Lecture 10

Forecasting and Fitting ARIMA Models

2/20/2018

Forecasting

Forecasting ARMA

- Forecasts for stationary models necessarily revert to mean
  - Remember, $E(y_t) \neq \delta$ but rather $\delta/(1 - \sum_{i=1}^{p} \phi_i)$.
  - Differenced models revert to trend (usually a line)
  - Why? AR gradually damp out, MA terms disappear
- Like any other model, accuracy decreases as we extrapolate / prediction interval increases

Take a fitted ARMA(1,1) process where we know both δ, ϕ, and θ then

$$E(Y_{t+1} \mid Y_t\ Y_{t-1} \ldots Y_1 \ldots) = \hat{Y}_{t+1}$$

$$E(v_n \mid Y_t \ldots Y_1 \ldots) = \begin{cases} 0 & n > t \\ v_n & n \leq t \end{cases}$$

$$Y_t = \delta + \phi_1 Y_{t-1} + \theta v_{t-1}\ v_t$$

One step ahead forecasting

Take a fitted ARMA(1,1) process where we know both δ, ϕ, and θ then

$$\hat{Y}_{t+1} = E(\delta + \phi Y_t + \theta v_t + v_{t+1} \mid \cdot)$$

$$= \delta + \phi Y_t + \theta v_t + 0$$

$$\hat{Y}_{t+2} = E(\delta + \phi Y_{t+1} + \theta v_{t+1} + v_{t+2} \mid \cdot)$$

$$= \delta + \phi \hat{Y}_{t+1} + 0 + 0$$

$$= \delta + \phi(\delta + \phi Y_t + \theta v_t) = \delta + \phi\delta + \phi^2 Y_t + \phi\theta v_t$$

$$\hat{Y}_{t+h} = \delta + \phi\delta + \dots + \phi^{h-1}\delta + \cancelto{0}{\phi^h Y_t} + \cancelto{0}{\phi^{h-1}\theta v_t} = \frac{\delta}{1-\phi}$$

ARIMA(3,1,1) example

$$\phi_p(L)\,(1-L)^d\,y_t = \theta_q(L)\,v_t$$

$$(1-\phi_1 L - \phi_2 L^2 - \phi_3 L^3)(1-L)\,y_t = (1+\theta L)\,v_t$$

$$(1-\phi_1 L - \phi_2 L^2 - \phi_3 L^3)(y_t - y_{t-1}) = v_t + \theta v_{t-1}$$

$$\begin{aligned} y_t &- \phi_1 y_{t-1} - \phi_2 y_{t-2} - \phi_3 y_{t-3} \\ &- y_{t-1} + \phi_1 y_{t-2} + \phi_2 y_{t-3} + \phi_3 y_{t-4} = v_t + \theta v_{t-1} \end{aligned}$$

$$y_t = (1+\phi_1)\,y_{t-1} - (\phi_1+\phi_2)\,y_{t-2} - (\phi_2+\phi_3)\,y_{t-3} - \phi_3 y_{t-4} + v_t + \theta v_{t-1}$$

$$\hat{y}_{t+1} = (1+\phi_1)\,y_t - (\phi_1-\phi_2)\,y_{t-1} - (\phi_2-\phi_3)\,y_{t-2} + \phi_3 y_{t-3} + \theta v_t$$

$$\hat{y}_{t+2} = (1+\phi_1)\,\hat{y}_{t+1} - (\phi_1+\phi_2)\,y_t - (\phi_2-\phi_3)\,y_{t-1} + \phi_3 y_{t-2} + 0$$

Model Fitting

Fitting ARIMA

For an $ARIMA(p, d, q)$ model

- Requires that the data be stationary after differencing

- Handling d is straight forward, just difference the original data d times (leaving $n - d$ observations)

$$y'_t = \Delta^d \, y_t$$

- After differencing, fit an $ARMA(p, q)$ model to y'_t.

- To keep things simple we'll assume $w_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_w)$

MLE - Stationarity & iid normal errors

If both of these conditions are met, then the time series y_t will also be normal.

If both of these conditions are met, then the time series y_t will also be normal.

In general, the vector $\mathbf{y} = (y_1, y_2, \ldots, y_t)'$ will have a multivariate normal distribution with mean $\{\boldsymbol{\mu}\}_i = E(y_i) = E(y_t)$ and covariance $\boldsymbol{\Sigma}$ where $\{\boldsymbol{\Sigma}\}_{ij} = \gamma_{i-j}$.

The joint density of $\mathbf{y}$ is given by

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{1}{(2\pi)^{t/2} \det(\boldsymbol{\Sigma})^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})' \, \Sigma^{-1} \, (\mathbf{y} - \boldsymbol{\mu})\right)$$

AR

Fitting $AR(1)$

$$y_t = \delta + \phi\, y_{t-1} + w_t$$

We need to estimate three parameters: δ, ϕ, and σ_w^2, we know

$$E(y_t) = \frac{\delta}{1-\phi}$$
$$Var(y_t) = \frac{\sigma_w^2}{1-\phi^2}$$
$$\gamma_h = \frac{\sigma_w^2}{1-\phi^2}\phi^{|h|}$$

Using these properties it is possible to write the distribution of $\mathbf{y}$ as a MVN but that does not make it possible to write down closed forms for the MLE estimate for δ, ϕ, and σ_w^2.

