Time Series

Hisam Sabouni

12/3/2018

Goal

The primary focus of this chapter is to introduce methods to analyze economic data that evolves over time. We will first cover some basic statistical properties of time-series data then proceed to introduce a variety of econometric methods that can be utilized to estimate relationships in economic variables that can be used for things like forecasting.

Stationarity

A time series $\{r_t\}$ is said to be strictly stationary if all of the moments of the distribution are assumed to be fixed across time. A *weaker* version of this assumption is weak-stationarity, which implies that the first and second moments are fixed across time. That is for all time, (1) $E(r_t) = \mu$, (2)$Var(r_t) = \gamma_0$, (3) $Cov(r_t, r_{t-l}) = \gamma_l$. Weak stationarity implies that if we were to visualize our data over time, we would find the data would have a constant variation around a fixed level. Weak-stationarity is a very common assumption in econometrics. The second assumption of a fixed second moment across time can however be loosened up through use of conditional heteroskedastic models.

Autocorrelation Function (ACF)

Correlation measures the strength of linear dependence between to random variables. Autocorrelation is a measure of linear dependence between r_t and its past values r_{t-l}. Autocorrelation is measured as:

$$\rho_t = \frac{Cov(r_t, r_{t-l})}{\sqrt{Var(r_t)Var(r_{t-l})}} = \frac{Cov(r_t, r_{t-l})}{Var(r_t)} = \frac{\gamma_l}{\gamma_0}$$

In other words, given a sample of returns $\{r_t\}_{t=1}^{T}$ the sample lag-l autocorrelation function is measured as:

$$\hat{\rho}_l = \frac{\sum\limits_{t=l+1}^{T}(r_t - \bar{r})(r_{t-l} - \bar{r})}{\sum\limits_{t=1}^{T}(r_t - \bar{r})^2}$$

Box, Jenkins, and Reinsel (1994) found that $\hat{\rho}_l$ is asymptotically normal with mean zero and variance of $1/T$ for any fixed positive integer l. Furthermore, if r_t is weakly stationary satisfying $r_t = \mu + \sum\limits_{i=0}^{q}\psi_i a_{t-i}$, where $\psi_0 = 1$ and $\{a_j\}$ is a set of i.i.d random variables with mean zero, then $r\hat{h}o_l$ is asymptotically normal with mean zero and variance $(1 + 2\sum\limits_{i=1}^{q}\rho_i^2)/T$. As a result, the significance of a given level of autocorrelation can be tested by forming a t-statistic:

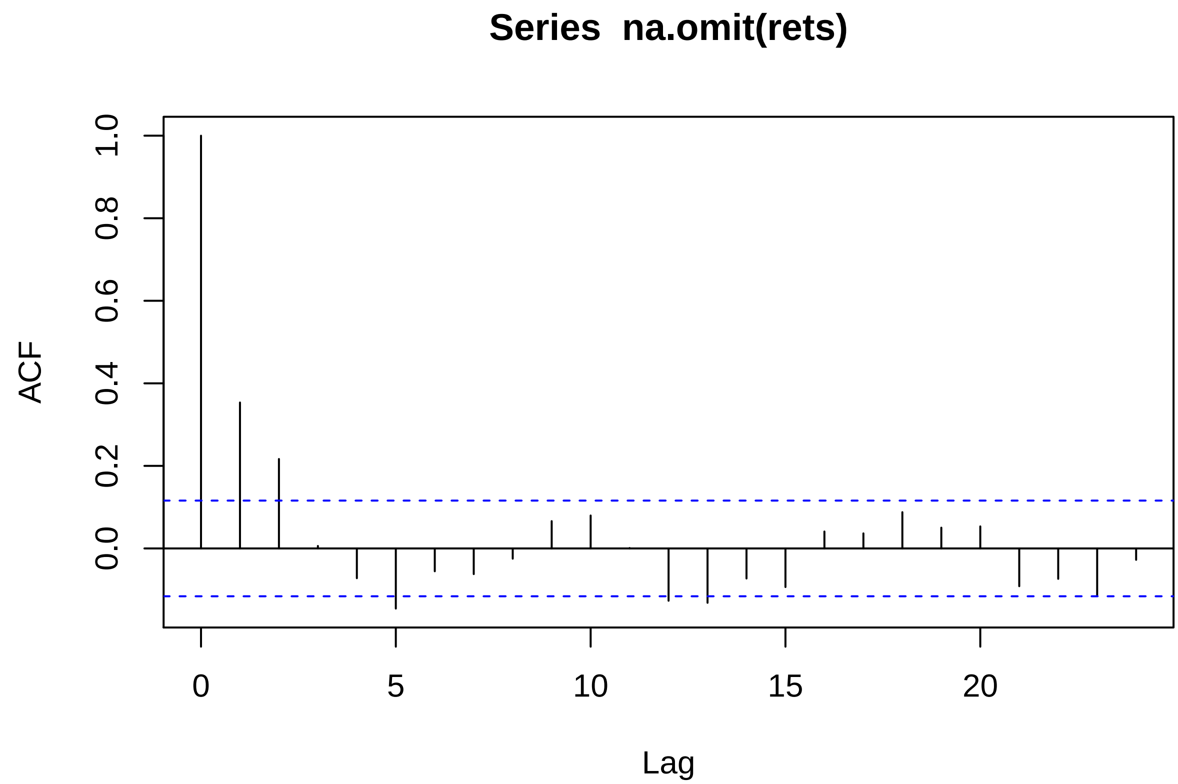
$$t = \frac{\hat{\rho}_l}{(1 + 2\sum\limits_{i=1}^{q}\rho_i^2)/T}$$

R has the ability to estimate the autocorrelation function of a given series through the acf() function.

```r
library(quantmod)
# Real gross domestic product per capita
# (A939RX0Q048SBEA)
getSymbols("A939RX0Q048SBEA", src = "FRED")
```

```
## [1] "A939RX0Q048SBEA"
```

```r
rets <- 400 * Delt(A939RX0Q048SBEA, type = "log")
# Acf
acf(na.omit(rets))
```



Autoregressive Models

If one finds significant autocorrelation levels, it makes sense to leverage this information in a regression based context to generate predictions. An autoregressive model does exactly that:

$$r_t = \phi_0 + \sum_{i=1}^{P} \phi_i r_{t-i} + a_t$$

Here we have an autoregressive model with P lags, AR(P), of the dependent variable as predictors of the next time period, where $a_t \sim N(0, 1)$. Under the assumptions of weak stationarity we can also derive some useful properties about this model. For instance if we take the expectation of the right and left hand side of the above equation we can estimate the mean of the model:

$$E(r_t) = E(\phi_0 + \sum_{i=1}^{P} \phi_i r_{t-i} + a_t)$$

By applying the linearity property of the expectation operator we can break apart the right hand side into separate expectations.

