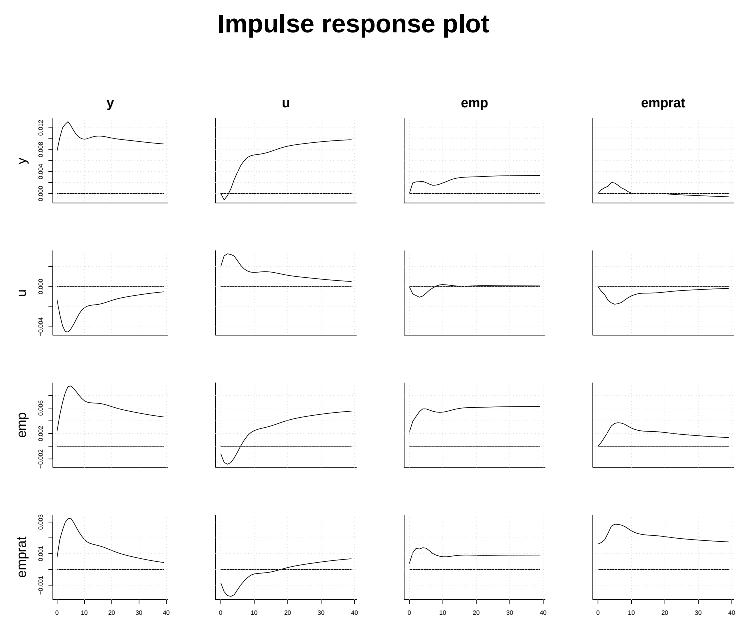
ANSWER FOR VAR EXERCISE

Because I'm trying to get this answer sheet posted in time to be useful in studying for the exam, I include answers for only the first three parts. Part three, in particular, was not done properly in all the answers turned in. Parts 4-7, though they involved a lot of computation, seemed on the basis of our class discussions and presentations to have been done well by most people.

I should have suggested that you scale the employment rate (which is in per cent units) to match the log scale of the output and employment data, i.e. divide employment rate by 100. The unemployment rate was already transformed this way in the R data file on the course web site.

(1) The plotted impulse responses are below.

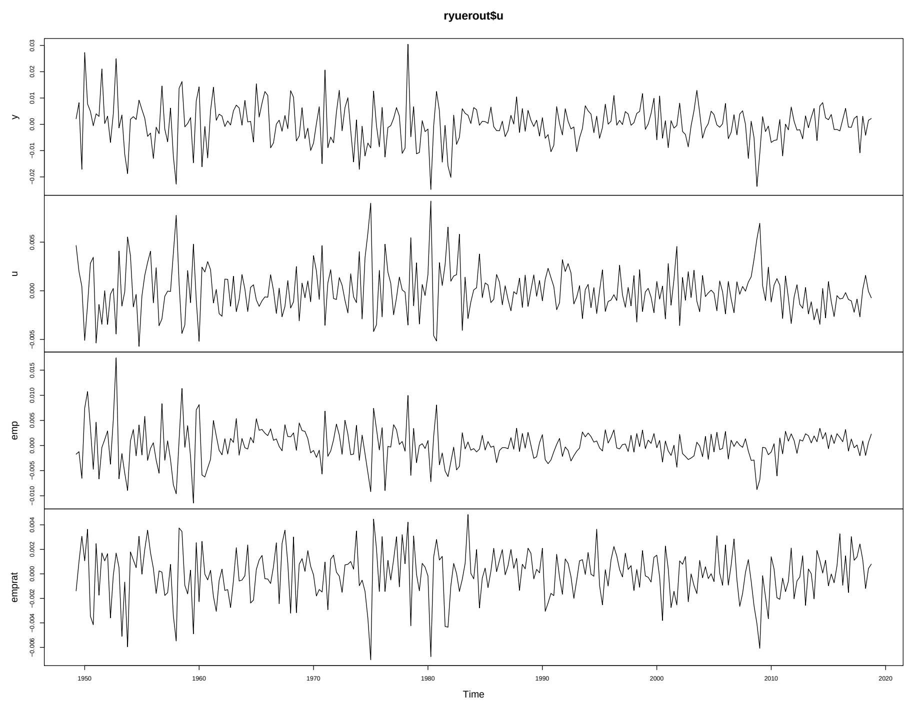


In the plot, columns correspond to shocks, rows correspond to responding variables. As one would expect, most variance in all four variables is accounted for by a shock that moves output up, unemployment down, employment up, and the employment rate up. The responses of output and employment to this shock are very persistent, while the employment and unemployment rates return slowly to their means.

The second shock, corresponding to unemployment, is constrained to be a shock to the part of employment surprise not correlated with out put surprises. That it has no immediate effect on output is therefore not a surprise. But it has a large and persistent *positive* effect on output. This could represent a labor supply effect in part, though that does not account for the initial negative effect of the shock on employment. It could also represent a monetary policy response to a surprise increase in unemployment, though the persistence of the positive effect on output is then hard to explain.

The employment rate has quite a bit of persistence, explained mainly by the fourth shock.

