Chapter 3. Autoregressive Integrated Moving Average (ARIMA) Models

Dexter Cahoy

November 29, 2018

Outline

Estimation in R

- The `sarima()` function from `astsa` package estimates ARIMA(p,d=0,q) models:
- ARIMA orders : `(p,d,q)`
- The constant reported in the output (labeled as "xmean") of the `arima` function is in fact the mean μ.
- Syntax: `sarima( data, p, d, q)`

Example 1

- **AR(2):** $x_t = 1.5x_{t-1} - 0.75x_{t-2} + w_t, n = 144$.

```
$ttable
      Estimate     SE  t.value p.value
ar1     1.5246 0.0518  29.4599   0.000
ar2    -0.7802 0.0516 -15.1137   0.000
xmean  -0.2688 0.3192  -0.8423   0.401
```

- What is the fitted model?
- Using the large-sample property for MLEs, the 95% confidence interval for ϕ_1 and ϕ_2 are

$$1.524 \pm (1.96)0.051 = (1.424, \;\; 1.624) \quad \text{and}$$

$$-0.78 \pm (1.96)0.051 = (-0.88, \;\; -0.68)\,, \quad \textit{respectively}.$$

Example 2

- **MA(1):** $x_t = w_t + 0.9 w_{t-1}, \qquad n = 150.$

```
$ttable
      Estimate     SE t.value p.value
ma1     0.8662 0.0427 20.2760  0.0000
xmean   0.0816 0.1681  0.4853  0.6282
```

- Algebraic expression of your model?

- What is an approximate 95% confidence interval for θ?

Example 3

- **ARMA(1,1):** $x_t = 0.83x_{t-1} + w_t - 0.43w_{t-1}, n = 150.$

```
$ttable
      Estimate     SE t.value p.value
ar1     0.7863 0.0763 10.3118  0.0000
ma1    -0.2863 0.1162 -2.4636  0.0149
xmean  -0.2487 0.2719 -0.9147  0.3618
```

- What are the approximate 95% confidence intervals for μ, ϕ and θ? What is the algebraic expression of the fitted model?

Example 4: El Niño's Recruitment Data

```
$ttable
      Estimate     SE  t.value p.value
ar1     1.3512 0.0416  32.4933       0
ar2    -0.4612 0.0417 -11.0687       0
xmean  61.8585 4.0039  15.4494       0
```

- What is the algebraic expression of the fitted model?

Forecasting

- We are given a sample path up until time $t = n$, say, $x_{1:n} = c(x_1, x_2, \ldots, x_n)$.

- Our goal is to forecast $x_{t+1}, x_{t+2}, x_{t+3}, \ldots$. In general, we want to predict, $x_{n+m}, m \geq 1$. Note that we are forecasting future random values of the process. We call m the **lead** time.

- *CRITERION:* Find an estimator $g(x_{1:n})$ that minimizes the mean squared error(MSE),

$$E\left[x_{n+m} - g(x_{1:n})\right]^2.$$

- The **minimum MSE**(MMSE) predictor of x_{n+m} is

$$x_{n+m}^n = E\left(x_{n+m} | x_{1:n}\right).$$

Forecasting

- DETERMINISTIC TREND: Assume $x_t = \mu_t + y_t$, where μ_t is a deterministic function, and y_t is a white noise with zero mean and variance γ_0. Then

$$x_{n+m}^n = \mu_{n+m}$$

  is the MMSE forecast.

- The **forecast error** is $x_{n+m} - x_{n+m}^n$.

- We can use the `sarima.for()` function from `astsa` package for SARIMA models.

Forecasting

- *Simple Linear Trend:* $\mu_t = \beta_0 + \beta_1 g(t), t = 1, \ldots, n$ and $g-$known and depends only on t. Then

$$x_{n+m}^n = \mu_{n+m} = \beta_0 + \beta_1 g(n+m).$$

- *Cosine Trend:* $\mu_t = \beta_0 + \beta_1 \cos(2\pi ft) + \beta_2 \sin(2\pi ft), t = 1, \ldots, n.$ Then

$$x_{n+m}^n = \mu_{n+m} = \beta_0 + \beta_1 \cos(2\pi f(n+m)) + \beta_2 \sin(2\pi f(n+m)).$$

- MMSE forecasts (the coefficients) are to be estimated.

Forecasting

- **Global temp deviation data (1880-2015)**. We fitted $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$ and obtained

```
> mod1=lm(globtemp~time+time2,dat)
> mod1

Call:
lm(formula = globtemp ~ time + time2, data = dat)

Coefficients:
(Intercept)         time        time2
  2.844e+02   -2.992e-01    7.860e-05
```

