VECM model

Weiheng Zhang

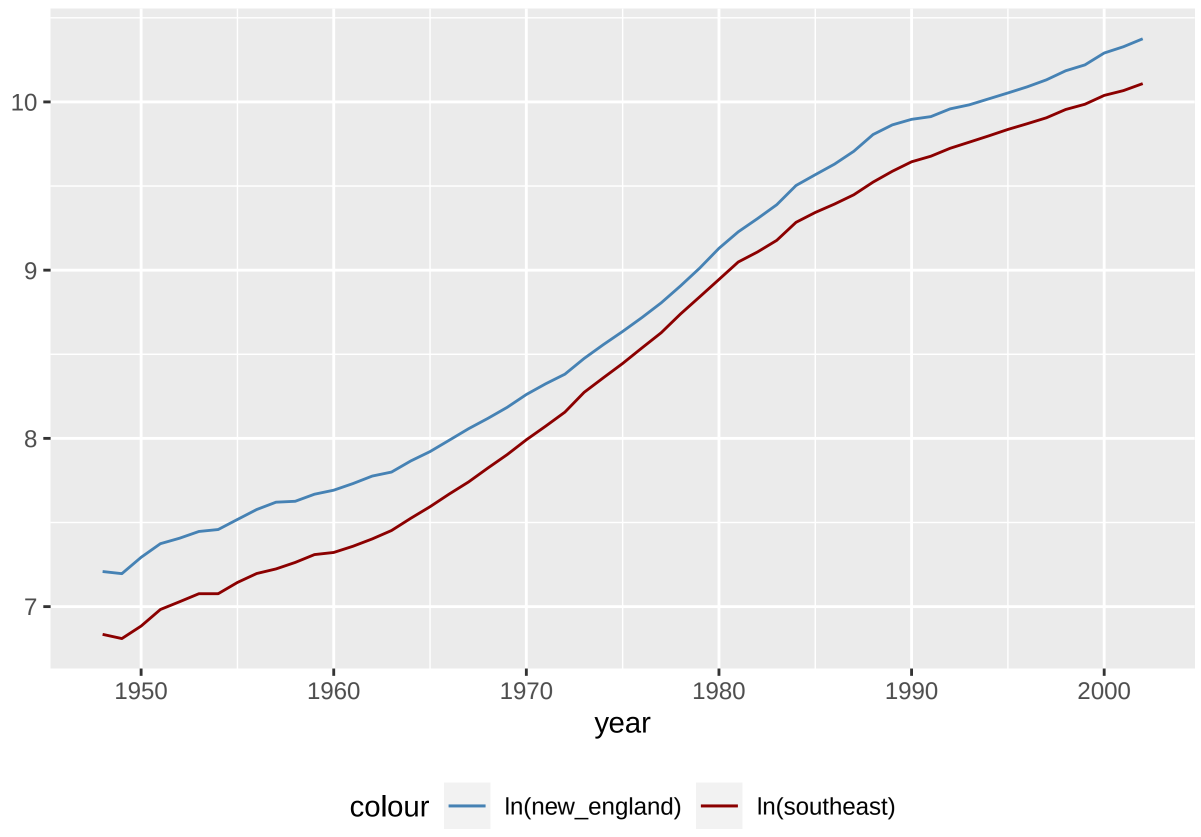
2022-04-25

```r
library(FinMetric)
library(tsDyn)
library(vars)
digits <- function(x, n){
    format(round(x, n), nsmall = n)
}
```

```r
dat <- read_stata("https://www.stata-press.com/data/r17/rdinc")
```

```
## . use "https://www.stata-press.com/data/r17/rdinc"
```

```r
ggplot(dat) +
    geom_line(aes(x = year, y = ln_ne, color = "ln(new_england)")) +
    geom_line(aes(x = year, y = ln_se, color = "ln(southeast)")) +
    scale_color_manual(values = c("ln(new_england)" = "steelblue", "ln(southeast)" = "darkred")) +
    theme(legend.position = "bottom") +
    scale_x_continuous(breaks = seq(1950, 2000, 10)) +
    labs(y = "")
```