We can rewrite the density as follows,

$$
\begin{aligned}
f_{\mathbf{y}} &= f_{y_t,\, y_{t-1},\, \ldots,\, y_2,\, y_1} \\
&= f_{y_t|\, y_{t-1},\, \ldots,\, y_2,\, y_1} f_{y_{t-1}|y_{t-2},\, \ldots,\, y_2,\, y_1} \cdots f_{y_2|y_1} f_{y_1} \\
&= f_{y_t|\, y_{t-1}} f_{y_{t-1}|y_{t-2}} \cdots f_{y_2|y_1} f_{y_1}
\end{aligned}
$$

where,

$$
\begin{aligned}
y_1 &\sim \mathcal{N}\left(\frac{\delta}{1-\phi},\ \frac{\sigma_w^2}{1-\phi^2}\right) \\
y_t|y_{t-1} &\sim \mathcal{N}\left(\delta + \phi\, y_{t-1},\ \sigma_w^2\right) \\
f_{y_t|y_{t-1}}(y_t) &= \frac{1}{\sqrt{2\pi\, \sigma_w^2}} \exp\left(-\frac{1}{2}\frac{(y_t - \delta + \phi\, y_{t-1})^2}{\sigma_w^2}\right)
\end{aligned}
$$

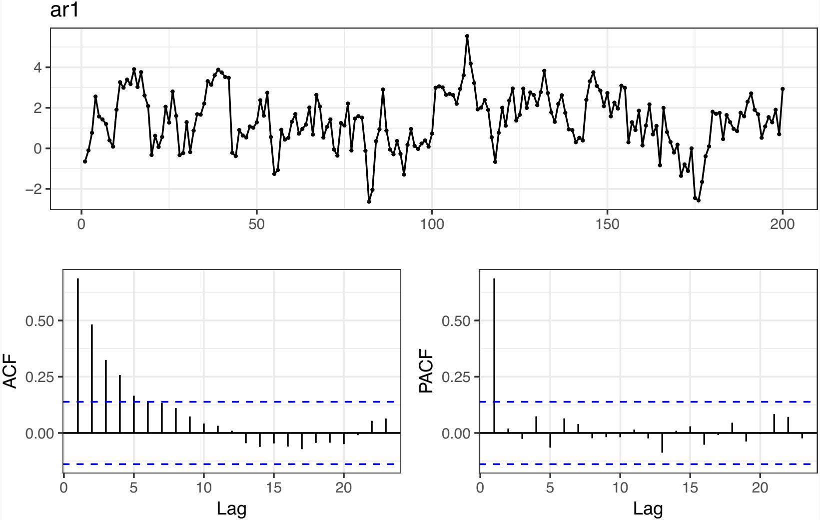
Log likelihood of AR(1)

$$\log f_{y_t|y_{t-1}}(y_t) = -\frac{1}{2}\left(\log 2\pi + \log \sigma_w^2 + \frac{1}{\sigma_w^2}(y_t - \delta + \phi\, y_{t-1})^2\right)$$

$$\begin{aligned}
\ell(\delta, \phi, \sigma_w^2) = \log f_{\mathbf{y}} &= \log f_{y_1} + \sum_{i=2}^{t} \log f_{y_i|y_{i-1}} \\
&= -\frac{1}{2}\left(\log 2\pi + \log \sigma_w^2 - \log(1-\phi^2) + \frac{(1-\phi^2)}{\sigma_w^2}\Big(y_1 - \frac{\delta}{1-\phi}\Big)^2\right) \\
&\quad - \frac{1}{2}\left((n-1)\log 2\pi + (n-1)\log \sigma_w^2 + \frac{1}{\sigma_w^2}\sum_{i=2}^{n}(y_i - \delta + \phi\, y_{i-1})^2\right) \\
&= -\frac{1}{2}\Bigg(n \log 2\pi + n \log \sigma_w^2 - \log(1-\phi^2) \\
&\qquad + \frac{1}{\sigma_w^2}\left((1-\phi^2)\Big(y_1 - \frac{\delta}{1-\phi}\Big)^2 + \sum_{i=2}^{n}(y_i - \delta + \phi\, y_{i-1})^2\right)\Bigg)
\end{aligned}$$

AR(1) Example

with $\phi = 0.75$, $\delta = 0.5$, and $\sigma_w^2 = 1$,



```
ar1_arima = forecast::Arima(ar1, order = c(1,0,0))
summary(ar1_arima)
## Series: ar1
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.6953  1.3708
## s.e.  0.0510  0.2296
##
## sigma^2 estimated as 1.011:  log likelihood=-284.23
## AIC=574.45   AICc=574.57   BIC=584.35
##
## Training set error measures:
##                       ME     RMSE       MAE       MPE     MAPE      MASE
## Training set 0.009862074 1.000539 0.7890342 -2924.705 3132.561 0.9192265
##                     ACF1
## Training set -0.01152999
```