- Hence,

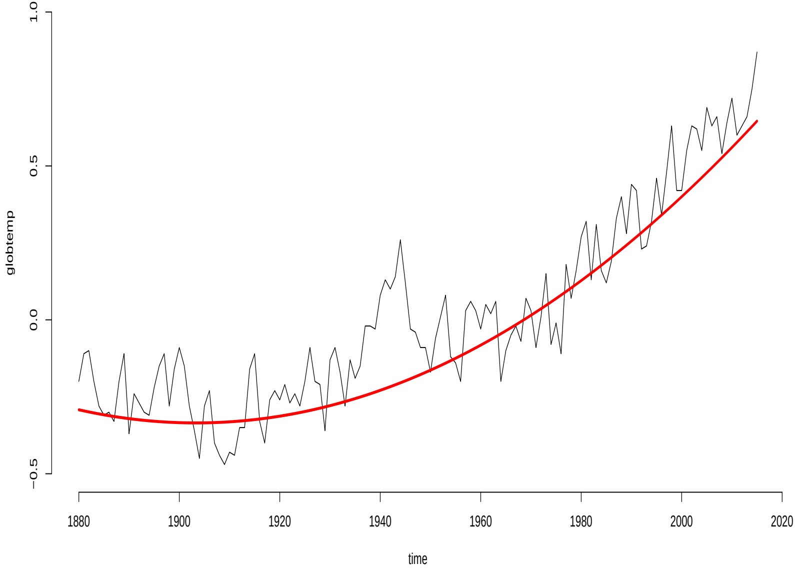
$$\hat{\mu}_t = 2.844e + 02 - 2.992e - 01 * t + 7.860e - 05 * t^2.$$

and

$$\hat{\mu}_{n+m} = 2.844e+02-2.992e-01*(n+m)+7.860e-05*(n+m)^2.$$

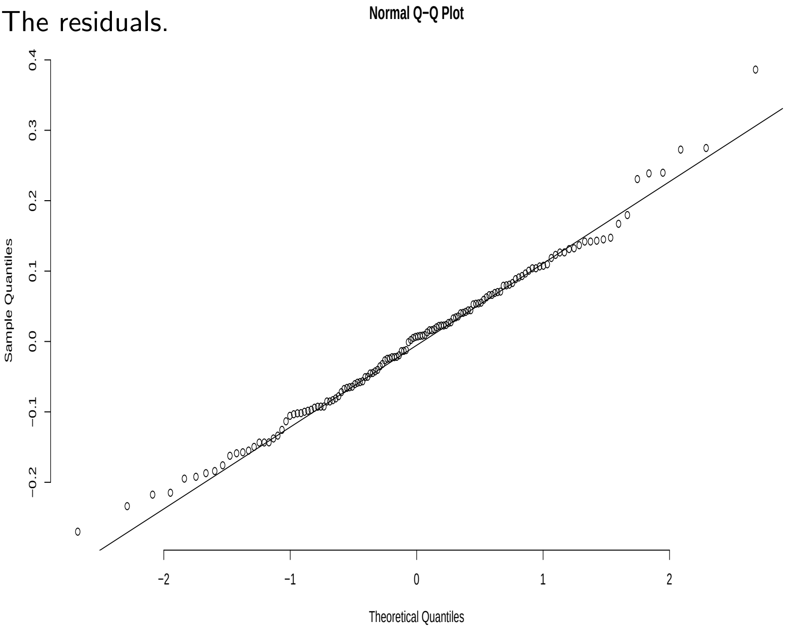
Forecasting

- The data and the fitted model.



Forecasting

- The residuals.



Forecasting

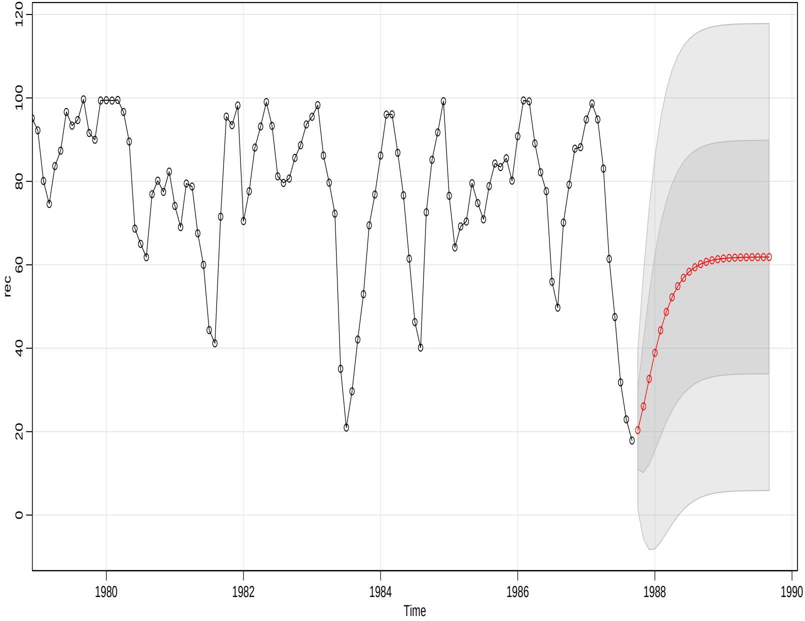
- **Global temp deviation data (1880-2015)**. Our forecast for 2018 is

$$\begin{aligned}\hat{\mu}_{2015+3} &= 2.844e+02 - 2.992e-01 * (2015+3) \\ &+ 7.860e-05 * (2015+3)^2 = 0.6990664.\end{aligned}$$

- Forecast for 2050 is

$$\begin{aligned}\hat{\mu}_{2015+35} &= 2.844e+02 - 2.992e-01 * (2015+35) \\ &+ 7.860e-05 * (2015+35)^2 = 1.3565.\end{aligned}$$

Forecasting El Niño's Recruitment Data



Lab for Today

- Apply the Box-Jenkins method (transform to stationarity if necessary and identify the time series model(e.g., ARMA(p,q))) for the luteinizing hormone data. Write down the algebraic expression of the fitted model. How is your "final" model compared with an AR(3) ?
- Use your final model to forecast the next 24 luteinizing hormone measurements.