The graph indicates a differential between the two series that shrinks between 1960 and about 1980 and then grows until it stabilizes around 1990. We next estimate the parameters of a bivariate VECM with one cointegrating relationship.

```
vec1 <- VECM(dat[, c("ln_ne", "ln_se")], lag = 1, estim = "ML")
# STATA use ML to estimate
summary(vec1)
```

```
## #############
## ###Model VECM
## #############
## Full sample size: 55     End sample size: 53
## Number of variables: 2   Number of estimated slope parameters 8
## AIC -884.0597    BIC -866.3271    SSR 0.03287063
## Cointegrating vector (estimated by ML):
##    ln_ne      ln_se
## r1     1 -0.9433708
##
##
##                  ECT                Intercept         ln_ne -1
## Equation ln_ne -0.4338(0.0721)*** 0.3868(0.0594)*** 0.7169(0.1889)***
## Equation ln_se -0.3544(0.0755)*** 0.3201(0.0622)*** 0.3367(0.1976).
##                  ln_se -1
## Equation ln_ne -0.6749(0.2118)**
## Equation ln_se -0.1606(0.2216)
```

Rather than being covariance stationary, many economic time series appear to be “first-difference stationary”. This means that the level of a time series is not stationary but its first difference is. This kind of series denotes as $I(1)$ processes, while covariance stationary series denotes as $I(0)$. In general, a process whose d-th difference is stationary is an integrated process of order d, or $I(d)$.

Multivariate VECM specification

Consider a VAR with p lags:

$$y_t = v + A_1 y_{t-1} + \cdots + A_p y_{t-p} + \varepsilon_t$$

where y_t is a $1 \times k$ vector.

We can construct following VECM model:

$$\Delta y_t = v + (\sum_1^p A_j - I_k) y_{t-1} + \sum_1^{p-1} (- \sum_{j=i+1}^p A_j) \Delta y_{t-i} + \varepsilon_t$$
$$\Delta y_t = v + \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + \varepsilon_t$$

if p = 2, it will be:

$$y_t - y_{t-1} = v + (A_1 + A_2 - I) y_{t-1} + (-A_2) \Delta y_{t-1} + \varepsilon_t$$

In general we can contain time trend in VECM model:

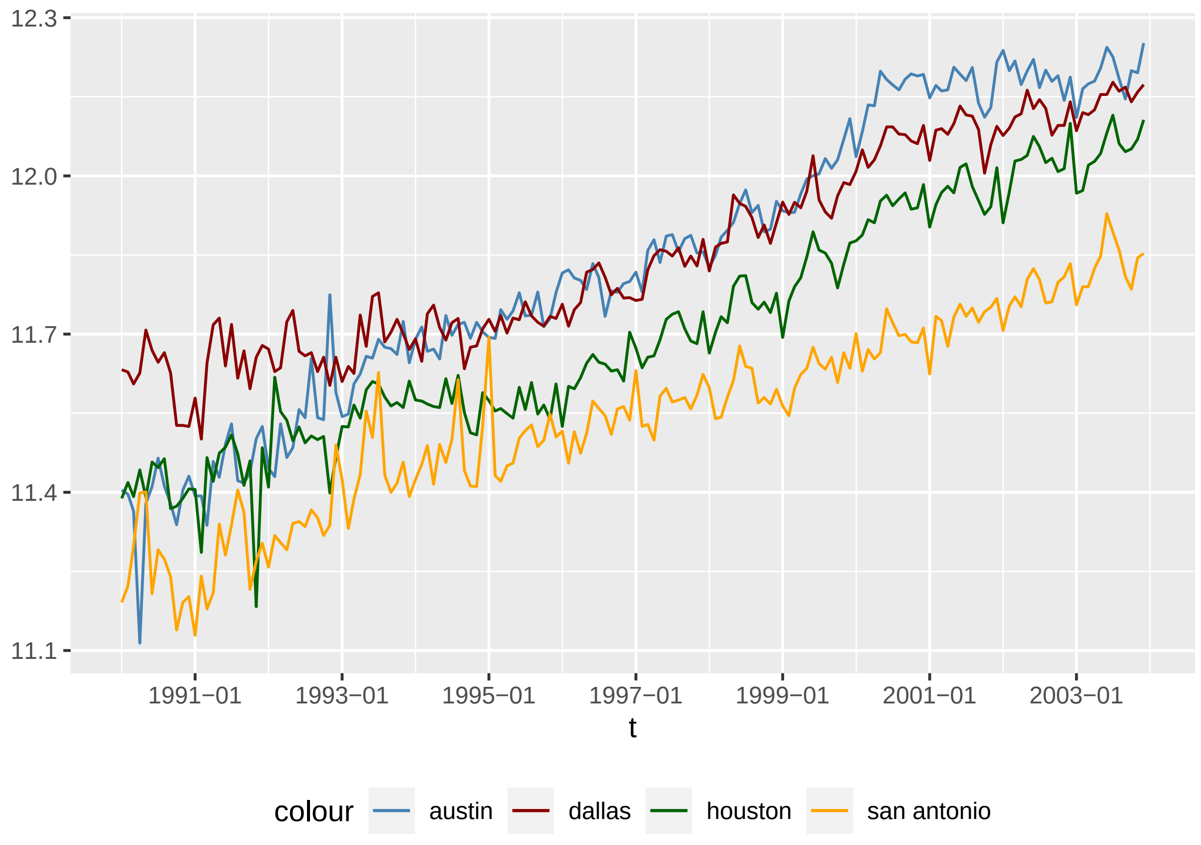
$$\Delta y_t = \alpha(\beta y_{t-1} + \mu + \rho t) + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + \gamma + \tau t + \varepsilon_t$$

- Case 1: Unrestricted trend
  - If no restrictions are placed on the trend parameters, the equation implies that there are quadratic trends in the levels of the variables and that the cointegrating equations are stationary around time trends (trend stationary).
- Case 2: Restricted trend ($\tau = 0$)
  - We assume that the trends in the levels of the data are linear but not quadratic.
- Case 3: Unrestricted constant ($\tau = 0$ and $\rho = 0$)
  - We restrict the cointegrating equation to be stationary around constant means. Because γ is not restricted to zero, this specification still puts a linear time trend int he levels of the data.
- Case 4: Restricted constant ($\tau = 0$, $\rho = 0$ and $\gamma = 0$)
  - We assume there are no linear time trends int he levels of the data. The cointegrating equations to be stationary around a constant mean, but it allows no other trends and constant terms.
- Case 5: No trend: ($\tau = 0$, $\rho = 0$, $\gamma = 0$ and $\mu = 0$)
  - It assumes that the equations are stationary with means of zero and that the differences and the levels of the data have means of zero.

```
dat <- read_stata("https://www.stata-press.com/data/r17/txhprice")
```

```
## . use "https://www.stata-press.com/data/r17/txhprice"
```

```
dat$t <- seq(as.Date("1990-01-01"), as.Date("2003-12-01"), by = "month")
ggplot(dat) +
    geom_line(aes(x = t, y = austin, color = "austin")) +
    geom_line(aes(x = t, y = dallas, color = "dallas")) +
    geom_line(aes(x = t, y = houston, color = "houston")) +
    geom_line(aes(x = t, y = sa, color = "san antonio")) +
    scale_color_manual(values = c("austin" = "steelblue",
                                      "dallas" = "darkred",
                                      "houston" = "darkgreen",
                                      "san antonio" = "orange")) +
    theme(legend.position = "bottom") +
    labs(y = "") +
    scale_x_date(breaks = scales::breaks_width("2 year"), date_labels = "%Y-%m")
```