$\mu = \delta$

$E(y_t) = \dfrac{\delta}{1-\phi}$

lm

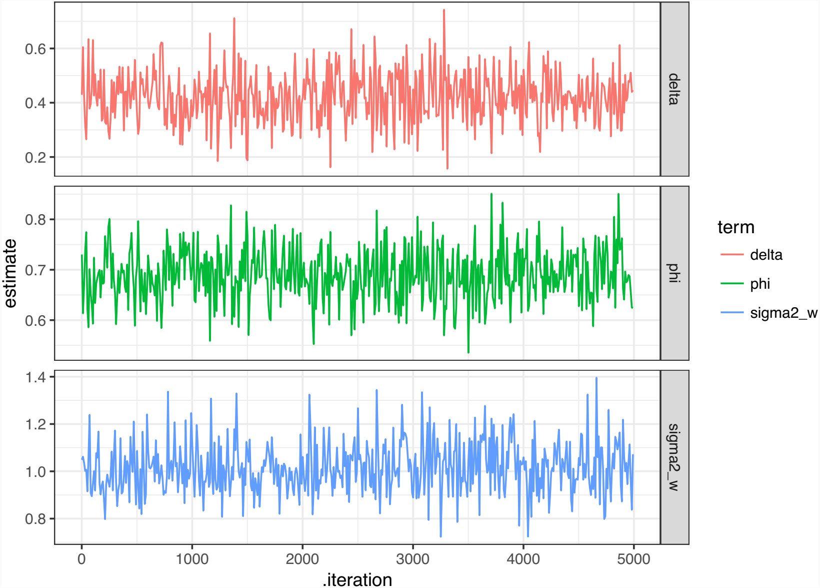
```
d = data_frame(y = ar1 %>% strip_attrs(), t=seq_along(ar1))
ar1_lm = lm(y~lag(y), data=d)
summary(ar1_lm)
##
## Call:
## lm(formula = y ~ lag(y), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06604 -0.63196  0.01817  0.70052  2.60532
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.44017    0.09985   4.408 1.71e-05 ***
## lag(y)       0.69144    0.05125  13.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.003 on 197 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4802, Adjusted R-squared:  0.4776
## F-statistic:   182 on 1 and 197 DF,  p-value: < 2.2e-16
```

$y_t = \beta \, \mathrm{lag}(y_t, 1) + c$

Bayesian AR(1) Model

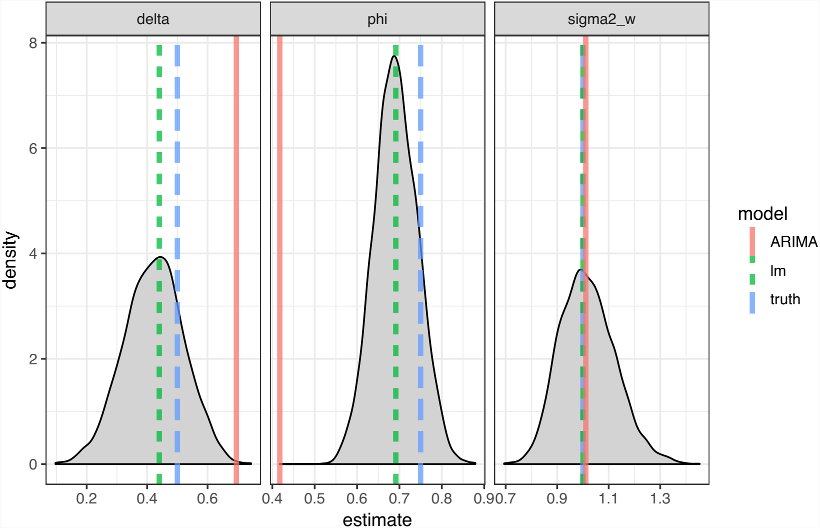
```
ar1_model = "model{
# likelihood
  y[1] ~ dnorm(delta/(1-phi), (sigma2_w/(1-phi^2))^-1)
  y_hat[1] ~ dnorm(delta/(1-phi), (sigma2_w/(1-phi^2))^-1)

  for (t in 2:length(y)) {
    y[t] ~ dnorm(delta + phi*y[t-1], 1/sigma2_w)
    y_hat[t] ~ dnorm(delta + phi*y[t-1], 1/sigma2_w)
  }

  mu = delta/(1-phi)

# priors
  delta ~ dnorm(0,1/1000)
  phi ~ dnorm(0,1)
  tau ~ dgamma(0.001,0.001)
  sigma2_w <- 1/tau
}"
```