$$E(r_t) = E(\phi_0) + E(\sum_{i=1}^{P} \phi_i r_{t-i}) + E(a_t)$$

By assuming weak stationarity, we have that $E(r_t) = \mu \ \forall t$. Furthermore, by our assumptions $E(a_t) = 0$.

$$\mu = \phi_0 + \sum_{i=1}^{P} \phi_i \mu$$

$$\mu = \frac{\phi_0}{1 - \sum_{i=1}^{P} \phi_i}$$

Notice that this implies that if $\phi_0 = 0 \rightarrow \mu = 0$ and that $\sum_{i=1}^{P} \phi_i \neq 1$. We can also estimate the variance of our model as follows:

$$Var(r_t) = Var(\phi_0 + \sum_{i=1}^{P} \phi_i r_{t-i} + a_t)$$

$$Var(r_t) = Var(\sum_{i=1}^{P} \phi_i r_{t-i}) + Var(a_t)$$

$$Var(r_t) = \sum_{i=1}^{P} \phi_i^2 Var(r_{t-i}) + Var(a_t)$$

$$Var(r_t) - \sum_{i=1}^{P} \phi_i^2 Var(r_{t-i}) = \sigma^2$$

$$Var(r_t) = \frac{\sigma^2}{1 - \sum_{i=1}^{P} \phi_i^2}$$

Here σ^2 is the variance of a_t. Notice that the second moment implies that $\sum_{i=1}^{P} \phi_i^2 < 1$, otherwise we have a negative variance.

Autocorrelation of AR(P) Models (Side-note)

From our autoregressive models we can directly derive our autocorrelation functions. Lets take a simple AR(1) model as an example.

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t$$

If we again, take the expectation of our autogressive model we can solve for ϕ_0:

$$E(r_t) = E(\phi_0) + E(\phi_1 r_{t-1}) + E(a_t)$$

$$\phi_0 = \mu - \phi_1 \mu$$

Now, if we plug this value back into our AR(P) model, we have:

$$r_t = (\mu - \phi_1 \mu) + \phi_1 r_{t-1} + a_t$$

If we re-arrange a few terms, we have:

$$r_t - \mu = \phi_1 (r_{t-1} - \mu) + a_t$$

If we multiply both sides of our equation by $r_{t-l} - \mu$ and take the expectation of both sides we will have a measure of the covariance between time t and time $t - l$.

$$E((r_t - \mu)(r_{t-l} - \mu)) = \phi_1 E((r_{t-1} - \mu)(r_{t-l} - \mu)) + E(a_t (r_{t-l} - \mu))$$

For cases where l is greater than zero, we have that the lag-l autocovariance is:

$$\gamma_l = \phi_1 \gamma_{l-1}$$

For the case where l is exactly zero, we have that

$$\gamma_l = \phi_1 \gamma_{l-1} + \sigma^2$$

In practice

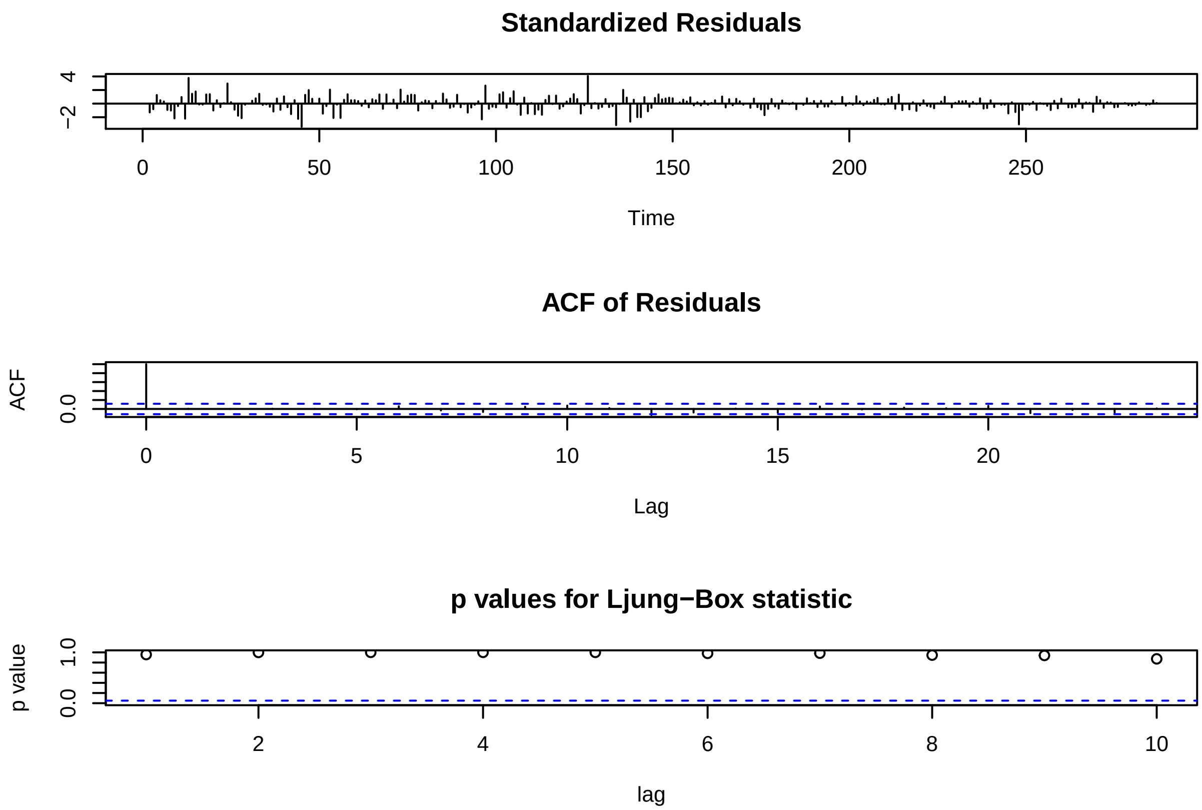
In general, one should determine the optimal number of autoregressive lags to include in their models by analyzing the residuals of their fitted AR(P) model (residuals = $r_t - \hat{r}_t$, where $\hat{r}_t$ is the predicted value of r_t using the AR(P) model). Typically, lags are successively added until there is no significant autocorrelation present in the residuals. Check for significant autocorrelation in the residuals is done through Ljung and Box's (1978) test:

$$Q(m) = T(T+2)\sum_{l=1}^{m} \frac{\hat{\rho}_l^2}{T-l}$$

Here the null hypothesis $H_0 : \rho_1 = ... = \rho_m = 0$ and the alternative hypothesis is $H_a : p_i \neq 0$ for some $i \in \{1, ..., m\}$. Q is asymptotically chi-squared with m degrees of freedom.