Here, we focus on the housing prices in Dallas and Houston.

```
y <- dat[, c("dallas", "houston")]
VARselect(y)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n) 
##      9      2      2      9 
## 
## $criteria
##                    1             2             3             4             5
## AIC(n) -1.271026e+01 -1.283318e+01 -1.283064e+01 -1.280257e+01 -1.278342e+01
## HQ(n)  -1.266303e+01 -1.275446e+01 -1.272043e+01 -1.266088e+01 -1.261024e+01
## SC(n)  -1.259396e+01 -1.263935e+01 -1.255927e+01 -1.245367e+01 -1.235699e+01
## FPE(n)  3.020008e-06  2.670783e-06  2.677781e-06  2.754363e-06  2.808184e-06
##                    6             7             8             9            10
## AIC(n) -1.281413e+01 -1.283276e+01 -1.285687e+01 -1.290715e+01 -1.287750e+01
## HQ(n)  -1.260947e+01 -1.259660e+01 -1.258922e+01 -1.260801e+01 -1.254688e+01
## SC(n)  -1.231016e+01 -1.225126e+01 -1.219783e+01 -1.217057e+01 -1.206339e+01
## FPE(n)  2.724055e-06  2.674860e-06  2.612522e-06  2.486071e-06  2.562989e-06
```

Here we will consider 2 lags based on HQ (Hannan–Quinn information criterion) and SC (Schwarz Bayesian information criterion).