Integrated Models for Nonstationary Data

- A time series data follows an **ARIMA (p,d,q)** process if

$$\nabla^d x_t \text{ is } ARMA(p, q).$$

- Thus,

$$\phi(B)(1 - B)^d x_t = \theta(B) w_t.$$

- x_t is not stationary as $\phi(B)(1 - B)^d$ has roots on the unit circle. Stationarity can only be imposed on $(1 - B)^d x_t$.

- Use `sarima()` for fitting and `arima.sim()` for simulating ARIMA(p,d,q).

Integrated Models for Nonstationary Data

- Autoregressive (AR) models, moving average (MA) models, and autoregressive moving average (ARMA) models are all members of the ARIMA(p, d, q) family.

- ARIMA "=" AutoRegressive Integrated Moving Average

- Recall the ARIMA(p, d, q) model: $\phi(B)(1-B)^d x_t = \theta(B) w_t$.

$$
\begin{aligned}
AR(p) &\iff ARIMA(p,0,0) \\
MA(q) &\iff ARIMA(0,0,q) \\
ARMA(p,q) &\iff ARIMA(p,0,q) \\
ARI(p,d) &\iff ARIMA(p,d,0) \\
IMA(d,q) &\iff ARIMA(0,d,q).
\end{aligned}
$$

Integrated Models for Nonstationary Data

- *BEWARE* of **over-differencing**, i.e., differencing using higher orders (large d).

- Prediction in non-stationary processes is difficult as the mean-square prediction error or forecast error variance increases with lead time m.

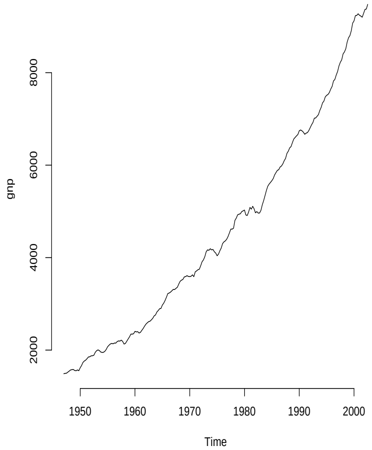
- We apply many of the same techniques (estimation, prediction etc) to test the fit of ARIMA(p, d, q) models or to ARMA(p,q) models after data differencing.

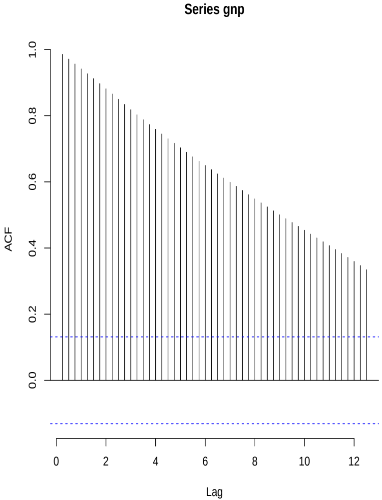
Building ARIMA

- Plot the data to inspect anomalies.
- Transform the data to stabilize variability or to stationarity.
- Identify the orders p, d, q for ARIMA(p,d,q).
- Estimate parameters
- Diagnostics
- Finalize model and apply (e.g., forecast, etc).

Building ARIMA

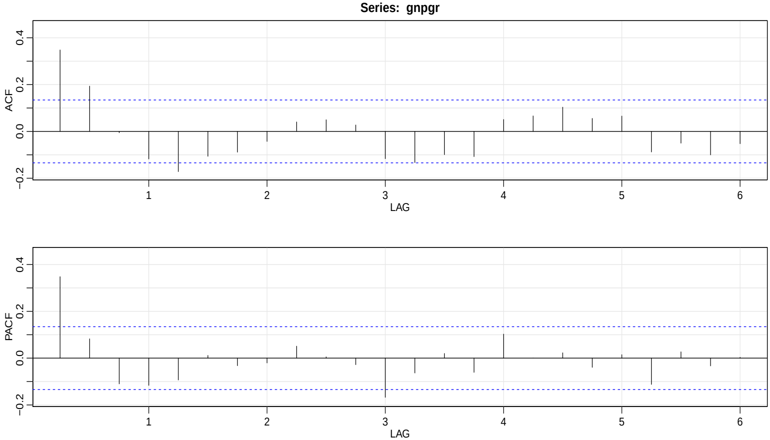
- **GNP Data**: y_t is the quarterly US GNP (in billions of dollars) from 1947-2002 (n=223). The data is obtained from `http://research.stlouisfed.org`. In finance, the growth rate (percent change) $x_t = \nabla \log y_t$ is often analyzed instead.





