcy - y
    # Calculate Measures
    bias <- sum(diff) / n          # Bias
    MSE <- sum(diff^2) / n         # Mean Squared Error
    BP <- bias^2 / MSE             # Bias Proportion
    VP <- (sdy - sdfcy)^2 / MSE    # Variance Proportion
    CP <- 2 * (1 - cor(y, fcy)) * sdy * sdfcy / MSE # Covariance
Proportion
    RMSE <- sqrt(MSE)              # Root MSE
    R2 <- 1 - MSE / sdy^2          # R-Squared
    SE <- sd(diff) * nobessel      # Standard Forecast Error
    MAE <- sum(abs(diff)) / n      # Mean Absolute Error
    MPE <- sum(diff / y) / n * 100 # Mean Percentage Error
```

```
    MAPE <- sum(abs(diff / y)) / n * 100 # Mean Absolute Percentage
Error
    U1 <- RMSE / (sqrt(sum(y^2) / n) + sqrt(sum(fcy^2) / n))   # Theils
U1
    U2 <- sqrt(mean.default((diff / lagycc)^2, na.rm = TRUE) / # Theils
U2 (= MSE(fc)/MSE(Naive))
                mean.default((y / lagycc - 1)^2, na.rm = TRUE))
    # Output
    return(c(Bias = bias, MSE = MSE, RMSE = RMSE, `R-Squared` = R2, SE
= SE,
      MAE = MAE, MPE = MPE, MAPE = MAPE, U1 = U1, U2 = U2,
      `Bias Prop.` = BP, `Var. Prop.` = VP, `Cov. Prop.` = CP))
  }, numeric(13))
  attr(res, "naive.added") <- add.naive
  attr(res, "n.ahead") <- n.ahead
  attr(res, "call") <- match.call()
  class(res) <- "eval_forecasts"
  return(res)
}

# Print method
print.eval_forecasts <- function(x, digits = 3, ...)
print.table(round(x, digits))

ECM_oos_fc_cc <- na_omit(ECM_oos_fc)
eval_forecasts(ECM_oos_fc_cc[, "D1.lrev"], ECM_oos_fc_cc[, "ECM2_fc"])
##               Naive  ECM2_fc
## Bias         -0.041    0.001
## MSE           0.072    0.005
## RMSE          0.268    0.070
## R-Squared    -2.414    0.748
## SE            0.265    0.070
## MAE           0.260    0.060
## MPE        -194.319   48.495
## MAPE        194.319   62.696
## U1            0.974    0.219
## U2            1.000    0.233
## Bias Prop.    0.024    0.000
## Var. Prop.    0.006    0.248
## Cov. Prop.    0.970    0.752
```

The metrics show that the ECM forecast is clearly better than a naive forecast using the previous quarters value. The bias proportion of the forecast error is 0, but the variance proportion 0.25, suggesting, together with the plot, that the variance of the forecasts is a bit too large compared to the variance of the data.