Chains



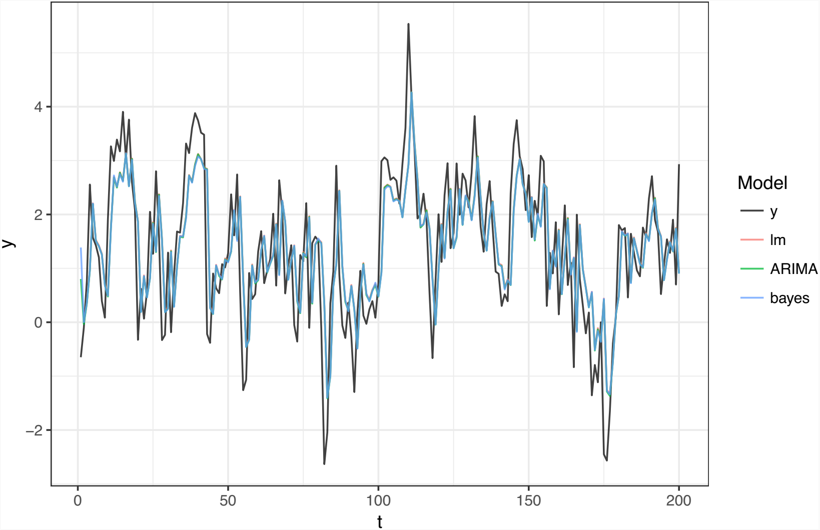
0.6
0.4
0.2
delta
0.8
0.7
0.6
phi
1.4
1.2
1.0
0.8
sigma2_w
estimate
0
1000
2000
3000
4000
5000
.iteration
term
delta
phi
sigma2_w

Posteriors

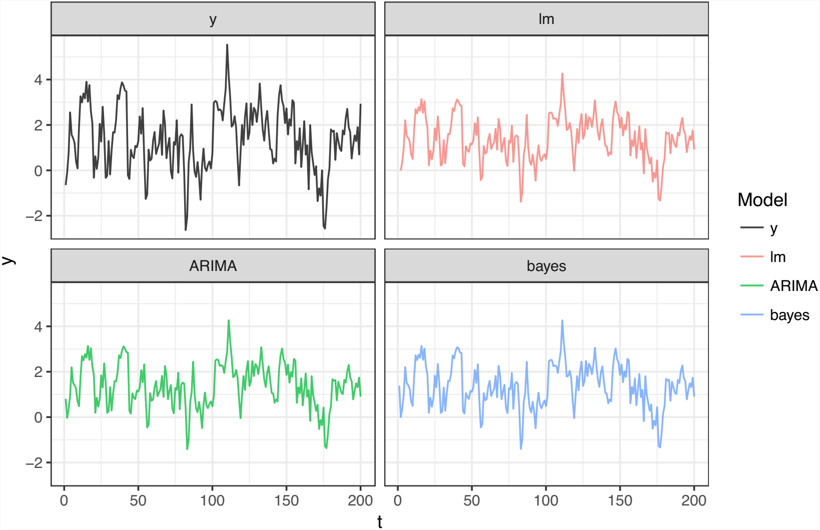


delta
phi
sigma2_w
density
8
6
4
2
0
0.2
0.4
0.6
0.4
0.5
0.6
0.7
0.8
0.9
0.7
0.9
1.1
1.3
estimate
model
ARIMA
lm
truth

Predictions



Faceted



We can rewrite the density as follows,

$$
\begin{aligned}
f(\mathbf{y}) &= f(y_t,\, y_{t-1},\, \ldots,\, y_2,\, y_1) \\
&= f(y_n | y_{n-1}, \ldots, y_{n-p}) \cdots f(y_{p+1} | y_p, \ldots, y_1) f(y_p,\, \ldots, y_1)
\end{aligned}
$$

Fitting AR(p) - Lagged Regression

We can rewrite the density as follows,

$$\begin{aligned} f(\mathbf{y}) &= f(y_t,\, y_{t-1},\, \dots,\, y_2,\, y_1) \\ &= f(y_n|y_{n-1}, \dots, y_{n-p}) \cdots f(y_{p+1}|y_p, \dots, y_1) f(y_p,\, \dots, y_1) \end{aligned}$$

Regressing y_t on $y_{t-p}, \dots, y_{t-1}$ gets us an approximate solution, but it ignores the $f(y_1,\, y_2,\, \dots, y_p)$ part of the likelihood.

How much does this matter (vs. using the full likelihood)?

- If p is near to n then probably a lot
- If $p << n$ then probably not much

Fitting AR(p) - Method of Moments

Recall for an AR(p) process,

$$\gamma(0) = \sigma_w^2 + \phi_1 \gamma(1) + \phi_2 \gamma(2) + \ldots + \phi_p \gamma(p)$$
$$\gamma(h) = \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2) + \ldots \phi_p \gamma(h-p)$$

We can rewrite the first equation in terms of σ_w^2,

$$\sigma_w^2 = \gamma(0) - \phi_1 \gamma(1) - \phi_2 \gamma(2) - \ldots - \phi_p \gamma(p)$$

these are called the Yule-Walker equations.