Building ARIMA

- The sample ACF and the sample PACF of the growth rate x_t:



- ACF cuts off after lag 2 ; PACF tails off suggesting MA(2) for x_t or ARIMA(0,1,2) for log(GNP).
- ACF tails off; PACF cuts off after lag 1 suggesting AR(1) for x_t or ARIMA(1,1,0) for log(GNP).

Building ARIMA

- Using MLE and `sarima()` in R, we fit an MA(2) to the growth rate x_t and obtain

```
>gnpgr = diff(log(gnp))
>sarima(gnpgr, 0, 0, 2)

$ttable
      Estimate     SE t.value p.value
ma1     0.3028 0.0654  4.6272  0.0000
ma2     0.2035 0.0644  3.1594  0.0018
xmean   0.0083 0.0010  8.7178  0.0000
```

- Hence the MA(2) model is

$$\hat{x}_t = .008_{.001} + .303_{.065} w_{t-1} + .204_{.064} w_{t-2} + w_t.$$

Building ARIMA

- For AR(1):

```
>gnpgr = diff(log(gnp))
>sarima(gnpgr, 1, 0, 0)

$ttable
      Estimate     SE t.value p.value
ar1     0.3467 0.0627  5.5255       0
xmean   0.0083 0.0010  8.5398       0
```

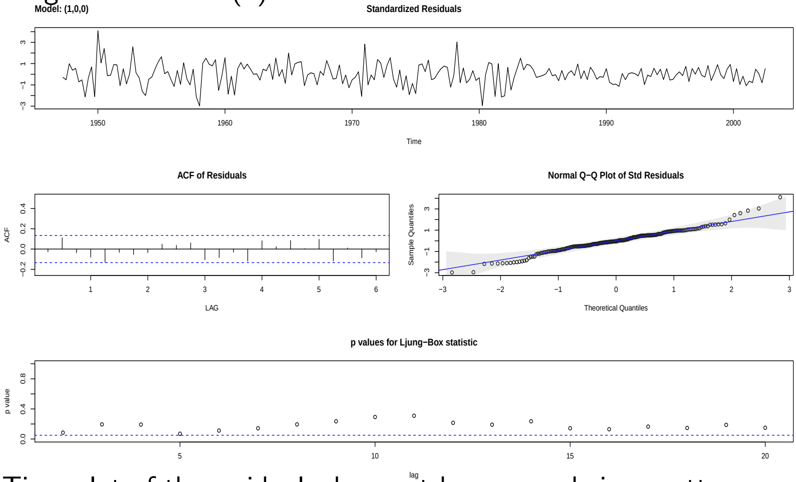
- Hence, the AR(1) model is

$$\hat{x}_t = .008_{.001}(1 - .347) + .347_{.063} x_{t-1} + w_t.$$

- Note that .008 is labeled as "constant" in `sarima(log(gnp), 1, d=1, 0)` output, which is needed to be in the model.

Building ARIMA

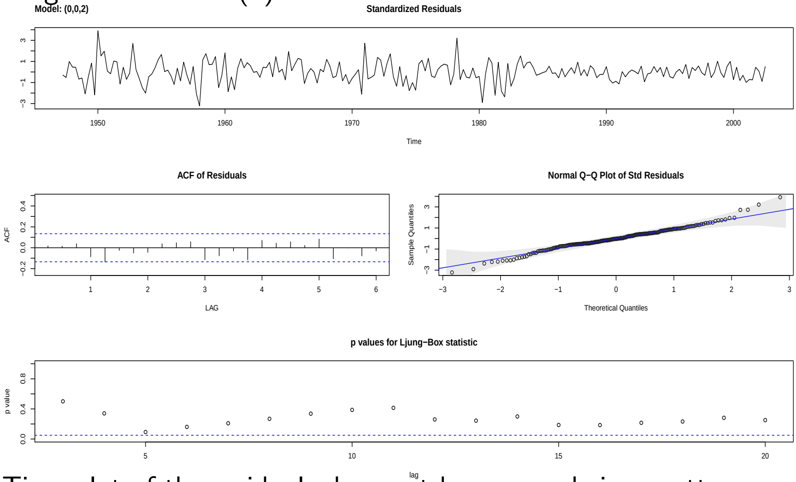
- Diagnostics for AR(1):



- Time plot of the residuals does not have any obvious pattern. The ACF's are all within the limits. The QQ plot shows some points falling just outside the confidence limits. The p-values of the Ljung-Box test are all above the significance level.

Building ARIMA

- Diagnostics for MA(2):



- Time plot of the residuals does not have any obvious pattern. The ACF's are all within the limits. The p-values of the Ljung-Box test are all above the significance level. The QQ plot affirms normality of the residuals.

