Lecture 10

Forecasting and Model Fitting

Colin Rundel

02/20/2017

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

One step ahead forecasting

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

$$\hat{y}_n = \delta + \phi\, y_{n-1} + \theta\, w_{n-1} + w_n$$

↳ $y_{n-1} - \hat{y}_{n-1}$

$$\begin{aligned}\hat{y}_{n+1} &= \delta + \phi\, y_n + \theta\, w_n + w_{n+1} \\ &\approx \delta + \phi\, y_n + \theta\,(y_n - \hat{y}_n) + 0\end{aligned}$$

$$\begin{aligned}\hat{y}_{n+2} &= \delta + \phi\, y_{n+1} + \theta\, w_{n+1} + w_{n+2} \\ &\approx \delta + \phi\, \hat{y}_{n+1} + \theta\, 0 + 0\end{aligned}$$

ARIMA(3,1,1) Example

$$(1-\phi_3(L))\underbrace{(1-L)^1 y_t} = (1+\theta_1 L)\, w_t$$

$$(1-\phi_1 L-\phi_2 L^2-\phi_3 L^3)(y_t - y_{t-1}) = w_t + \theta_1 w_{t-1}$$

$$y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \phi_3 y_{t-3}$$

$$- y_{t-1} + \phi_1 y_{t-2} + \phi_2 y_{t-3} + \phi_3 y_{t-4} = w_t + \theta w_{t-1}$$

$$\boxed{-(1+\phi_1)\, y_{t-1}}$$

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

$$\hat{y}_{t+1} = (1+\phi_1) y_t - (\phi_1 - \phi_2) y_{t-1} - (\phi_2 - \phi_3) y_{t-2} + \phi_3 y_{t-3} + 0 + \theta w_t$$

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

Model Fitting

Fitting ARIMA - MLE

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 $w_t \overset{iid}{\sim} \mathcal{N}(0, \sigma_w^2)$

Stationarity & normal errors

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

In general, the vector $\boldsymbol{y} = (y_1, y_2, \dots, y_t)'$ will have a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ where $\Sigma_{ij} = Cov(y_t, y_{t+i-j}) = \gamma_{i-j}$.

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

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

AR

Fitting $AR(1)$

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

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

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

$$Var(y_t) = \frac{\sigma_w^2}{1-\phi^2}$$

$$Cov(y_t, y_{t+h}) = \frac{\sigma_w^2}{1-\phi^2}\phi^{|h|}$$

$$\Sigma = \begin{pmatrix} \frac{\sigma_w^2}{1-\phi^2} & \cdots \\ \frac{\sigma_w^2}{1-\phi^2}\phi & \ddots \\ \vdots & \end{pmatrix}$$

Using these properties it is possible to write down the MVN distribution of $\boldsymbol{y}$ but not that easy to write down a closed form density which we can then use to find the MLE.

Conditional Density

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,

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

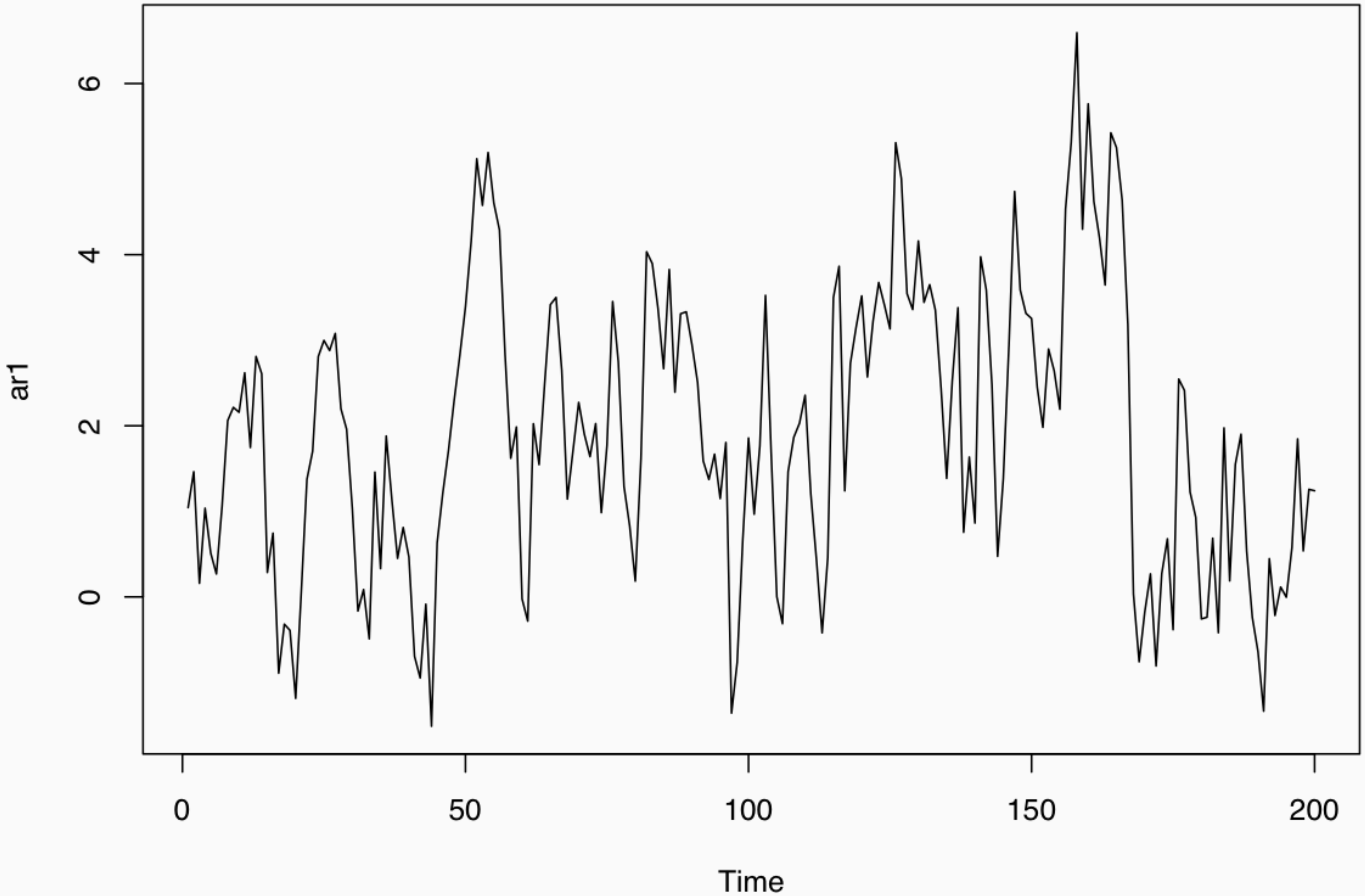
$$\log f_{y_t|y_{t-1}}(y_t) = -\frac{1}{2}\left(\log 2\pi + \log\sigma_w^2 + \frac{1}{\sigma_w^2}(y_t - \delta + \phi\, y_{t-1})^2\right)$$