(2) The residual plots are below



All four variables, show reduced variability during 1984-2007, though this is not quite as clear for the employment rate as for the others. Note that the plots appear to run past the end of the sample (which is 2017:III). This is because `rfvar3` includes the dummy observations implementing the persistence priors in the residuals, at the end. That these dummy residuals do not stand out as very small or very large tells us that the prior is neither so weak as to have no influence (as would be implied by tiny values of the dummy residuals) nor so strong as to dominate the data (as would be implied by huge values of the dummy residuals).

(3) There are three eigenvalues within $1/T$ of 1.0 — two complex roots that are almost exactly 1.0 and one real root of .9989. The eigenvectors corresponding to the remaining 17 stable eigenvalues are the last 17 rows of P^{-1}, where P is the matrix of eigenvectors produced as **`eigen`**`(Bmat)`**`$`**`vectors` in R. (R orders the eigenvalues from larges to smallest. Possibly other languages order them differently.). However, P here is near-singular, which could cause numerical problems. So instead

I calculated **eigen**(**t**(Bmat)), where Bmat is again the system matrix. The eigenvector matrix from this has left eigenvectors of Bmat as its columns, and we denote by $\bar{P}$ the last 17 columns of this matrix (the eigenvectors corresponding to the stable roots). The transformed system replaces y_t, the vector of 5 stacked current and lagged values of the four variables in the system, by $z_t = P^{-1}y_t$. We ignore the unstable components of this vector and consider the system $z_t^s = \Lambda \bar{z}_{t-1}^s + \bar{P}'c + \bar{P}'\varepsilon$ satisfied by the stable components $z^s = \bar{P}'y$. (I hope the going back and forth between math and R notation is a net benefit. The actual R code is below.) Remember that the point of transforming from y to z is that the system matrix for z is diagonal with eigenvalues on the diagonal.

We can calculate the mean of z_{it} for stable components as $g_i/(1-\lambda_i)$, where the g vector is $\bar{P}'c$. The covariance matrix for z_t^s is the solution to

$$V = \bar{P}'\Sigma_u\tilde{\bar{P}} + \Lambda V \tilde{\Lambda}\,, \qquad (*)$$

where the $\sim$'s indicate complex conjugation. This is a standard Ricatti equation, made simpler by the system matrix Λ being diagonal. It can be solved by a doubling algorithm, by calling a Lyapunov solver (lyap in matlab), or, because of the diagonality of Λ, analytically. If we denote by Ω (with typical element ω_{ij}) the first matrix on the right of $(*)$, the solution V has typical element

$$\frac{\omega_{ij}}{1-\lambda_i\tilde{\lambda}_j}\,.$$

The ratios of initial deviations from means to unconditional standard deviations of the stable z components then emerge as having absolute values:

```
             [,1]
[1,] 0.04370017
[2,] 1.14621367
[3,] 2.15824827
[4,] 2.15824827
[5,] 1.73436300
[6,] 1.73436300
[7,] 0.86169551
[8,] 0.86169551
[9,] 0.82580840
[10,] 0.82580840
[11,] 1.45454690
[12,] 1.45454690
[13,] 3.48462816
[14,] 3.48462816
[15,] 0.70508182
[16,] 0.70508182
[17,] 0.60213637
```

None of these are outrageously large, but the 13-14 pair might be worth looking at more closely in a real application. They correspond to a complex root with absolute value .58 and period 3.16 quarters, so they are not contributing to a persistent effect of initial conditions through the sample, but it's possible that incorporating into the likelihood a distribution for initial conditions pushing these initial deviations toward zero could affect estimates of short run dynamics.

R code:

```
> ryuerout <- rfvar3(yuer, lags=5)
> Bmat <- sysmat(ryuerout$By)
> eBi <- eigen(t(Bmat))
> Pbar <- eBi$vectors[ , 4:20]
> tsp(ryuerout$u)
[1] 1949.25 2018.75    4.00
> tsp(yuer)
[1] 1948.0 2017.5    4.0
> sigu <- cov(window(ryuerout$u, end=2017.5))
> eBi$values
[1]  1.0000163+0.0000630i  1.0000163-0.0000630i  0.9988548+0.0000000i
[4]  0.9530782+0.0000000i  0.8663778+0.0000000i  0.6963659+0.3726227i
[7]  0.6963659-0.3726227i  0.0402700+0.7259043i  0.0402700-0.7259043i
[10] -0.6425679+0.0607575i -0.6425679-0.0607575i -0.1146596+0.6341691i
[13] -0.1146596-0.6341691i  0.5810619+0.1457320i  0.5810619-0.1457320i
[16] -0.2342709+0.5319853i -0.2342709-0.5319853i -0.0496826+0.3936119i
[19] -0.0496826-0.3936119i -0.3687851+0.0000000i
> sigu <- matrix(0, 20, 20)
> sigu[1:4, 1:4] <- cov(window(ryuerout$u, end=2017.5))
> siguz <- t(Pbar) %*% sigu %*% Conj(Pbar)
> lam <- eB$values[4:20]
> lammat <- 1 / (1 - (lam %o% Conj(lam)))
> vz <- siguz * lammat
> y0 <- c(t(yuer[5:1, ]))
> z0 <- t(Pbar) %*% y0
> zbar <- (t(Pbar) %*% c(ryuerout$Bx, rep(0, 16))) / (1-lam)
> t0 <- (z0 - zbar) / sqrt(diag(vz))
```