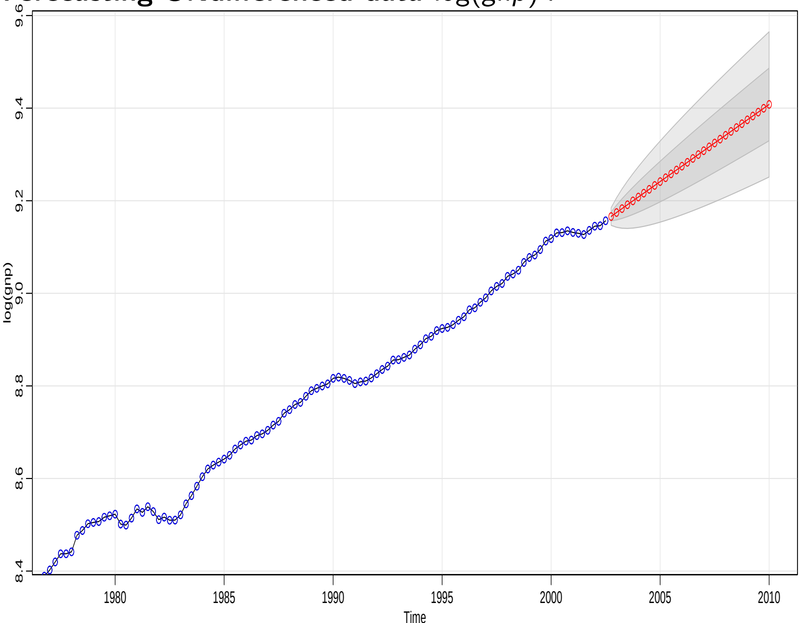
Building ARIMA

- To choose the final model, compare AIC, AICc, and BIC. The smaller the better.
- ```
  > sarima(gnpgr, 0, 0, 2)      # MA(2)
  $AIC
  [1] -8.297695
  $AICc
  [1] -8.287855
  $BIC
  [1] -9.251712

  > sarima(gnpgr, 1, 0, 0)      # AR(1)
  $AIC
  [1] -8.294403
  $AICc
  [1] -8.284898
  $BIC
  [1] -9.263748
  ```
- AIC and AICc prefer MA(2), whereas BIC prefers AR(1). BIC often selects the model of smaller order than the AIC and AICc. It is not unreasonable to accept AR(1) as it is easier to work with and is more parsimonious. How about `auto.arima`'s model?
  ```

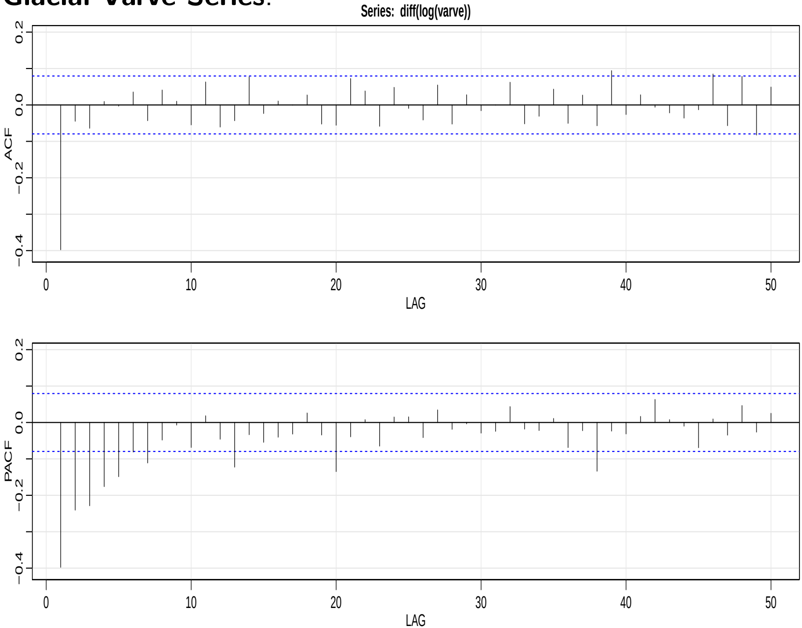
Building ARIMA

- **Forecasting UNdifferenced data** $\log(gnp)$:



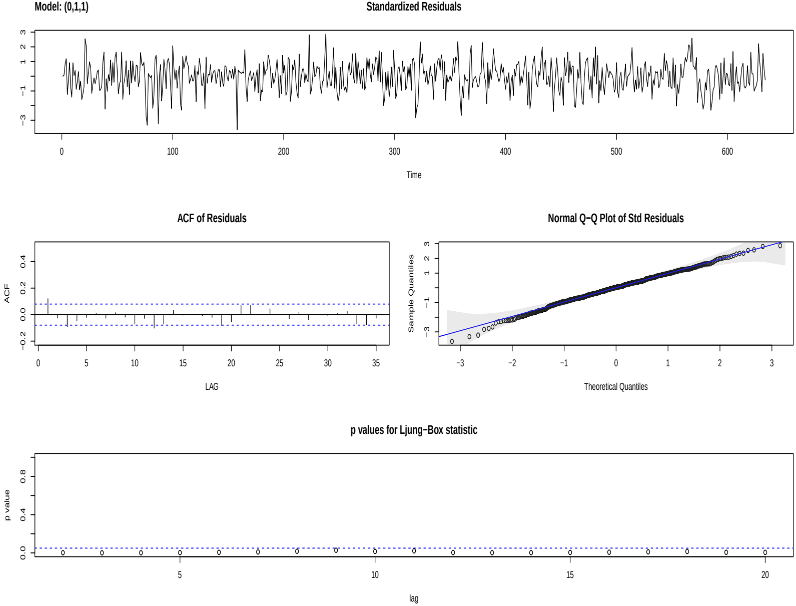
Building ARIMA

- **Glacial Varve Series**:



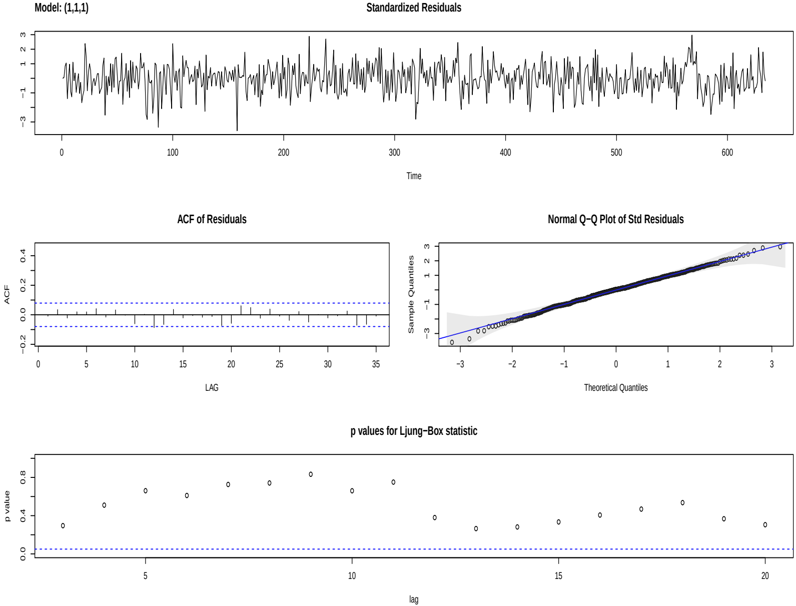
Building ARIMA

- **Glacial Varve Series**: ARIMA(0,1,1) fit to the log (varve data)



Building ARIMA

- **Glacial Varve Series**: ARIMA(1,1,1) fit to the log (varve data).



Building ARIMA

- To choose the final model, compare AIC, AICc, and BIC. The smaller the better.
- ```
  > sarima(log(varve), 0, 1, 1, no.constant=TRUE)   # ARIMA(0,1,1)
  $AIC
  [1] -0.4436731
  $AICc
  [1] -0.4404885
  $BIC
  [1] -1.436651