We can estimate AR(P) models in R quite easily through the arima() function as well as analyze the residuals through the tsdiag() function.

```
## Estimate arima(x,order=c(AR,Diff,MA))
mdl <- arima(rets, order = c(5, 0, 0))
## Check residuals
tsdiag(mdl)
```




```r
## Predict
predict(mdl, 10)
```

```
## $pred
## Time Series:
## Start = 288
## End = 297
## Frequency = 1
##  [1] 2.427571 2.136830 1.972722 1.762722 1.776554 1.798923 1.858893
##  [8] 1.905119 1.945303 1.957724
##
## $se
## Time Series:
## Start = 288
## End = 297
## Frequency = 1
##  [1] 3.415118 3.579575 3.676935 3.680252 3.681025 3.706130 3.718040
##  [8] 3.725494 3.726704 3.726729
```

```
# coef(mdl)[6]/(1-sum(coef(mdl)[1:5]))
```

Given an AR(P) model we can quite easily generate forecasts as well. For example, if we have an AR(1) model and are at time t wanting to forecast time $t+1$, we can take our estimated model and do as follows:

$$\hat{r}_{t+1} = \hat{\phi}_0 + \hat{\phi}_1 r_t + a_{t+1}$$

If we condition on the information available at time t, denoted by F, then we have:

$$E(\hat{r}_{t+1}|F) = E(\hat{\phi}_0 + \hat{\phi}_1 r_t + a_{t+1}|F)$$

$\hat{\phi}_0$ is a constant, so we can pull that directly out of the conditional expectation operator, $\hat{\phi}_1 r_t$ is readily available as of time t and as a result should be treated as a constant which can also be pulled out of the conditional expectation, a_{t+1} is i.i.d. white noise which is independent of information as at time t (its a *shock* at time t+1). As a result we have:

$$E(\hat{r}_{t+1}|F) = \hat{\phi}_0 + \hat{\phi}_1 r_t + E(a_{t+1})$$

$$E(\hat{r}_{t+1}|F) = \hat{\phi}_0 + \hat{\phi}_1 r_t$$

This same logic can be extended to a multi-step forecast for an AR(P) model:

$$E(\hat{r}_{t+h}|F) = \hat{\phi}_0 + \sum_{i=1}^{P} \hat{\phi}_i \hat{r}_t$$

where $\hat{r}_t$ is previously forecasted return for $h > 0$ and the actual return for $h \leq 0$.

Moving Average Models

In addition to including lags of our dependent variable in our regressions, we can also include lags of previous *shocks*. That is we can include lags of our error term as well:

$$r_t = \phi_0 + \sum_{i=1}^{P} \phi_i r_{t-i} + a_t + \sum_{j=0}^{Q} \theta a_{t-j}$$

Models of this sort are referred to as ARMA(P,Q). The mean of an ARMA(P,Q) is the same as that of an ARMA(P), as a result of $a_t \sim N(0, 1)$. The variance of an ARMA(P,Q) however changes. Lets take an ARMA(1,1) as an example:

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t + \theta a_{t-1}$$

If we multiply both sides by a_t and take the expectation we have:

$$E(a_t r_t) = E(a_t(\phi_0 + \phi_1 r_{t-1} + a_t + \theta a_{t-1})) = \sigma^2$$

If we had taken the variance of the ARMA(1,1) we would have:

$$Var(r_t) = Var(\phi_0 + \phi_1 r_{t-1} + a_t + \theta_1 a_{t-1})$$

$$Var(r_t) = Var(a_t) + Var(\theta_1 a_{t-1} + \phi_1 r_{t-1})$$

$$Var(r_t) = Var(a_t) + Var(\phi_1 r_{t-1}) + Var(\theta_1 a_{t-1}) + 2E(\phi_1(\phi_0 + \phi_1 r_{t-2} + a_{t-1} + \theta_1 a_{t-2})\theta_1 a_{t-1})$$

$$Var(r_t) = Var(a_t) + Var(\phi_1 r_{t-1}) + Var(\theta_1 a_{t-1}) + 2E(\phi_1 a_{t-1}\theta_1 a_{t-1})$$

$$Var(r_t) - \phi_1^2 Var(r_{t-1}) = \sigma^2 + \theta_1 \sigma^2 + 2\phi_1 \theta_1 \sigma^2$$

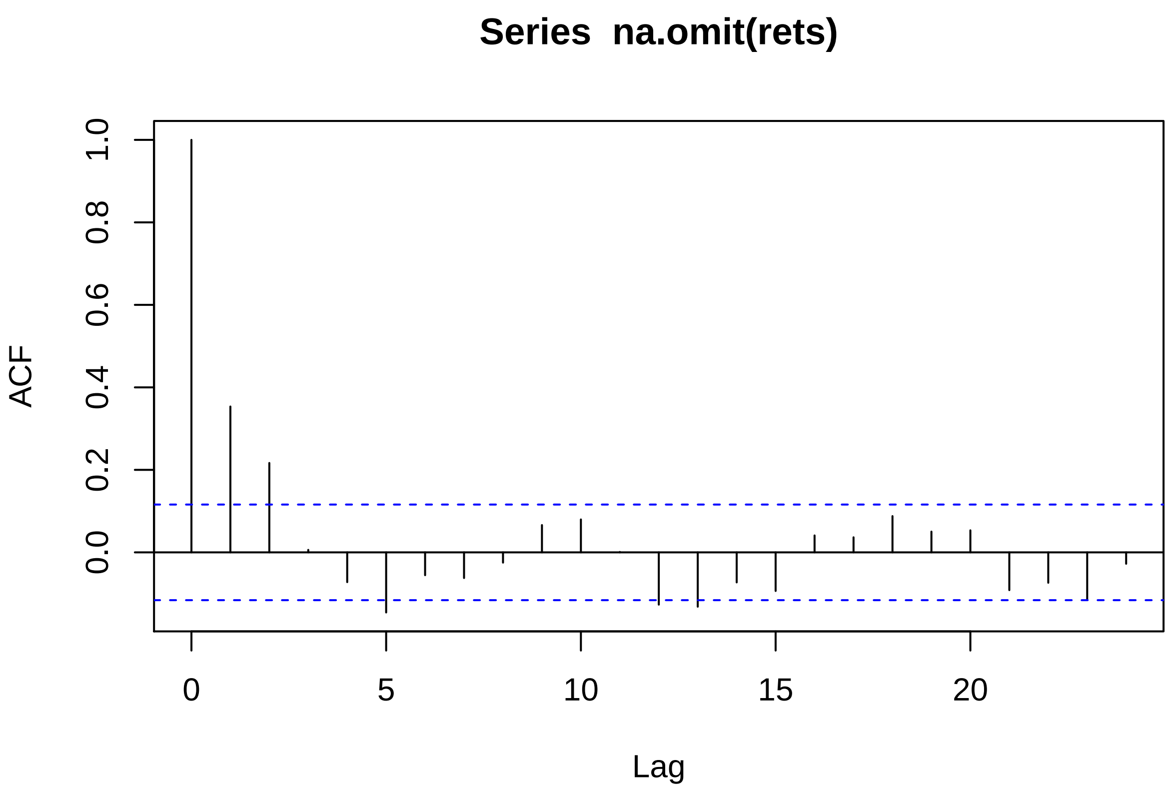
$$Var(r_t) = \frac{\sigma^2(1 + \theta_1 + 2\phi_1\theta_1)}{1 - \phi_1^2}$$