These equations can be rewritten into matrix notation as follows

$$\underset{p \times p}{\boldsymbol{\Gamma}_p} \underset{p \times 1}{\boldsymbol{\phi}} = \underset{p \times 1}{\boldsymbol{\gamma}_p} \qquad\qquad \underset{1 \times 1}{\sigma_w^2} = \underset{1 \times 1}{\gamma(0)} - \underset{1 \times p}{\boldsymbol{\phi}'} \underset{p \times 1}{\boldsymbol{\gamma}_\mathbf{p}}$$

where

$$\underset{p \times p}{\boldsymbol{\Gamma}_\mathbf{p}} = \{\gamma(j-k)\}_{j,k}$$

$$\underset{p \times 1}{\boldsymbol{\phi}} = (\phi_1, \phi_2, \dots, \phi_p)'$$

$$\underset{p \times 1}{\boldsymbol{\gamma}_p} = (\gamma(1), \gamma(2), \dots, \gamma(p))'$$

These equations can be rewritten into matrix notation as follows

$$\underset{p\times p}{\boldsymbol{\Gamma}_p}\,\underset{p\times 1}{\boldsymbol{\phi}} = \underset{p\times 1}{\boldsymbol{\gamma}_p} \qquad\qquad \underset{1\times 1}{\sigma_w^2} = \underset{1\times 1}{\gamma(0)} - \underset{1\times p}{\boldsymbol{\phi}'}\,\underset{p\times 1}{\boldsymbol{\gamma}_\mathbf{p}}$$

where

$$\underset{p\times p}{\boldsymbol{\Gamma}_\mathbf{p}} = \{\gamma(j-k)\}_{j,k}$$

$$\underset{p\times 1}{\boldsymbol{\phi}} = (\phi_1, \phi_2, \dots, \phi_p)'$$

$$\underset{p\times 1}{\boldsymbol{\gamma}_p} = (\gamma(1), \gamma(2), \dots, \gamma(p))'$$

If we estimate the covariance structure from the data we obtain $\hat{\boldsymbol{\gamma}}_p$ which can plug in and solve for $\boldsymbol{\phi}$ and σ_w^2,

$$\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\Gamma}}_p^{-1}\hat{\boldsymbol{\gamma}}_p \qquad\qquad \sigma_w^2 = \gamma(0) - \hat{\boldsymbol{\gamma}}_p{}'\hat{\boldsymbol{\Gamma}}_p^{-1}\hat{\boldsymbol{\gamma}}_p$$

ARMA

Fitting $ARMA(2, 2)$

$$y_t = \delta + \phi_1\, y_{t-1} + \phi_2\, y_{t-2} + \theta_1 w_{t-1} + \theta_2 w_{t-2} + w_t$$

Need to estimate six parameters: δ, ϕ_1, ϕ_2, θ_1, θ_2 and σ_w^2.

Fitting $ARMA(2,2)$

$$y_t = \delta + \phi_1 \, y_{t-1} + \phi_2 \, y_{t-2} + \theta_1 w_{t-1} + \theta_2 w_{t-2} + w_t$$

Need to estimate six parameters: δ, ϕ_1, ϕ_2, θ_1, θ_2 and σ_w^2.

We could figure out $E(y_t)$, $Var(y_t)$, and $Cov(y_t, y_{t+h})$, but the last two are going to be pretty nasty and the full MVN likehood is similarly going to be unpleasant to work with.

Fitting $ARMA(2,2)$

$$y_t = \delta + \phi_1 \, y_{t-1} + \phi_2 \, y_{t-2} + \theta_1 w_{t-1} + \theta_2 w_{t-2} + w_t$$

Need to estimate six parameters: δ, ϕ_1, ϕ_2, θ_1, θ_2 and σ_w^2.

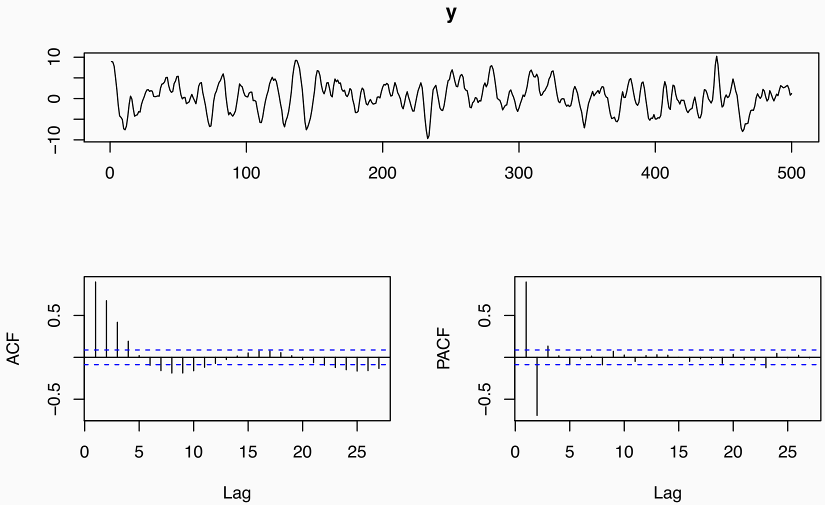
We could figure out $E(y_t)$, $Var(y_t)$, and $Cov(y_t, y_{t+h})$, but the last two are going to be pretty nasty and the full MVN likehood is similarly going to be unpleasant to work with.