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

y_i $\hat{y}_i$

AR(1) Example

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



Arima

```r
Arima(ar1, order = c(1,0,0)) %>% summary()
## Series: ar1
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.7593  1.8734
## s.e.  0.0454  0.3086
##
## sigma^2 estimated as 1.149:  log likelihood=-297.14
## AIC=600.28   AICc=600.4   BIC=610.17
##
## Training set error measures:
##                       ME     RMSE       MAE       MPE     MAPE
## Training set 0.004616374 1.066741 0.8410635 -327.6919 664.3204
##                   MASE        ACF1
## Training set 0.9186983 -0.00776572
```

$$\mu = \frac{\delta}{1-\phi}$$

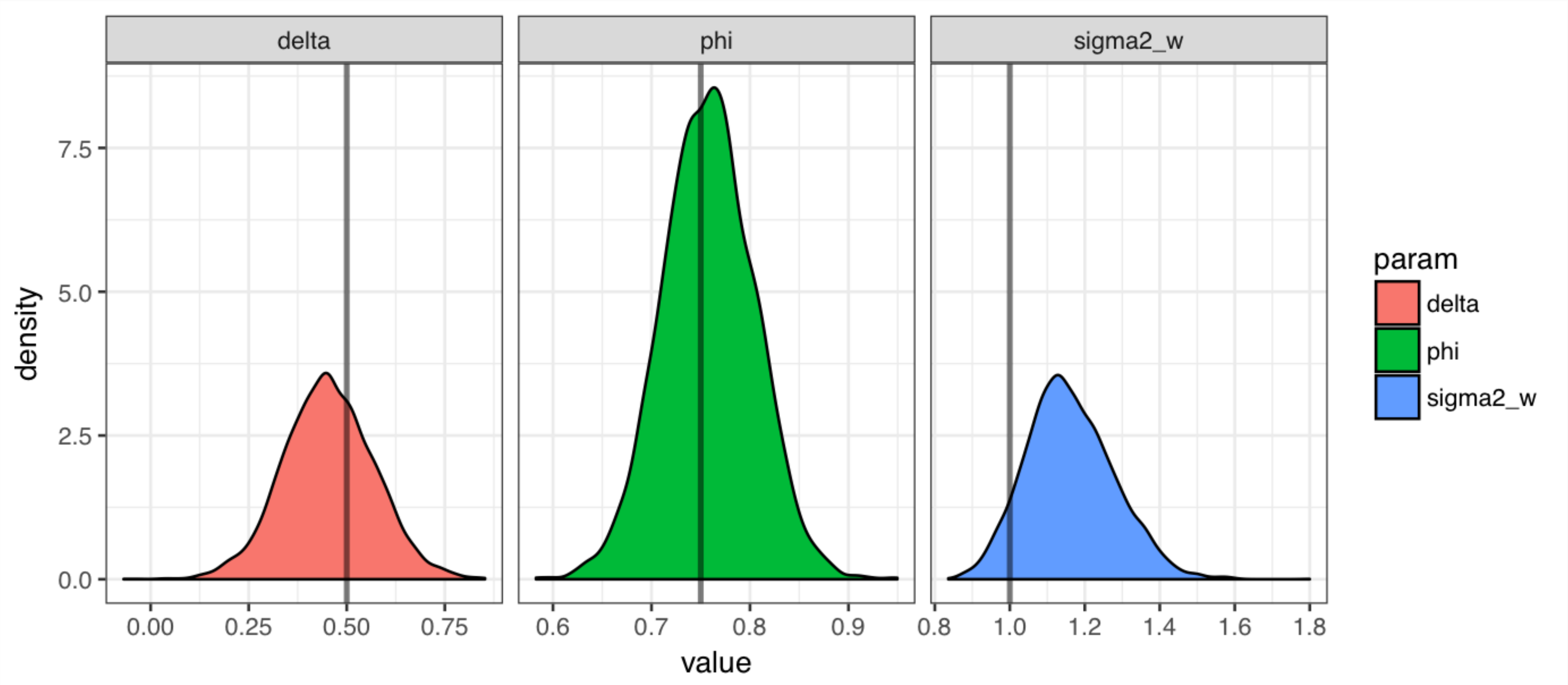
lm

```
lm(ar1~lag(ar1)) %>% summary()
##
## Call:
## lm(formula = ar1 ~ lag(ar1))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1863 -0.7596  0.0779  0.6099  2.8638
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4530     0.1161   3.904  0.00013 ***
## lag(ar1)     0.7621     0.0461  16.530  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 197 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.5811, Adjusted R-squared:  0.5789
## F-statistic: 273.2 on 1 and 197 DF,  p-value: < 2.2e-16
```