We can generalize the above to an ARMA(P,Q) as:

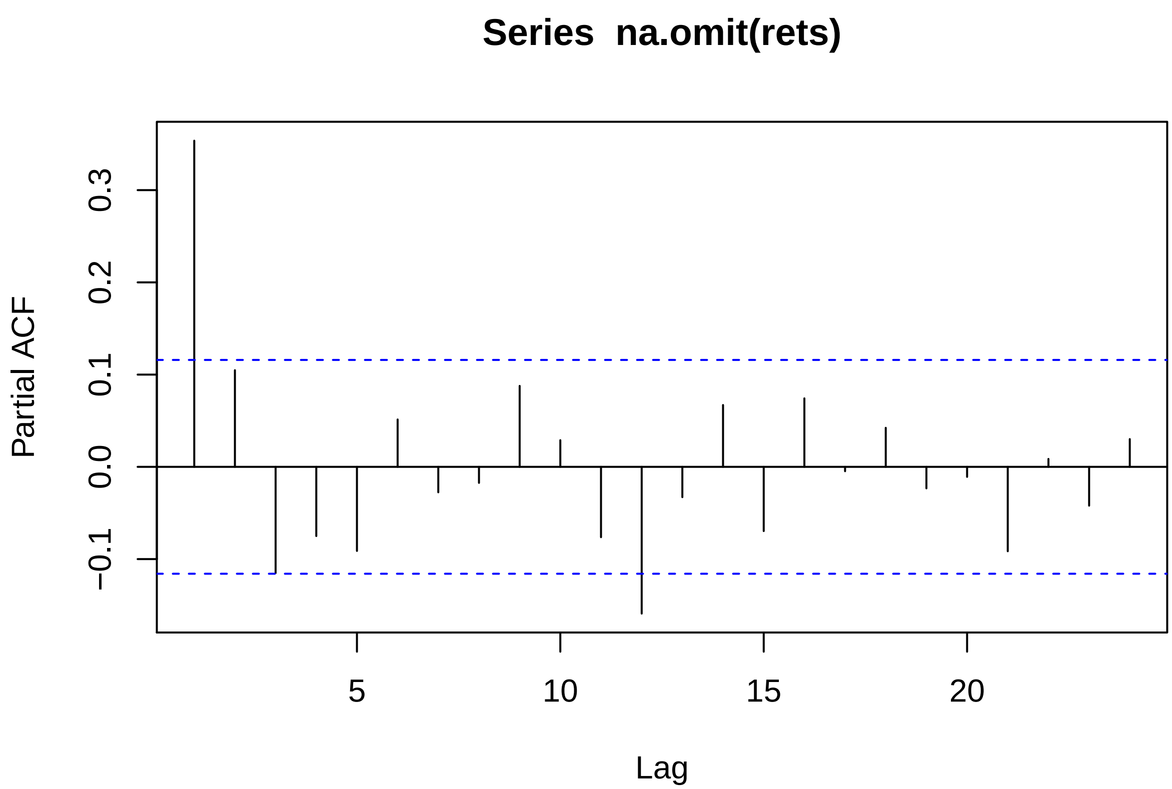
$$Var(r_t) = \frac{\sigma^2(1 + \theta_1 + 2\sum_{k=1}^{M} \phi_k\theta_k)}{1 - \sum_{i=1}^{P} \phi_i^2}$$

where $M = min(P, Q)$. ARMA(P,Q) models are also directly estimatable through the arima() function in R.

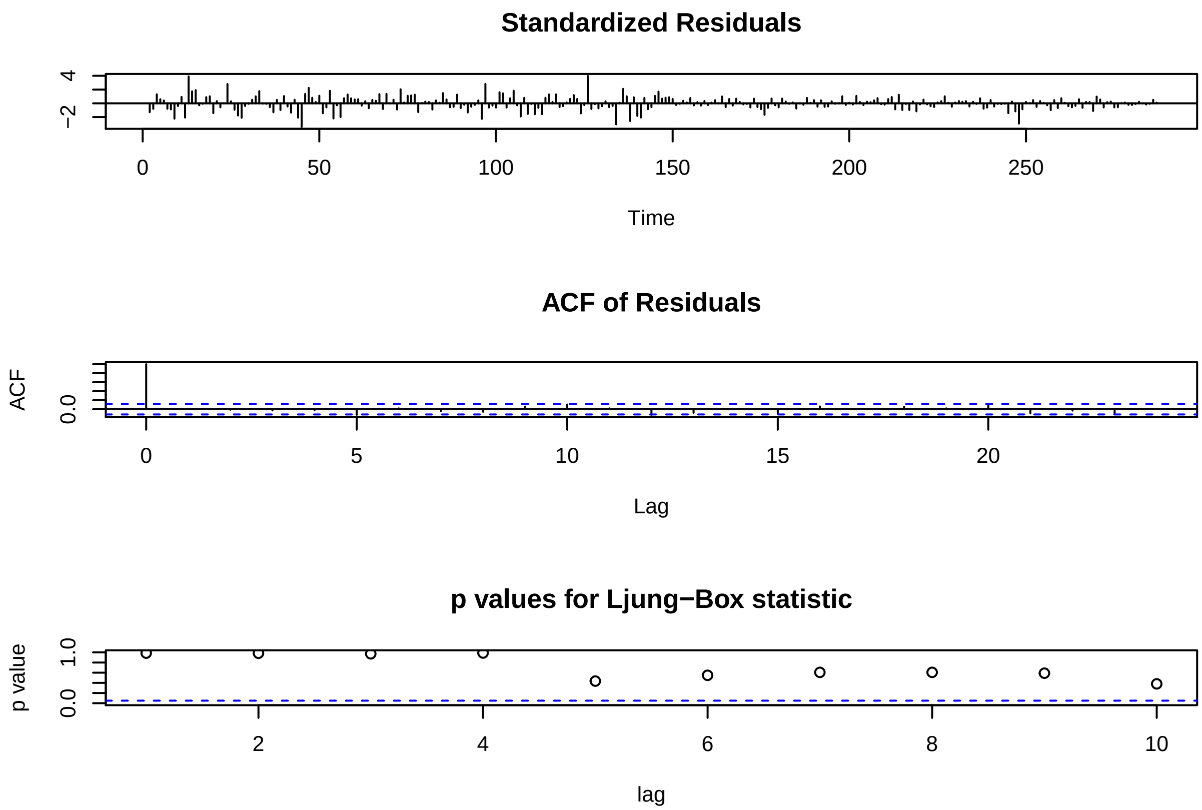
```
## ACF
acf(na.omit(rets))
```




```
pacf(na.omit(rets))
```