Like the AR(1) and AR(p) processes we want to use conditioning to simplify things.

$$\begin{aligned} y_t | \delta, y_{t-1}, y_{t-2}, w_{t-1}, w_{t-2}, \phi_1, \phi_2, \theta_1, \theta_2 \\ \sim \mathcal{N}(\delta + \phi_1 \, y_{t-1} + \phi_2 \, y_{t-2} + \theta_1 w_{t-1} + \theta_2 w_{t-2}, \; \sigma_w^2) \end{aligned}$$

ARMA(2,2) Example

with $\phi = (1.3, -0.5)$, $\theta = (0.5, 0.2)$, $\delta = 0$, and $\sigma_w^2 = 1$ using the same models



ARIMA

```
forecast::Arima(y, order = c(2,0,2), include.mean = FALSE) %>% summary()
## Series: y
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2     ma1     ma2
##       1.3171  -0.5142  0.4332  0.1651
## s.e.  0.0881   0.0784  0.0955  0.0865
##
## sigma^2 estimated as 0.9517:  log likelihood=-696.8
## AIC=1403.61   AICc=1403.73   BIC=1424.68
##
## Training set error measures:
##                      ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.05247101 0.9716288 0.7755171 19.94389 95.20779 0.6500169
##                     ACF1
## Training set -0.01079679
```

AR only lm

```
lm(y ~ lag(y,1) + lag(y,2)) %>% summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7711 -0.6422  0.0119  0.6696  3.2150
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06476    0.04502   1.438    0.151
## lag(y, 1)    1.55990    0.03063  50.921   <2e-16 ***
## lag(y, 2)   -0.72697    0.03044 -23.879   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9969 on 495 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9147, Adjusted R-squared:  0.9144
## F-statistic:  2654 on 2 and 495 DF,  p-value: < 2.2e-16
```

1. Estimate a high order AR (remember AR $\Leftrightarrow$ MA when stationary + invertible)

2. Use AR to estimate values for unobserved w_t

3. Regress y_t onto $y_{t-1}, \ldots, y_{t-p}, \hat{w}_{t-1}, \ldots \hat{w}_{t-q}$

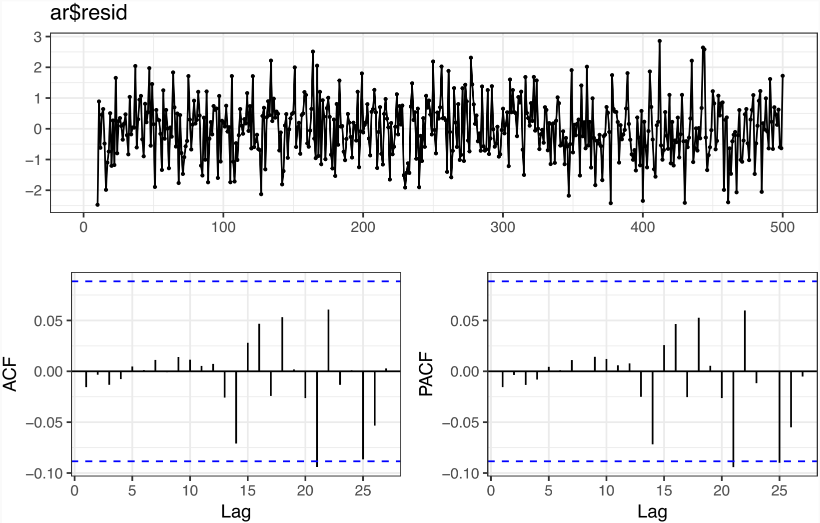
4. Update $\hat{w}_{t-1}, \ldots \hat{w}_{t-q}$ based on current model, refit and then repeat until convergence

Hannan-Rissanen - Step 1 & 2

```
ar = ar.mle(y, order.max = 10)
ar
##
## Call:
## ar.mle(x = y, order.max = 10)
##
## Coefficients:
##       1        2        3        4        5        6        7        8
##  1.7554  -1.1376   0.2222   0.0881  -0.0081  -0.2006   0.3130  -0.2674
##       9
##  0.0949
##
## Order selected 9  sigma^2 estimated as  0.916
```