Bayesian AR(1) 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
## }
```

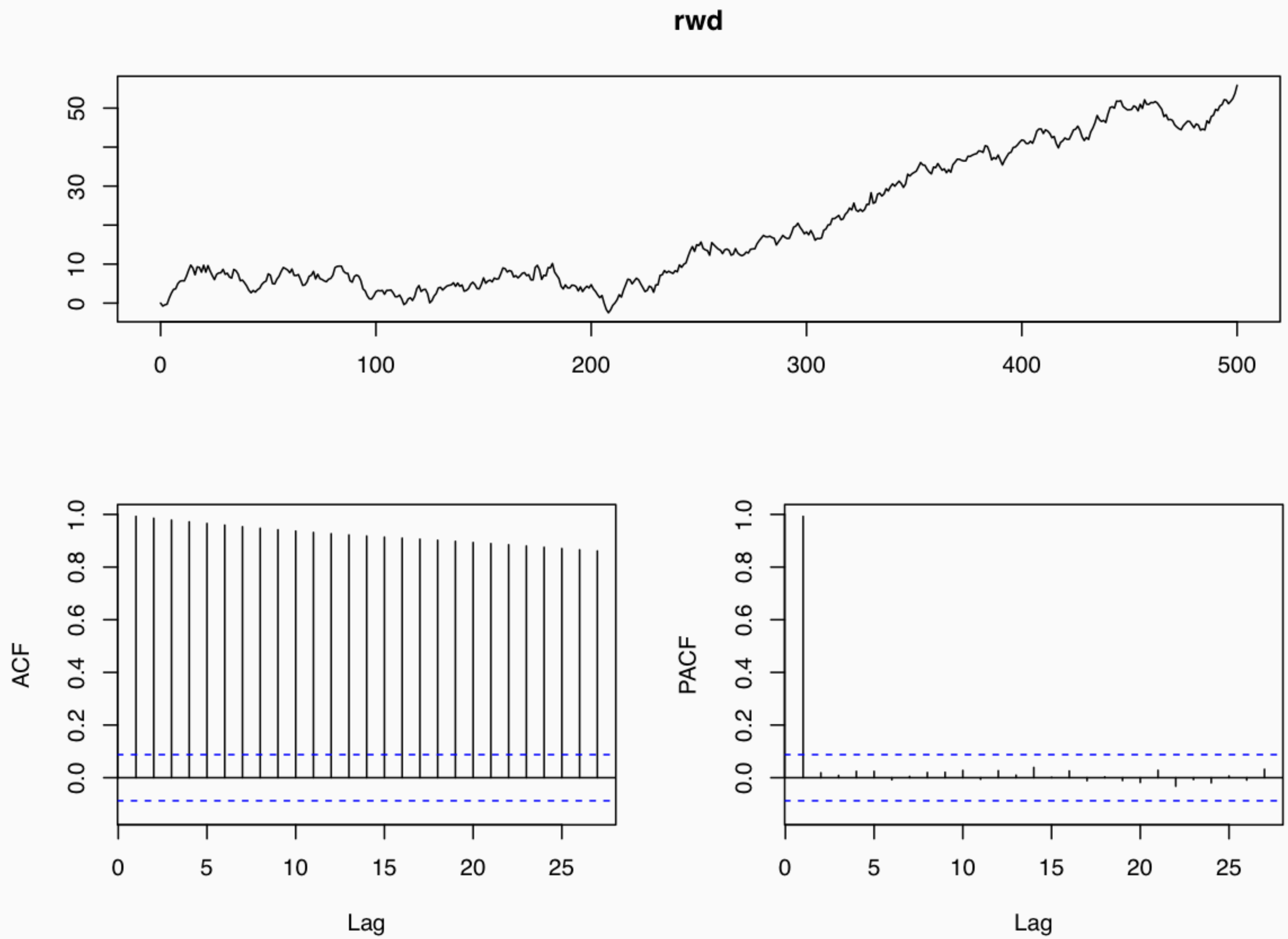
Posteriors



delta
phi
sigma2_w
7.5
5.0
2.5
0.0
density
0.00
0.25
0.50
0.75
0.6
0.7
0.8
0.9
0.8
1.0
1.2
1.4
1.6
1.8
value
param
delta
phi
sigma2_w

Random Walk with Drift

with $\phi = 1$, $\delta = 0.1$, and $\sigma_w^2 = 1$ using the same models



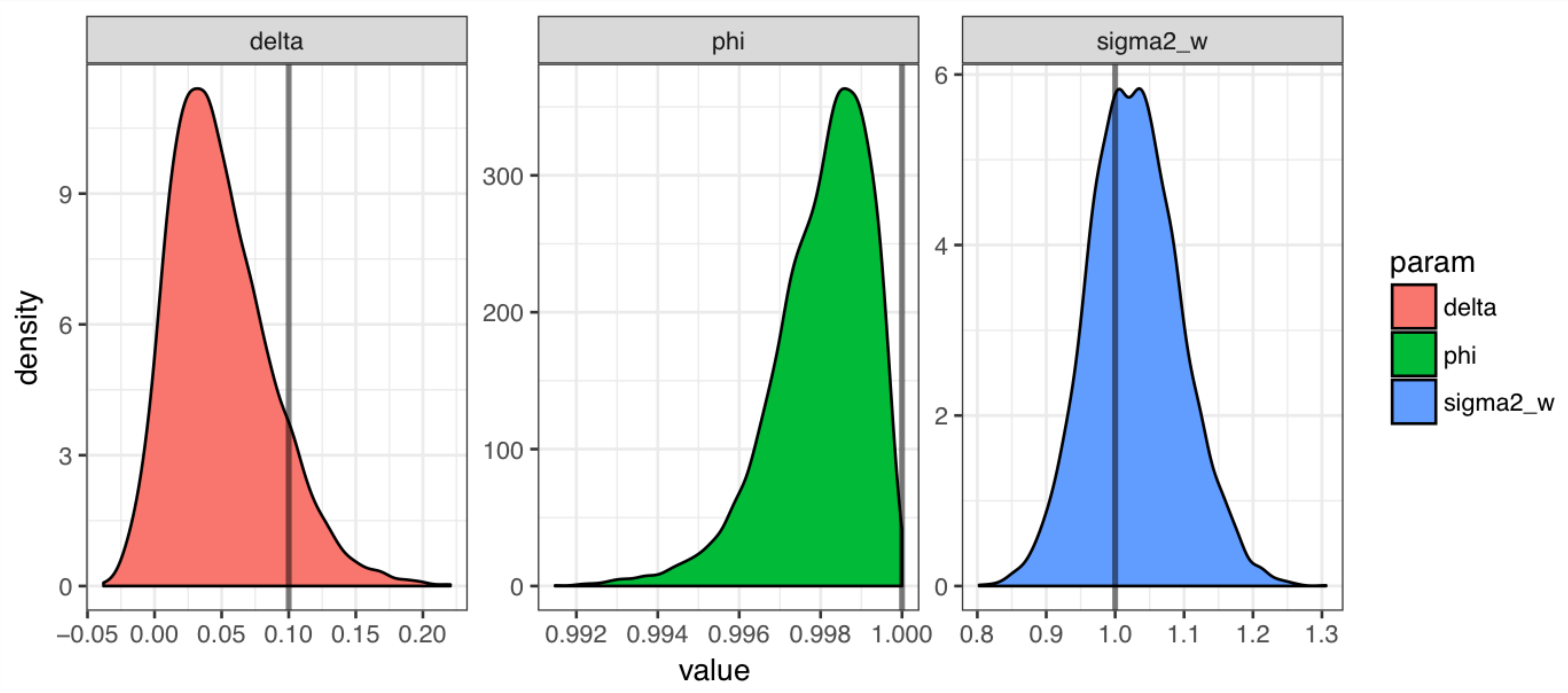
lm