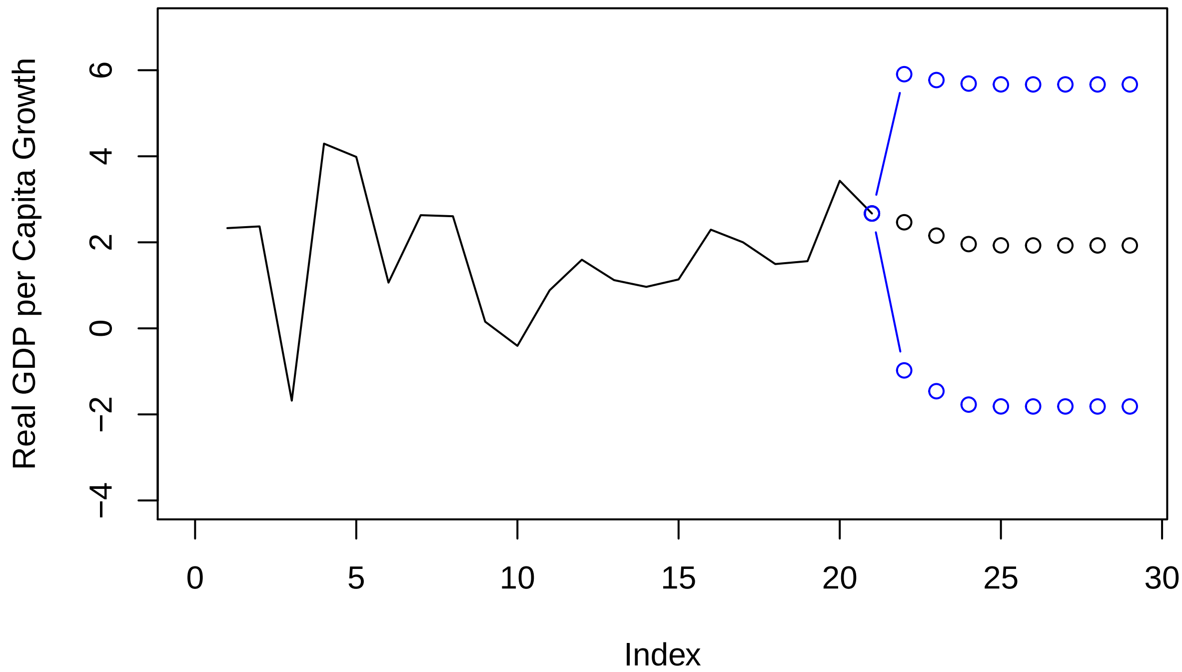



```r
## Arima
mdl <- arima(rets, order = c(1, 0, 3))
## tsdiag to check the residuals
tsdiag(mdl)
```




```
## Make a forecast
forecast <- predict(mdl, 8)

plot(c(as.numeric(rets)[(nrow(rets) - 20):nrow(rets)]),
    xlim = c(0, 29), typ = "l", ylim = c(-4, 7),
    ylab = "Real GDP per Capita Growth")
lines(21:29, c(as.numeric(rets)[nrow(rets)], forecast$pred),
    typ = "b")
lines(21:29, c(as.numeric(rets)[nrow(rets)], forecast$pred +
    forecast$se), typ = "b", col = 4)
lines(21:29, c(as.numeric(rets)[nrow(rets)], forecast$pred -
    forecast$se), typ = "b", col = 4)
```



VAR

A VAR is a n-equation, n-variable linear model in which each variable is in turn explained by its own lagged values, plus current and past values of the remaining n-1 variables initially proposed by Sims (1980). For example, a lag-2 model of two variables may take the form of:

$$X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \beta_3 Y_{t-1} + \beta_4 Y_{t-2} + \zeta_t$$

$$Y_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \alpha_3 Y_{t-1} + \alpha_4 Y_{t-2} + \eta_t$$

Each equation in the system can be estimated by ordinary least squares (OLS) regression. With the estimated models in hand one can simulate a shock to one variable and see how future values of other variables in the system are impacted. VARs are extremely useful in the real world, given their easy of interpretation and implementation. Typically, the number of lags selected in the model, if not guided by some economic theory, is chosen to remove autocorrelation from the residuals of the model and by some criterion. Akaike's information criterion (AIC) or Schwarz's Bayseian information criterion are usually used:

$$AIC = log\hat{\sigma}^2 + 2\frac{m * p + 1}{T}$$

$$BIC = log\hat{\sigma}^2 + \frac{m * p + 1}{T}log(T)$$

The model with the smallest AIC or BIC value is preferred.

```
library(quantmod)
library(vars)
getSymbols(c("GDPCTPI", "UNRATE", "DGS10", "A939RX0Q048SBEA"),
    src = "FRED")
```

```
## [1] "GDPCTPI"         "UNRATE"          "DGS10"           "A939RX0Q048SBEA"
```

```
# Ordering of variables matters for impulse
# response functions
macro_data <- merge(UNRATE, GDPCTPI, A939RX0Q048SBEA,
    DGS10)
macro_data <- na.omit(macro_data)
colnames(macro_data) <- c("unemployment", "gdp_deflator",
    "gdp", "ten_year")
macro_data$gdp_deflator <- 400 * Delt(macro_data$gdp_deflator,
    type = "log")
macro_data$gdp <- 400 * Delt(macro_data$gdp, typ = "log")
macro_data <- na.omit(macro_data)
macro_data <- macro_data[index(macro_data) >=
    "1960-01-01", ]
VARselect(macro_data)  #Information Criterion suggests 1, but 2 is used to remove auto
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $criteria
##                1          2          3          4          5          6
## AIC(n)  5.440990   5.443150   5.638839   5.536678   5.537847   5.704471
## HQ(n)   5.644670   5.809775   6.168409   6.229191   6.393305   6.722873
```

```
## SC(n)    5.943526   6.347714   6.945432   7.245299   7.648497   8.217149
## FPE(n) 230.735078 231.548433 282.535266 256.685313 259.585050 311.273209
##                 7          8          9         10
## AIC(n)   5.860540   5.824404   5.717179   5.470703
## HQ(n)    7.041887   7.168695   7.224415   7.140882
## SC(n)    8.775247   9.141140   9.435943   9.591495
## FPE(n) 371.570441 368.667914 343.668985 281.615945
```

```r
macro_var <- VAR(macro_data, p = 2)
summary(macro_var)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: unemployment, gdp_deflator, gdp, ten_year
## Deterministic variables: const
## Sample size: 114
## Log Likelihood: -910.201
## Roots of the characteristic polynomial:
## 0.9714 0.8527 0.5736 0.5736 0.3727 0.3727 0.2088 0.2088
## Call:
## VAR(y = macro_data, p = 2)
##
##
## Estimation results for equation unemployment:
## =============================================
## unemployment = unemployment.l1 + gdp_deflator.l1 + gdp.l1 + ten_year.l1 + unemploymen
##
##                   Estimate Std. Error t value Pr(>|t|)
## unemployment.l1  0.9904045  0.1213601   8.161 7.76e-13 ***
## gdp_deflator.l1  0.0313841  0.0076419   4.107 7.96e-05 ***
## gdp.l1          -0.0447136  0.0128288  -3.485 0.000718 ***
## ten_year.l1      0.0911875  0.0590847   1.543 0.125758
## unemployment.l2 -0.0976553  0.1161311  -0.841 0.402311
## gdp_deflator.l2 -0.0005152  0.0083460  -0.062 0.950897
## gdp.l2          -0.0150435  0.0104047  -1.446 0.151201
## ten_year.l2     -0.0734939  0.0575875  -1.276 0.204696
```

```
## const            0.5777005  0.2123918   2.720 0.007642 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.502 on 105 degrees of freedom
## Multiple R-Squared: 0.9151,  Adjusted R-squared: 0.9086
## F-statistic: 141.5 on 8 and 105 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gdp_deflator:
## ================================================
## gdp_deflator = unemployment.l1 + gdp_deflator.l1 + gdp.l1 + ten_year.l1 + unemploymen
##
##                 Estimate Std. Error t value Pr(>|t|)
## unemployment.l1 -3.74909    2.06486  -1.816   0.0723 .
## gdp_deflator.l1  0.20674    0.13002   1.590   0.1148
## gdp.l1          -0.20719    0.21827  -0.949   0.3447
## ten_year.l1     -0.04400    1.00529  -0.044   0.9652
## unemployment.l2  3.40822    1.97589   1.725   0.0875 .
## gdp_deflator.l2  0.06344    0.14200   0.447   0.6560
## gdp.l2          -0.04151    0.17703  -0.234   0.8151
## ten_year.l2      0.67027    0.97981   0.684   0.4954
## const            3.72256    3.61371   1.030   0.3053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 8.541 on 105 degrees of freedom
## Multiple R-Squared: 0.1163,  Adjusted R-squared: 0.04901
## F-statistic: 1.728 on 8 and 105 DF,  p-value: 0.1003
##
##
## Estimation results for equation gdp:
## ====================================
## gdp = unemployment.l1 + gdp_deflator.l1 + gdp.l1 + ten_year.l1 + unemployment.l2 + gd
```

```
##
##                  Estimate Std. Error t value Pr(>|t|)
## unemployment.l1 -0.89233    1.41331  -0.631   0.5292
## gdp_deflator.l1 -0.24620    0.08899  -2.766   0.0067 **
## gdp.l1           0.20670    0.14940   1.384   0.1694
## ten_year.l1     -1.01477    0.68808  -1.475   0.1433
## unemployment.l2  1.06276    1.35242   0.786   0.4337
## gdp_deflator.l2 -0.07325    0.09719  -0.754   0.4527
## gdp.l2           0.08297    0.12117   0.685   0.4950
## ten_year.l2      1.35901    0.67064   2.026   0.0453 *
## const            1.28237    2.47343   0.518   0.6052
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 5.846 on 105 degrees of freedom
## Multiple R-Squared: 0.1893,  Adjusted R-squared: 0.1276
## F-statistic: 3.066 on 8 and 105 DF,  p-value: 0.003838
##
##
## Estimation results for equation ten_year:
## =========================================
## ten_year = unemployment.l1 + gdp_deflator.l1 + gdp.l1 + ten_year.l1 + unemployment.l2
##
##                   Estimate Std. Error t value Pr(>|t|)
## unemployment.l1 -0.077489   0.205154  -0.378   0.7064
## gdp_deflator.l1  0.030271   0.012918   2.343   0.0210 *
## gdp.l1           0.022694   0.021686   1.046   0.2978
## ten_year.l1      0.718429   0.099880   7.193 9.75e-11 ***
## unemployment.l2  0.030776   0.196315   0.157   0.8757
## gdp_deflator.l2  0.007387   0.014109   0.524   0.6016
## gdp.l2          -0.023325   0.017589  -1.326   0.1877
## ten_year.l2      0.230351   0.097349   2.366   0.0198 *
## const            0.352119   0.359040   0.981   0.3290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##
## Residual standard error: 0.8486 on 105 degrees of freedom
## Multiple R-Squared: 0.9277,  Adjusted R-squared: 0.9222
## F-statistic: 168.5 on 8 and 105 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##              unemployment gdp_deflator     gdp ten_year
## unemployment       0.2520       -1.061 -1.7209  -0.1124
## gdp_deflator      -1.0610       72.956 32.2260   2.1847
## gdp               -1.7209       32.226 34.1786   0.9369
## ten_year          -0.1124        2.185  0.9369   0.7202
##
## Correlation matrix of residuals:
##              unemployment gdp_deflator     gdp ten_year
## unemployment       1.0000      -0.2475 -0.5863  -0.2639
## gdp_deflator      -0.2475       1.0000  0.6454   0.3014
## gdp               -0.5863       0.6454  1.0000   0.1888
## ten_year          -0.2639       0.3014  0.1888   1.0000
```

```r
Box.test(macro_var$varresult$gdp_deflator$residuals,
    lag = 8)
```

```
##
##  Box-Pierce test
##
## data:  macro_var$varresult$gdp_deflator$residuals
## X-squared = 1.6549, df = 8, p-value = 0.9898
```

```r
Box.test(macro_var$varresult$unemployment$residuals,
    lag = 8)
```

```
##
##  Box-Pierce test
##
## data:  macro_var$varresult$unemployment$residuals
```

```
## X-squared = 7.3406, df = 8, p-value = 0.5004
```