Residuals

```
forecast::ggtsdisplay(ar$resid)
```



Hannan-Rissanen - Step 3

```
d = data_frame(
  y = y %>% strip_attrs(),
  w_hat1 = ar$resid %>% strip_attrs()
)

(lm1 = lm(y ~ lag(y,1) + lag(y,2) + lag(w_hat1,1) + lag(w_hat1,2), data=d)) %>%
  summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2) + lag(w_hat1, 1) + lag(w_hat1,
##     2), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.49762 -0.67125  0.02098  0.60126  2.91051
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.09387    0.04428   2.120   0.0345 *
## lag(y, 1)       1.32970    0.06119  21.732  < 2e-16 ***
## lag(y, 2)      -0.52939    0.05275 -10.035  < 2e-16 ***
## lag(w_hat1, 1)  0.41219    0.07617   5.412 9.84e-08 ***
## lag(w_hat1, 2)  0.13377    0.07335   1.824   0.0688 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9638 on 484 degrees of freedom
##   (11 observations deleted due to missingness)
```

Hannan-Rissanen - Step 4.1

```
d = modelr::add_residuals(d,lm1,"w_hat2")

(lm2 = lm(y ~ lag(y,1) + lag(y,2) + lag(w_hat2,1) + lag(w_hat2,2), data=d)) %>%
  summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2) + lag(w_hat2, 1) + lag(w_hat2,
##     2), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45706 -0.66872  0.04817  0.61811  2.83199
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.09550    0.04460   2.141   0.0328 *
## lag(y, 1)       1.31434    0.06434  20.429  < 2e-16 ***
## lag(y, 2)      -0.51885    0.05473  -9.481  < 2e-16 ***
## lag(w_hat2, 1)  0.42052    0.07902   5.321 1.58e-07 ***
## lag(w_hat2, 2)  0.14499    0.07560   1.918   0.0557 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9663 on 482 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.9168, Adjusted R-squared:  0.9161
## F-statistic:  1328 on 4 and 482 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.2

```
d = modelr::add_residuals(d,lm2,"w_hat3")

(lm3 = lm(y ~ lag(y,1) + lag(y,2) + lag(w_hat3,1) + lag(w_hat3,2), data=d)) %>%
  summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2) + lag(w_hat3, 1) + lag(w_hat3,
##     2), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48875 -0.67021  0.00531  0.61885  2.89492
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.09424    0.04484   2.102   0.0361 *
## lag(y, 1)       1.31907    0.06520  20.232  < 2e-16 ***
## lag(y, 2)      -0.52227    0.05557  -9.398  < 2e-16 ***
## lag(w_hat3, 1)  0.41250    0.07983   5.167  3.5e-07 ***
## lag(w_hat3, 2)  0.13981    0.07620   1.835   0.0671 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9692 on 480 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.9166, Adjusted R-squared:  0.9159
## F-statistic:  1320 on 4 and 480 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.3

```
d = modelr::add_residuals(d,lm3,"w_hat4")

(lm4 = lm(y ~ lag(y,1) + lag(y,2) + lag(w_hat4,1) + lag(w_hat4,2), data=d)) %>%
  summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2) + lag(w_hat4, 1) + lag(w_hat4,
##     2), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.47852 -0.64596  0.01075  0.61344  2.90021
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.09868    0.04474   2.205   0.0279 *
## lag(y, 1)       1.33693    0.06573  20.339  < 2e-16 ***
## lag(y, 2)      -0.53737    0.05607  -9.584  < 2e-16 ***
## lag(w_hat4, 1)  0.38853    0.08042   4.831 1.83e-06 ***
## lag(w_hat4, 2)  0.12337    0.07625   1.618   0.1063
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9661 on 478 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.9174, Adjusted R-squared:  0.9167
## F-statistic:  1327 on 4 and 478 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.4

```
d = modelr::add_residuals(d,lm4,"w_hat5")

(lm5 = lm(y ~ lag(y,1) + lag(y,2) + lag(w_hat5,1) + lag(w_hat5,2), data=d)) %>%
  summary()
##
## Call:
## lm(formula = y ~ lag(y, 1) + lag(y, 2) + lag(w_hat5, 1) + lag(w_hat5,
##     2), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4846 -0.6551  0.0259  0.6065  2.9046
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.10066    0.04500   2.237   0.0258 *
## lag(y, 1)       1.34414    0.06521  20.612  < 2e-16 ***
## lag(y, 2)      -0.54288    0.05566  -9.754  < 2e-16 ***
## lag(w_hat5, 1)  0.38065    0.07984   4.768 2.48e-06 ***
## lag(w_hat5, 2)  0.11431    0.07581   1.508   0.1323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9675 on 476 degrees of freedom
##   (19 observations deleted due to missingness)
## Multiple R-squared:  0.9168, Adjusted R-squared:  0.9161
## F-statistic:  1312 on 4 and 476 DF,  p-value: < 2.2e-16
```

RMSEs