```
lm(rwd~lag(rwd)) %>% summary()
##
## Call:
## lm(formula = rwd ~ lag(rwd))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83634 -0.71725  0.00629  0.69476  3.13117
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.083981   0.068588   1.224    0.221
## lag(rwd)    1.001406   0.002632 380.494   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.004 on 498 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.9966
## F-statistic: 1.448e+05 on 1 and 498 DF,  p-value: < 2.2e-16
```

Arima

```
Arima(rwd, order = c(1,0,0), include.constant = TRUE) %>% summary()
## Series: rwd
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
##       0.9992  26.4894
## s.e.  0.0010  23.5057
##
## sigma^2 estimated as 1.021:  log likelihood=-718.33
## AIC=1442.66   AICc=1442.7   BIC=1455.31
##
## Training set error measures:
##                     ME     RMSE       MAE  MPE MAPE      MASE
## Training set 0.1041264 1.008427 0.8142404 -Inf  Inf 0.9996364
##                    ACF1
## Training set 0.01365841
```

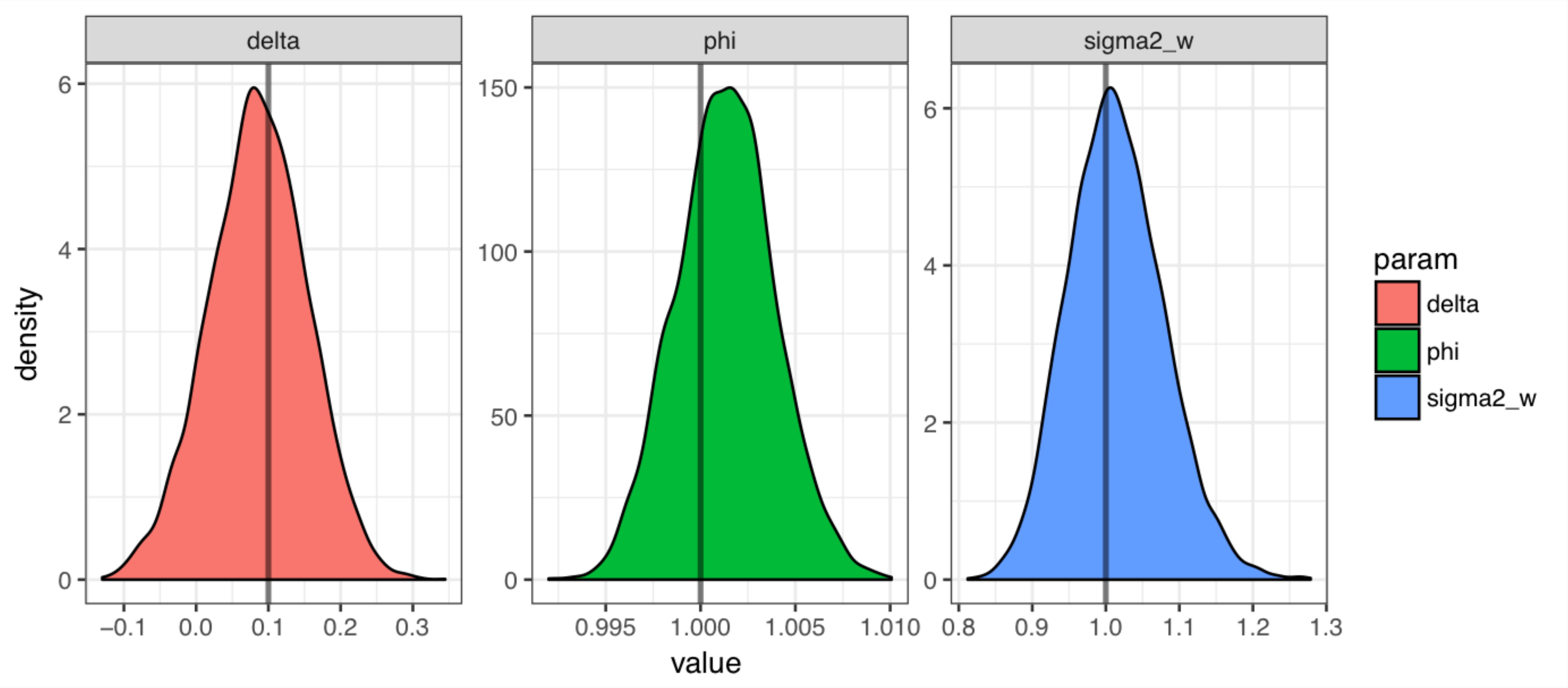
Bayesian Posteriors



Non-stationary Bayesian 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
## }
```

NS Bayesian Posteriors



Probability of being stationary

```
rwd_params$phi %>% abs() %>% {. < 1} %>% {sum(.) / length(.)}
## [1] 0.3046
```

Correct ARIMA

```
Arima(rwd, order = c(0,1,0), include.constant = TRUE) %>% summary()