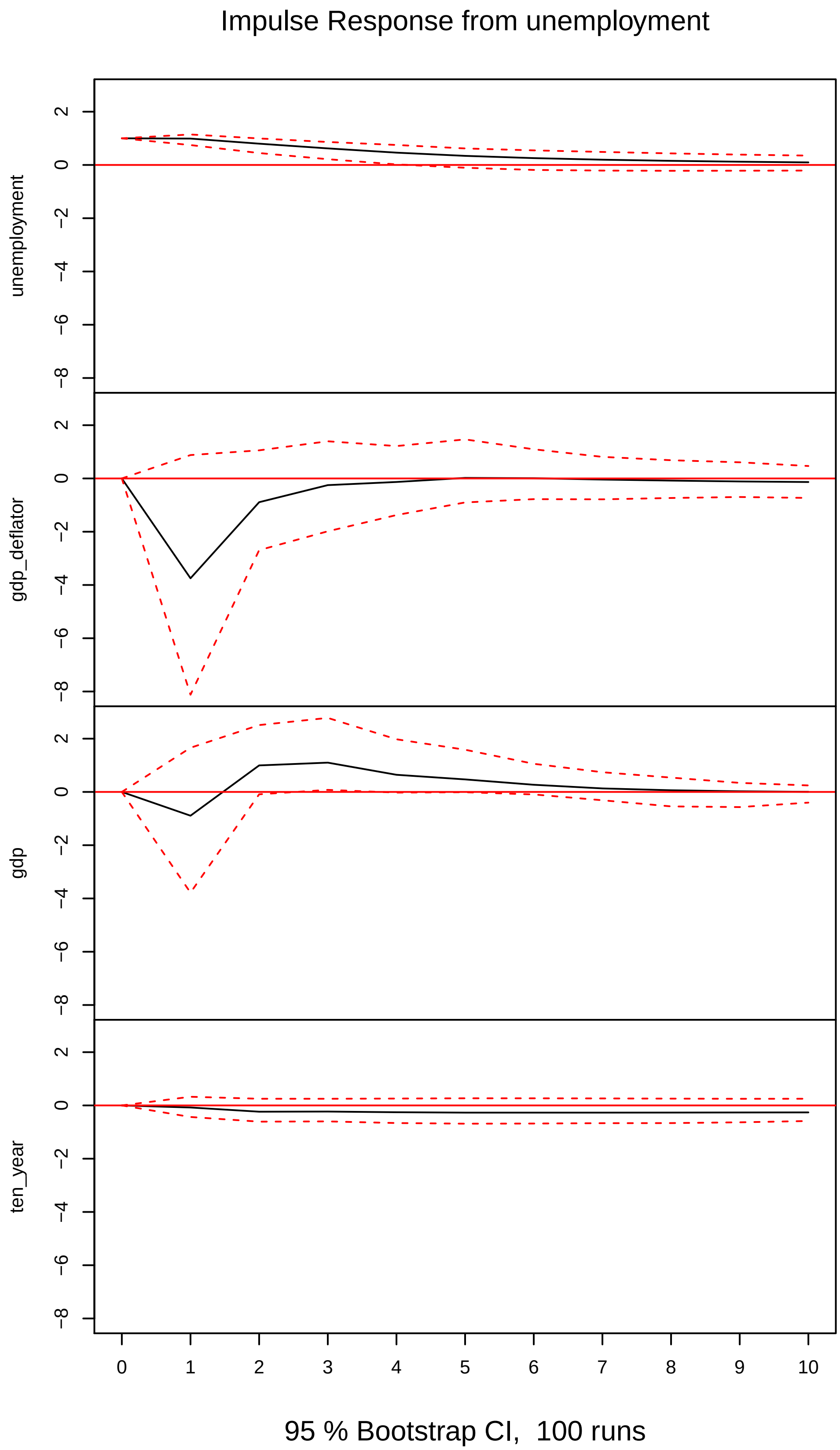
```
Box.test(macro_var$varresult$ten_year$residuals,
    lag = 8)
```

```
##
##  Box-Pierce test
##
## data:  macro_var$varresult$ten_year$residuals
## X-squared = 1.6309, df = 8, p-value = 0.9903
```

```
Box.test(macro_var$varresult$gdp$residuals, lag = 8)
```

```
##
##  Box-Pierce test
##
## data:  macro_var$varresult$gdp$residuals
## X-squared = 11.689, df = 8, p-value = 0.1656
```

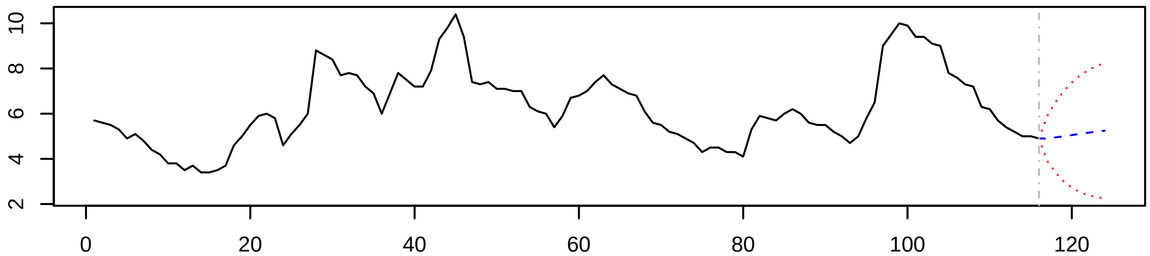
```
# Expected dynamics?
plot(irf(macro_var, impulse = "unemployment",
    boot = 500, ortho = F))
```



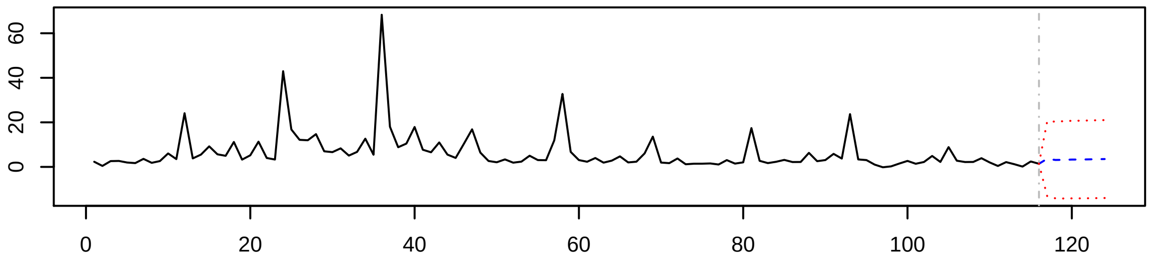
Impulse Response from unemployment
unemployment
gdp_deflator
gdp
ten_year
2
0
−2
−4
−6
−8
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs

```r
# Forecast all variables in our system for 8
# quarters ahead
forecast_macro <- predict(macro_var, n.ahead = 8)
plot(forecast_macro)
```