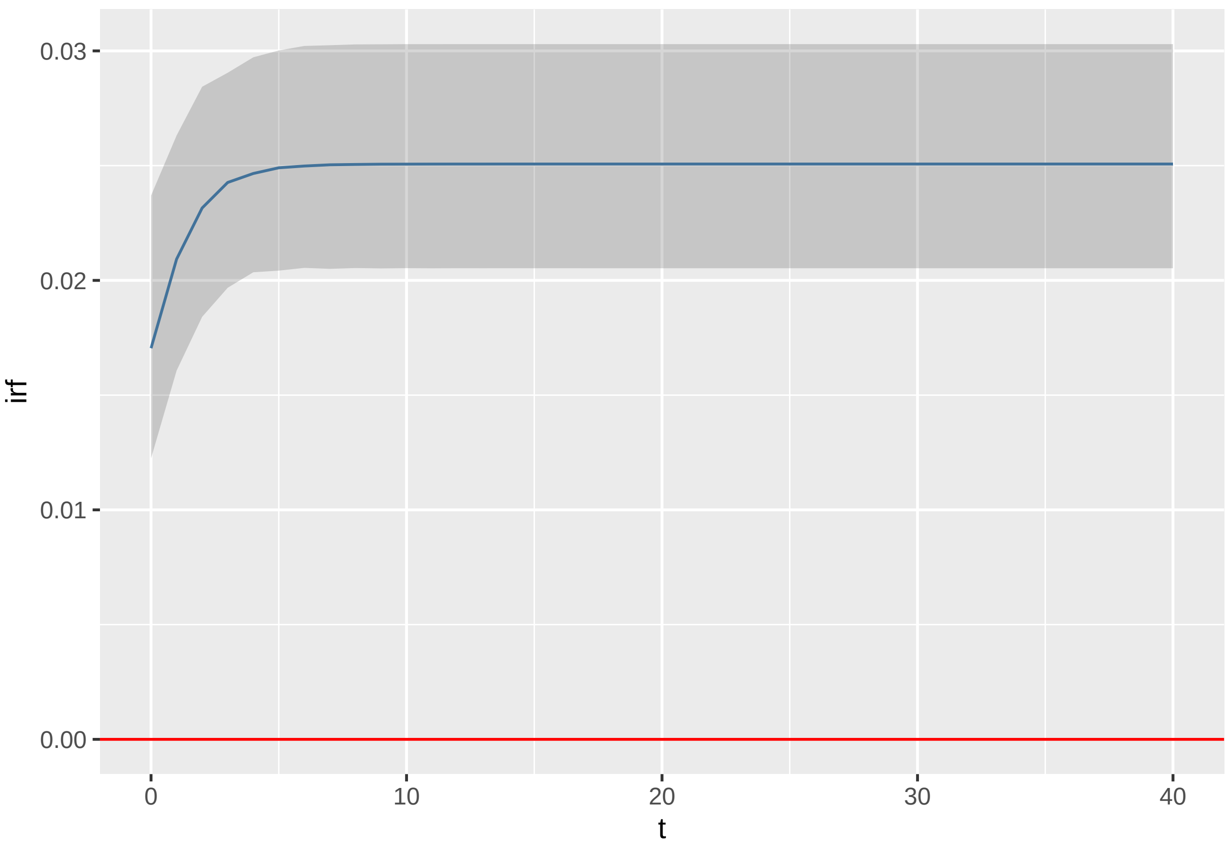
```
urca::ca.jo(y, type = "eigen", K = 2, ecdet = "const") %>%
    summary() # indicate it is a I(1) process
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , without linear trend and constant in cointegra
##
## Eigenvalues (lambda):
## [1]  2.455134e-01  2.028436e-02 -1.056946e-15
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  3.40  7.52  9.24 12.97
## r = 0  | 46.77 13.75 15.67 20.20
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##             dallas.l2 houston.l2   constant
## dallas.l2   1.0000000   1.000000   1.000000
## houston.l2 -0.8677167  -0.380674   1.374091
## constant   -1.6901493  -7.833694 -27.820782
##
## Weights W:
## (This is the loading matrix)
##
##            dallas.l2  houston.l2      constant
## dallas.d  -0.2957905 -0.01083876 -4.929289e-14
## houston.d  0.5106730 -0.01073306  8.079394e-14
```

```
vec1 <- VECM(y, lag = 1, estim = "ML")
summary(vec1)
```

```
## #############
## ###Model VECM
## #############
## Full sample size: 168    End sample size: 166
## Number of variables: 2   Number of estimated slope parameters 8
## AIC -2123.351    BIC -2095.343   SSR 0.5738377
## Cointegrating vector (estimated by ML):
##    dallas    houston
## r1      1 -0.8675936
##
##
##                   ECT                 Intercept          dallas -1
## Equation dallas  -0.3039(0.0909)**  0.5188(0.1539)***  -0.1647(0.0879).
## Equation houston 0.5027(0.1069)***  -0.8456(0.1810)*** -0.0620(0.1035)
##                   houston -1
## Equation dallas  -0.0998(0.0651)
## Equation houston -0.3328(0.0766)***
```

```
irf_vecm <- irf(vec1, boot = T, runs = 100, n.ahead = 40)
# Dallas -> Houston
data_plot <- data.frame(irf = irf_vecm$irf$dallas[, 2],lower = irf_vecm$Lower$dallas[, 2], upper = irf_
ggplot(data_plot, aes(x = t, y = irf, ymin = lower, ymax = upper)) +
    geom_line(color = "steelblue") +
    geom_ribbon(alpha = .2) +
    geom_hline(yintercept = 0, color = "red")
```



Based on the estimation we know that

$$\Delta y_t = \hat{\alpha}(\hat{\beta} y_{t-1}) + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + \hat{v} + \varepsilon_t$$

$\hat{\alpha} =$ (-0.304, 0.503)

$\hat{\beta} =$ (1.000, -0.868)

$\hat{v} =$ (0.519, -0.846)

$\hat{\Gamma} =$

```
digits(vec1$coefficients[, c(3, 4)], 3) %>%
    knitr::kable(col.names = c("dallas", "houston"))
```

	dallas	houston
Equation dallas	-0.165	-0.100
Equation houston	-0.062	-0.333