## Series: rwd
## ARIMA(0,1,0) with drift
##
## Coefficients:
##         drift
##        0.1117
## s.e.   0.0448
##
## sigma^2 estimated as 1.007:  log likelihood=-710.63
## AIC=1425.26   AICc=1425.29   BIC=1433.69
##
## Training set error measures:
##                         ME     RMSE       MAE  MPE MAPE      MASE
## Training set -2.228961e-07 1.001325 0.8082318 -Inf  Inf 0.9922597
##                    ACF1
## Training set 0.01027574
```

We can rewrite the density as follows,

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

Fitting AR(p)

We can rewrite the density as follows,

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

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

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

- If p is not much smaller than n then probably a lot
- If $p << n$ then probably not much

ARMA

Fitting $AR(2,2)$

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

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

Fitting $AR(2, 2)$

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

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 likely be pretty nasty and the full MVN likehood is similarly going to be unpleasant to work with.

Fitting $AR(2,2)$

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

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

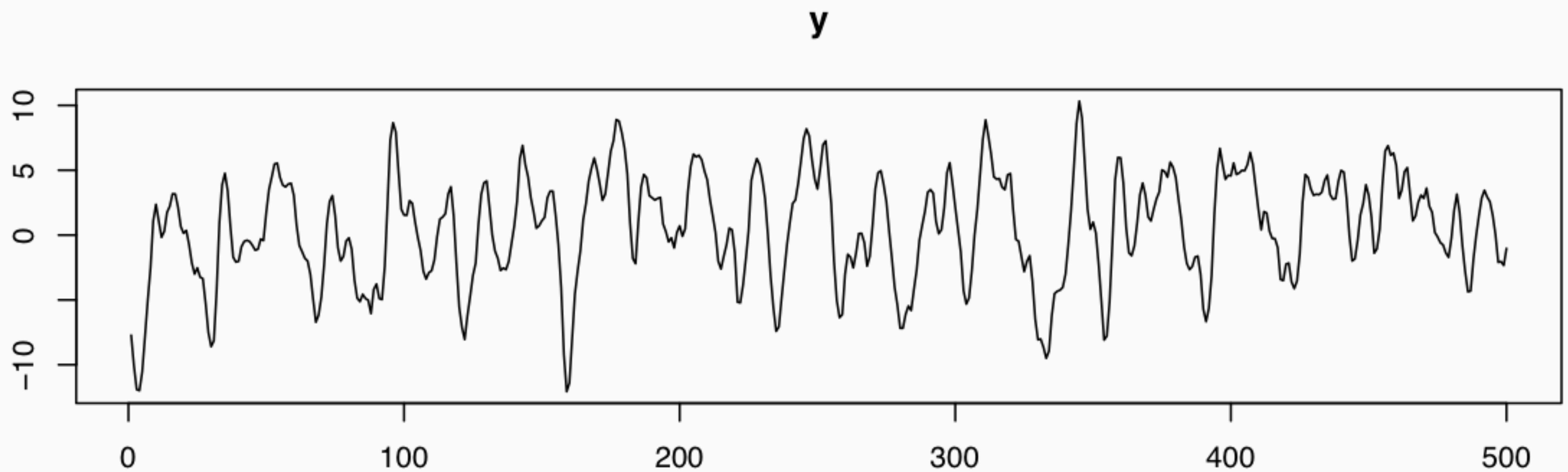
We could figure out $E(y_t)$, $Var(y_t)$, and $Cov(y_t, y_{t+h})$, but the last two are going to likely be pretty nasty and the full MVN likehood is similarly going to be unpleasant to work with.