  > sarima(log(varve), 1, 1, 1, no.constant=TRUE)   # ARIMA(1,1,1)
  $AIC
  [1] -0.4701992
  $AICc
  [1] -0.4669845
  $BIC
  [1] -1.456155
  ```
- AIC, AICc and BIC all prefer ARIMA(1,1,1). How about `auto.arima`'s model?
  ```

Building ARIMA

- Results for fitting ARIMA(1,1,1):

```
>f2=sarima(log(varve), 1, 1, 1, no.constant=TRUE)   # ARIMA(1,1,1)
>f2

$ttable
    Estimate     SE  t.value p.value
ar1   0.2330 0.0518   4.4994       0
ma1  -0.8858 0.0292 -30.3861       0
```

- The model can be written as

$$(1 - .233_{.052}B)(1 - B)\hat{x}_t = (1 - .886_{.029}B)w_t,$$

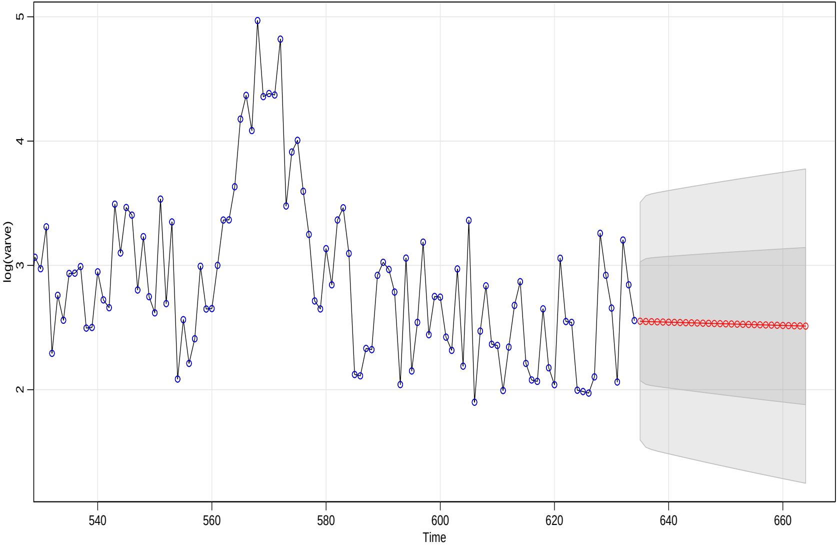
which is the same as

$$\hat{x}_t = (1 + .233_{.052})\, x_{t-2} - .233_{.052}x_{t-2} + w_t - .886_{.029}w_{t-1}.$$

- Clearly, $(1 - B)\hat{x}_t$ is stationary and invertible.