```r
modelr::rmse(lm1, data = d)
## [1] 0.9588745

modelr::rmse(lm2, data = d)
## [1] 0.9613757

modelr::rmse(lm3, data = d)
## [1] 0.9641811

modelr::rmse(lm4, data = d)
## [1] 0.9611267

modelr::rmse(lm5, data = d)
## [1] 0.9624088
```

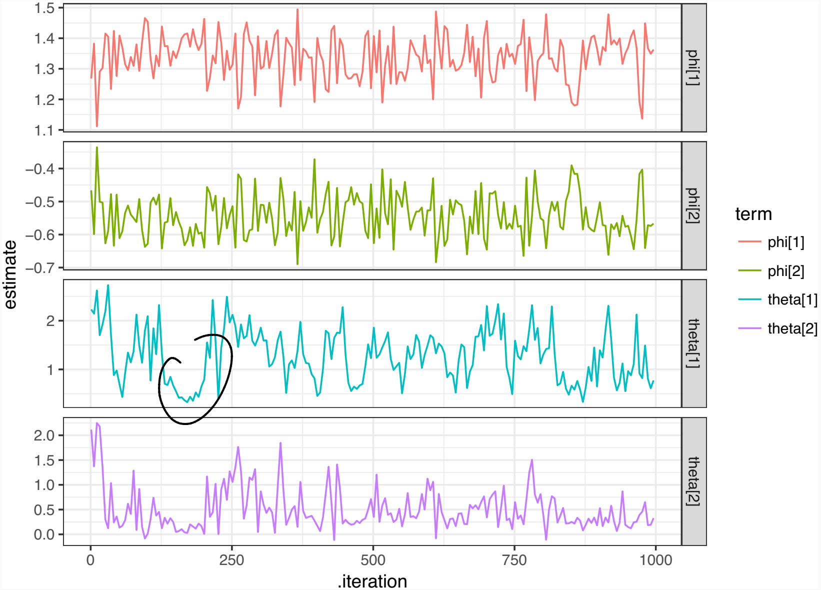
Bayesian Model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + e_t$$

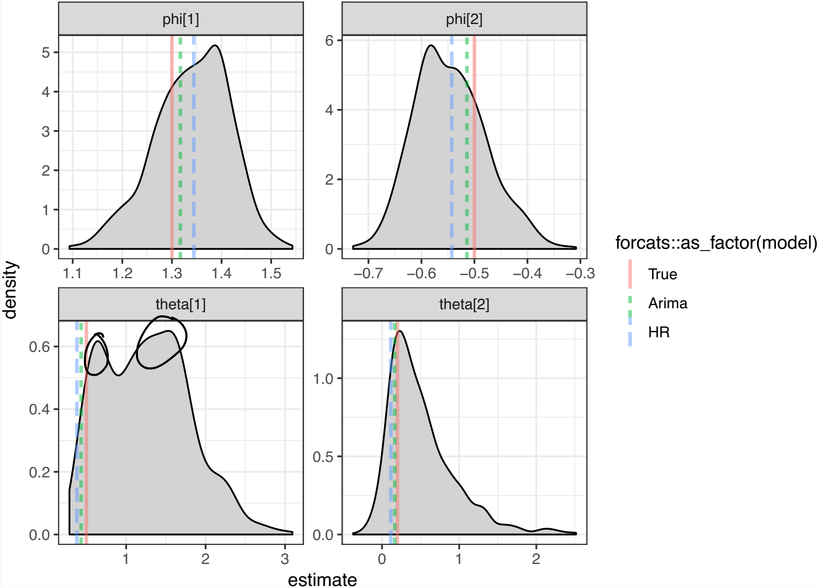
$$e_t \sim N(0, \sigma_e^2)$$

```
arma22_model = "model{
# Likelihood
  for (t in 1:length(y)) {
    y[t] ~ dnorm(mu[t], 1/sigma2_e)
  }

  mu[1] = phi[1] * y_0  + phi[2] * y_n1 + w[1] + theta[1]*w_0  + theta[2]*w_n1
  mu[2] = phi[1] * y[1] + phi[2] * y_0  + w[2] + theta[1]*w[1] + theta[2]*w_0
  for (t in 3:length(y)) {
    mu[t] = phi[1] * y[t-1] + phi[2] * y[t-2] + w[t] + theta[1] * w[t-1] + theta[2] * w[t-2]
  }

# Priors
  for(t in 1:length(y)){
    w[t] ~ dnorm(0,1/sigma2_w)
  }

  sigma2_w = 1/tau_w; tau_w ~ dgamma(0.001, 0.001)
  sigma2_e = 1/tau_e; tau_e ~ dgamma(0.001, 0.001)
  for(i in 1:2) {
    phi[i] ~ dnorm(0,1)
    theta[i] ~ dnorm(0,1)
  }

# Latent errors and series values
  w_0  ~ dt(0,tau_w,2)
  w_n1 ~ dt(0,tau_w,2)
  y_0  ~ dnorm(0,1/1000)
  y_n1 ~ dnorm(0,1/1000)
}"
```

Bayesian Fit



Posteriors


phi[1]
phi[2]
theta[1]
theta[2]
density
estimate
forcats::as_factor(model)
True
Arima
HR
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.5
1.0
1.1
1.2
1.3
1.4
1.5
−0.7
−0.6
−0.5
−0.4
−0.3