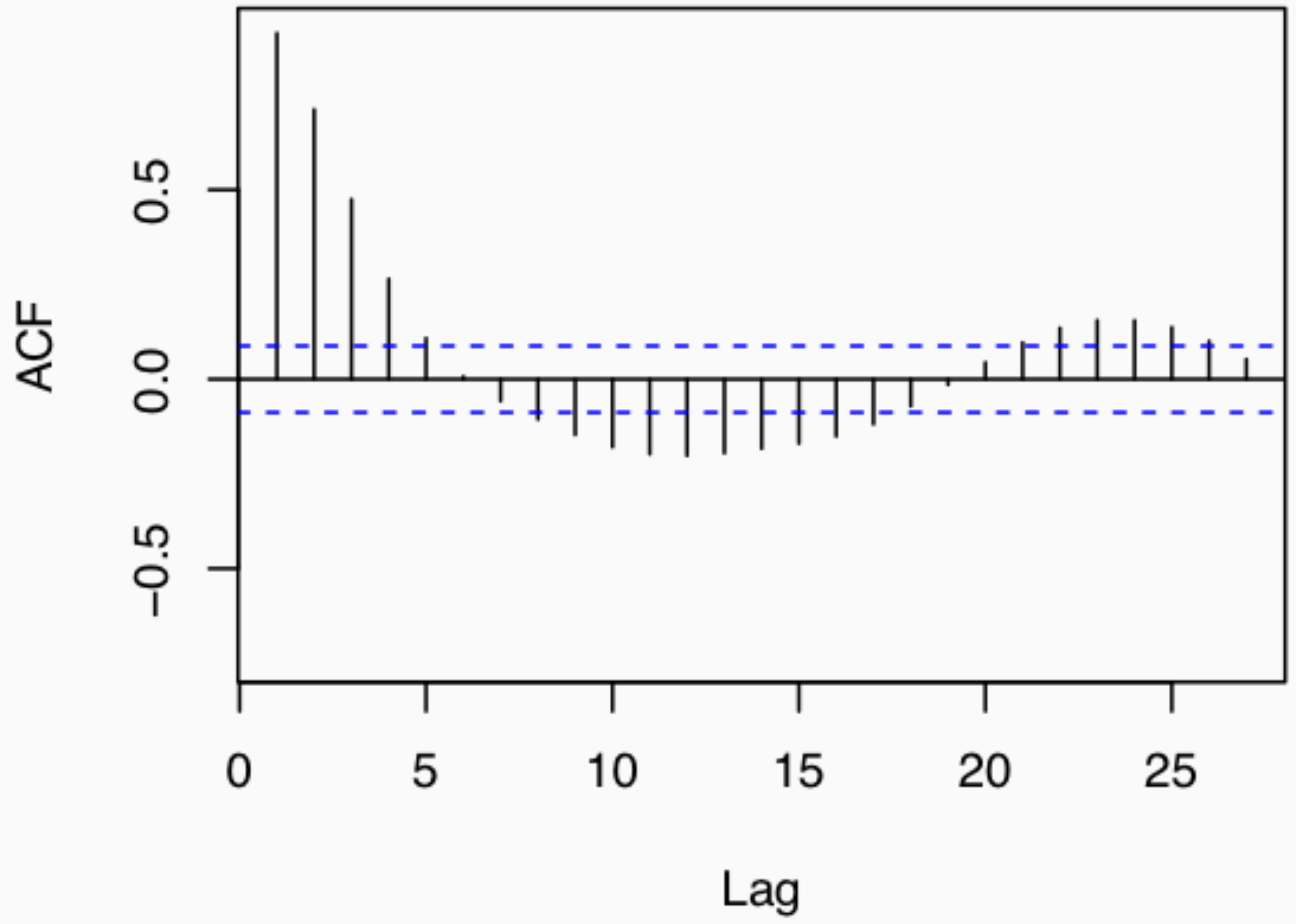
Like the AR(1) and AR(p) processes we want to use conditioning to simplify things.