Building ARIMA

- Minimum MSE Forecast for UNdifferenced data log(*varve*) :



Lab for Today

1. Fit an ARIMA(p, d, q) model to the global temperature data `globtemp` performing all of the necessary diagnostics. After deciding on an appropriate model, forecast (with limits) the next 10 years of the UNdifferenced data. Comment.
2. Use auto.arima() and compare your results in 1 above.

Multiplicative Seasonal ARIMA

- Often, dependence on the past tends to be strongest at multiples of some underlying lag s.

  For example, monthly economic data may exhibit strong quarterly or yearly annual trends.

- Define the *seasonal* AR(P) operator as

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps}.$$

- Let the *seasonal* MA(Q) operator as

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs}.$$

Multiplicative Seasonal ARIMA

- The **multiplicative seasonal autoregressive integrated moving average (SARIMA)** model, denoted as

  ARIMA(p,d,q)$\times$(P,D,Q)$_s$ is

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \Theta_Q(B^s)\theta(B)w_t,$$

  $\nabla^d = (1-B)^d$ is the ordinary difference, $\nabla_s^D = (1-B^s)^D$ is the seasonal difference,

  $\Phi_P(B^s)$ is the seasonal AR, and $\Theta_Q(B^s)$ is the seasonal MA component.

Multiplicative Seasonal ARIMA

GUIDELINES:

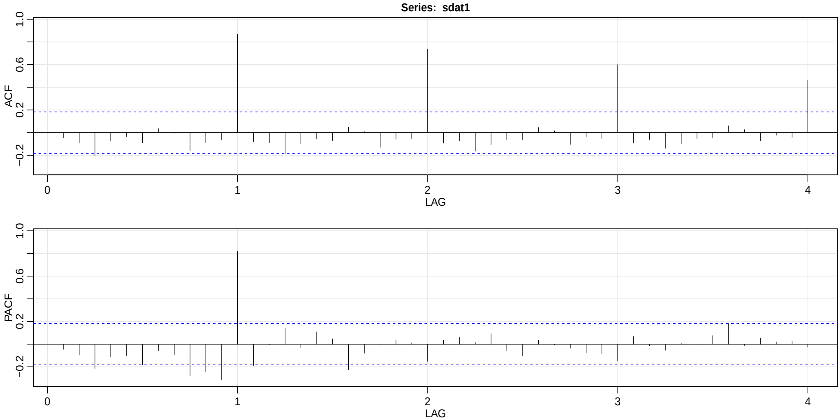
- *Seasonality* s will appear in the ACF by tapering slowly at multiples of s, or PACF is very large at s. Try $\nabla_s^D X_t$ for small D.

- *Seasonal terms*: Examine the patterns of the ACF and PACF at the first few **seasonal lags** that are multiples of s. Seasonal AR(P): PACF zero after $s, 2s, \ldots, Ps$. Seasonal MA(Q): ACF zero after $s, 2s, \ldots, Qs$.

- *Non-seasonal terms*: Examine the early lags $1, 2, 3, \ldots$ to judge non-seasonal terms like in non-seasonal ARMA. Spikes in the PACF (at low lags) indicate non-seasonal AR terms. Spikes in the ACF (at low lags) indicate non-seasonal MA terms.

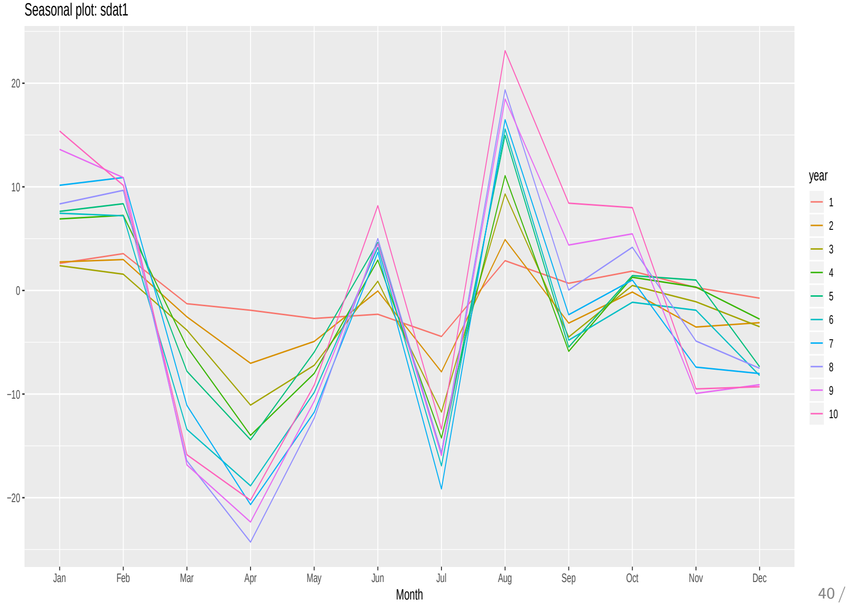
Multiplicative Seasonal ARIMA

- *E.g.* ARIMA(1,0,0)$\times$(1,1,0)$_{12}$ with $\phi = 0.5 = \Phi, s = 12$.
- CombMSC package: `sarima.Sim(n=10, period=12, model = list(order = c(1,0,0), ar=0.5),seasonal=list(order= c(1,1,0), ar = 0.8))`