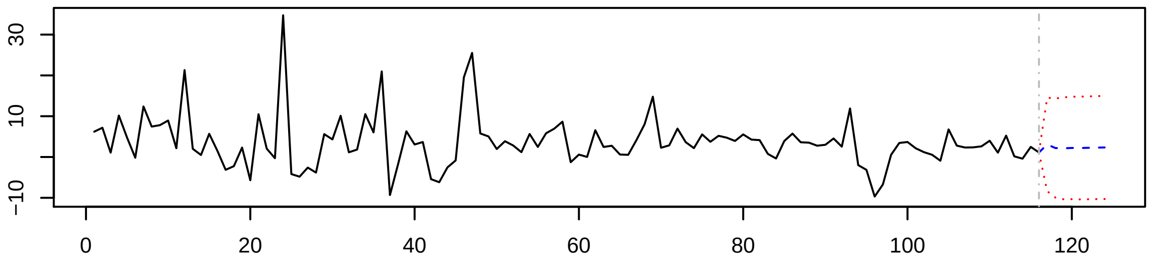
Forecast of series unemployment



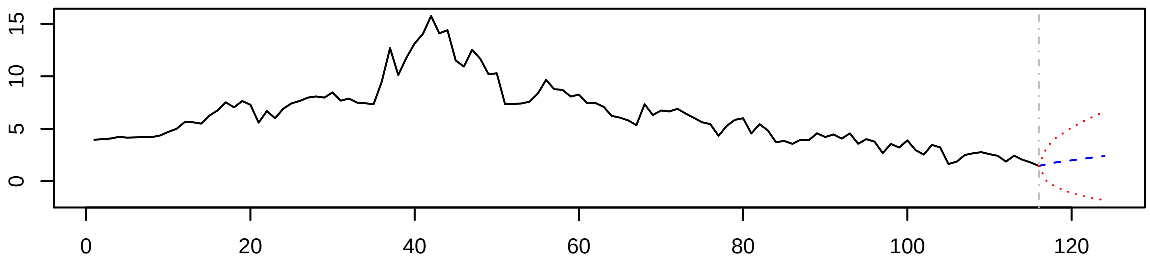
Forecast of series gdp_deflator



Forecast of series gdp



Forecast of series ten_year



Forecast Evaluation

Denote forecast errors as $e_{T+h} = y_{T+h} - \hat{y}_{T+h}$. The most commonly used criteria for forecast evaluation are mean absolute deviation:

$$MAD = \frac{1}{H}\sum_{h=1}^{H} |e_{T+h}|$$

and the root mean squared error:

$$RMSE = \sqrt{\frac{1}{H}\sum_{h=1}^{H} e_{T+h}^2}$$

The lower these measures the more accurate the forecasts. Lets see how our VAR model has performed historically.

```
performance_tracker_mad <- matrix(NA, 20, 4)
performance_tracker_rmse <- matrix(NA, 20, 4)
performance_tracker_in_range <- matrix(NA, 20,
    4)
steps_ahead <- 8
for (i in 89:(nrow(macro_data) - steps_ahead)) {
    macro_var <- VAR(macro_data[1:i], p = 2)
    forecast_macro <- predict(macro_var, n.ahead = steps_ahead)
    actual_data <- macro_data[(i + 1):(i + steps_ahead),
        ]
    for (j in 1:length(forecast_macro$fcst)) {
        # names(forecast_macro$fcst) #When j = 1, we
        # are getting unemployment data Mean absolute
        # error#
        performance_tracker_mad[i - 88, j] <- mean(abs(actual_data[,
            j] - forecast_macro$fcst[[j]][, 1]))
        # Root mean squared error#
        performance_tracker_rmse[i - 88, j] <- sqrt(mean((actual_data[,
            j] - forecast_macro$fcst[[j]][, 1])^2))
        # Check the percentage of actual values within
```

```r
        # our confidence interval#
        performance_tracker_in_range[i - 88, j] <- mean(ifelse(forecast_macro$fcst[[j]][,
            2] <= actual_data[, j] & actual_data[,
            j] <= forecast_macro$fcst[[j]][, 3],
            1, 0))
    }
}
colnames(performance_tracker_mad) <- colnames(macro_data)
print("------Mean Absolute Deviation------")
```

```
## [1] "------Mean Absolute Deviation------"
```

```r
summary(performance_tracker_mad)
```

```
##   unemployment     gdp_deflator        gdp            ten_year
##  Min.   :0.3989   Min.   :2.596   Min.   :1.601   Min.   :0.4995
##  1st Qu.:0.8004   1st Qu.:3.307   1st Qu.:2.047   1st Qu.:0.8342
##  Median :1.3633   Median :3.692   Median :4.493   Median :1.2375
##  Mean   :1.6943   Mean   :4.128   Mean   :4.063   Mean   :1.1928
##  3rd Qu.:2.6380   3rd Qu.:5.208   3rd Qu.:5.224   3rd Qu.:1.3740
##  Max.   :3.7432   Max.   :6.260   Max.   :6.564   Max.   :2.3948
```

```r
colnames(performance_tracker_rmse) <- colnames(macro_data)
print("------Root Mean Squared Error------")
```

```
## [1] "------Root Mean Squared Error------"
```

```r
summary(performance_tracker_rmse)
```

```
##   unemployment     gdp_deflator        gdp            ten_year
##  Min.   :0.4835   Min.   :2.769   Min.   :1.818   Min.   :0.5551
##  1st Qu.:0.9776   1st Qu.:3.547   1st Qu.:2.317   1st Qu.:1.0251
##  Median :1.4522   Median :3.985   Median :5.127   Median :1.4376
##  Mean   :1.9442   Mean   :4.624   Mean   :4.861   Mean   :1.3579
##  3rd Qu.:2.9182   3rd Qu.:5.432   3rd Qu.:6.863   3rd Qu.:1.5071
##  Max.   :4.0283   Max.   :7.612   Max.   :7.608   Max.   :2.4705
```

```r
colnames(performance_tracker_in_range) <- colnames(macro_data)
print("------Confidence Interval Check------")
```

```
## [1] "------Confidence Interval Check------"
```

```r
summary(performance_tracker_in_range)
```

```
##   unemployment     gdp_deflator      gdp             ten_year
##  Min.   :0.0000   Min.   :1     Min.   :0.8750   Min.   :0.8750
##  1st Qu.:0.4688   1st Qu.:1     1st Qu.:0.9688   1st Qu.:1.0000
##  Median :0.8750   Median :1     Median :1.0000   Median :1.0000
##  Mean   :0.7125   Mean   :1     Mean   :0.9688   Mean   :0.9938
##  3rd Qu.:1.0000   3rd Qu.:1     3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1     Max.   :1.0000   Max.   :1.0000
```