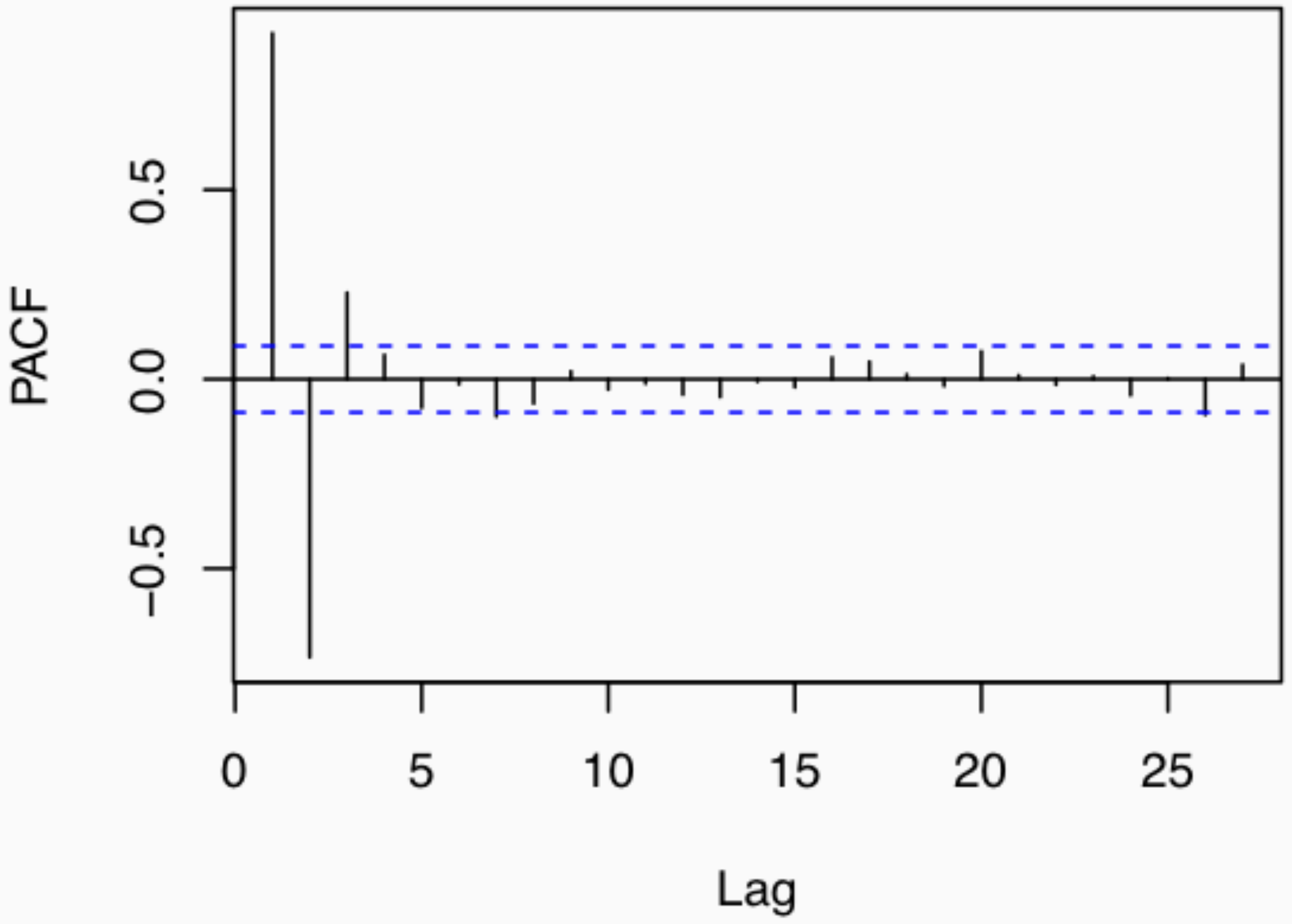
$$y_t | \delta, y_{t-1}, y_{t-2}, w_{t-1}, w_{t-2} \sim \mathcal{N}(\delta + \phi_1\, y_{t-1} + \phi_2\, y_{t-2} + \theta_1 w_{t-1} + \theta_2 w_{t-2},\ \sigma_w^2)$$

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(y, order = c(2,0,2), include.mean = FALSE) %>% summary()
## Series: y
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2     ma1     ma2
##       1.3154  -0.4991  0.5200  0.2481
## s.e.  0.0725   0.0677  0.0793  0.0633
##
## sigma^2 estimated as 1.067:  log likelihood=-725.52
## AIC=1461.04   AICc=1461.16   BIC=1482.11
##
## Training set error measures:
##                      ME     RMSE       MAE      MPE     MAPE
## Training set 0.05502909 1.028655 0.8260218 13.65446 86.84326
##                   MASE         ACF1
## Training set 0.6224348 -0.004832567
```

AR only lm

```
(lm_ar = 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
## -3.3908 -0.7164 -0.0235  0.7502  3.0950
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.08661    0.04867   1.779   0.0758 .
## lag(y, 1)    1.59893    0.02947  54.262   <2e-16 ***
## lag(y, 2)   -0.74823    0.02936 -25.482   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.078 on 495 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9307, Adjusted R-squared:  0.9304
## F-statistic:  3324 on 2 and 495 DF,  p-value: < 2.2e-16
```

1.2 -0.5

Hannan-Rissanen Algorithm

1. Estimate a high order AR (remember AR $\Leftrightarrow$ MA when stationary + invertible)
2. Use AR to estimate values for unobserved $\hat{w}_t$
3. Regress y_t onto $y_{t-1}, \dots, y_{t-p}, \hat{w}_{t-1}, \dots \hat{w}_{t-q}$
4. Update $\hat{w}_{t-1}, \dots \hat{w}_{t-q}$ based on current model, refit and then repeat until convergence

```
ar = ar.mle(y, order.max = 20)
ar
##
## Call:
## ar.mle(x = y, order.max = 20)
##
## Coefficients:
##       1        2        3        4        5        6        7
##  1.8272  -1.1903   0.1643   0.2110  -0.2331   0.2143  -0.1158
##
## Order selected 7  sigma^2 estimated as  1.041
ar$resid
## Time Series:
## Start = 1
## End = 500
## Frequency = 1
##   [1]           NA           NA           NA           NA           NA
##   [6]           NA           NA  0.013771505  1.815385143 -1.523601485
##  [11] -1.482175624  0.257910962  0.779526070  0.500584221 -0.874932004
##  [16]  1.000773447 -0.403367540 -0.432516832 -0.213762215  0.419791693
##  [21]  0.256815097 -1.532807297 -0.385121768 -0.074529360  0.545797695
##  [26] -1.622523443  0.190508558 -2.276038961 -1.218302454 -0.165499325
##  [31] -0.422165854  2.284211482  1.090020206 -0.831161663  0.147961063
##  [36] -0.913676024 -1.060367233 -0.313198281  0.401868246 -0.752567843
##  [41]  0.615242705 -0.630185112 -0.017926780  0.127457456 -0.382266477
##  [46] -0.212700408  0.045666952  0.373128585 -0.946750691  2.386716523
##  [51] -0.097543980  0.081058896  1.264579915 -0.312389635 -0.226815185
```

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
## -3.3756 -0.7172 -0.0223  0.6697  3.1330
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.10594    0.04752   2.229  0.02625 *
## lag(y, 1)        1.33911    0.05421  24.702  < 2e-16 ***
## lag(y, 2)       -0.52481    0.04817 -10.894  < 2e-16 ***
## lag(w_hat1, 1)   0.46734    0.07103   6.579 1.23e-10 ***
## lag(w_hat1, 2)   0.21347    0.07057   3.025  0.00262 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.036 on 486 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.9326, Adjusted R-squared:  0.9321
## F-statistic:  1681 on 4 and 486 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.1

```
d = 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
## -3.3589 -0.7487 -0.0288  0.6471  3.1058
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.12429    0.04735   2.625 0.008948 **
## lag(y, 1)       1.31028    0.05624  23.299  < 2e-16 ***
## lag(y, 2)      -0.50471    0.04923 -10.252  < 2e-16 ***
## lag(w_hat2, 1)  0.50130    0.07125   7.036  6.8e-12 ***
## lag(w_hat2, 2)  0.23462    0.07086   3.311 0.000999 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.027 on 484 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.9341, Adjusted R-squared:  0.9335
## F-statistic:  1714 on 4 and 484 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.2

```
d = 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
## -3.3717 -0.7489 -0.0311  0.6465  3.0557
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.12529    0.04754   2.636 0.008672 **
## lag(y, 1)       1.31322    0.05588  23.501  < 2e-16 ***
## lag(y, 2)      -0.50769    0.04897 -10.367  < 2e-16 ***
## lag(w_hat3, 1)  0.50220    0.07190   6.985 9.52e-12 ***
## lag(w_hat3, 2)  0.24635    0.07173   3.435 0.000645 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 482 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.9342, Adjusted R-squared:  0.9336
## F-statistic:  1710 on 4 and 482 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.3

```
d = 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
## -3.3167 -0.7553 -0.0292  0.6503  3.1417
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.12368    0.04761   2.598  0.00968 **
## lag(y, 1)       1.30982    0.05578  23.481  < 2e-16 ***
## lag(y, 2)      -0.50496    0.04889 -10.328  < 2e-16 ***
## lag(w_hat4, 1)  0.50657    0.07184   7.051 6.21e-12 ***
## lag(w_hat4, 2)  0.25342    0.07172   3.534  0.00045 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.028 on 480 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.9339
## F-statistic:  1710 on 4 and 480 DF,  p-value: < 2.2e-16
```

Hannan-Rissanen - Step 4.4

```
d = 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
## -3.3614 -0.7620 -0.0282  0.6656  3.1229
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.12142    0.04778   2.541 0.011353 *
## lag(y, 1)       1.31454    0.05590  23.517  < 2e-16 ***
## lag(y, 2)      -0.50870    0.04900 -10.382  < 2e-16 ***
## lag(w_hat5, 1)  0.50350    0.07219   6.975 1.02e-11 ***
## lag(w_hat5, 2)  0.24604    0.07203   3.416 0.000691 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.03 on 478 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.9343, Adjusted R-squared:  0.9338
## F-statistic:  1700 on 4 and 478 DF,  p-value: < 2.2e-16
```

RMSEs

```r
rmse(lm_ar, data = d)
## [1] 1.074382

rmse(lm1, data = d)
## [1] 1.030996

rmse(lm2, data = d)
## [1] 1.021696

rmse(lm3, data = d)
## [1] 1.022922

rmse(lm4, data = d)
## [1] 1.022807

rmse(lm5, data = d)
## [1] 1.024861
```

Bayesian 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