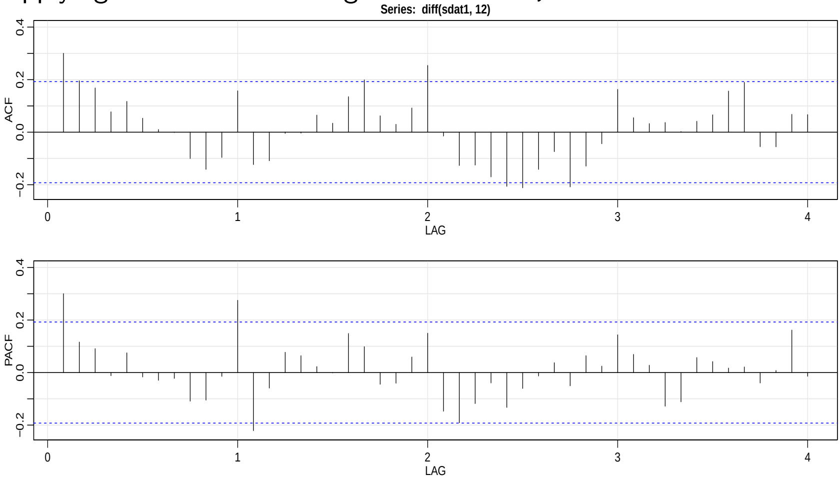
Multiplicative Seasonal ARIMA

- Seasonal plot: Data plotted against the seasons in separate years.



Multiplicative Seasonal ARIMA

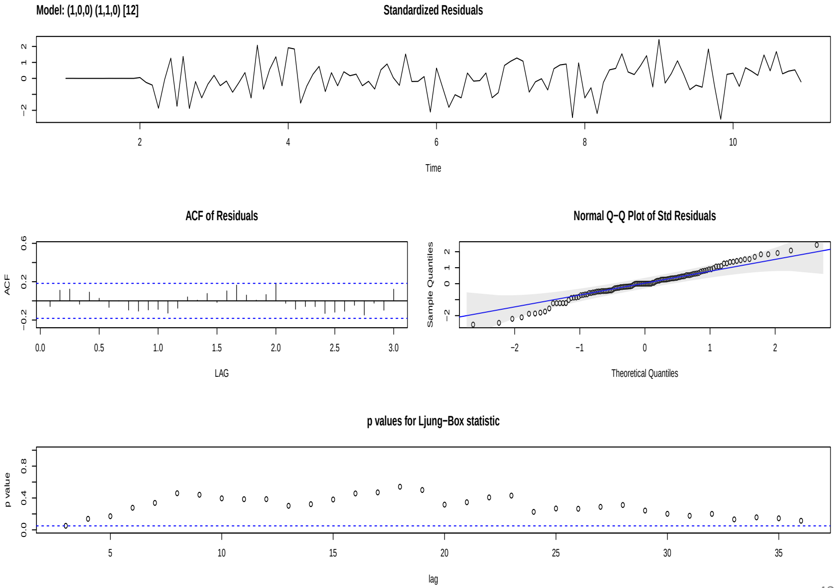
- Applying seasonal diferencing: `diff(sdat1,12)`



- *Seasonal*: ACF decays slowly at multiples of s. PACF cuts off after lag $1s = 1 \times 12$.
- *Non-seasonal*: ACF decays; PACF cuts off after lag 1.

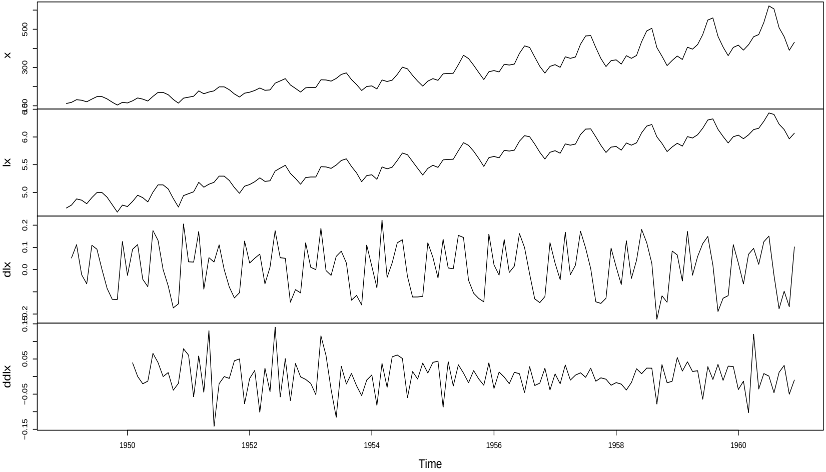
Multiplicative Seasonal ARIMA

- Start with `sarima(sdat1,1,0,0,1,1,0,12)`.

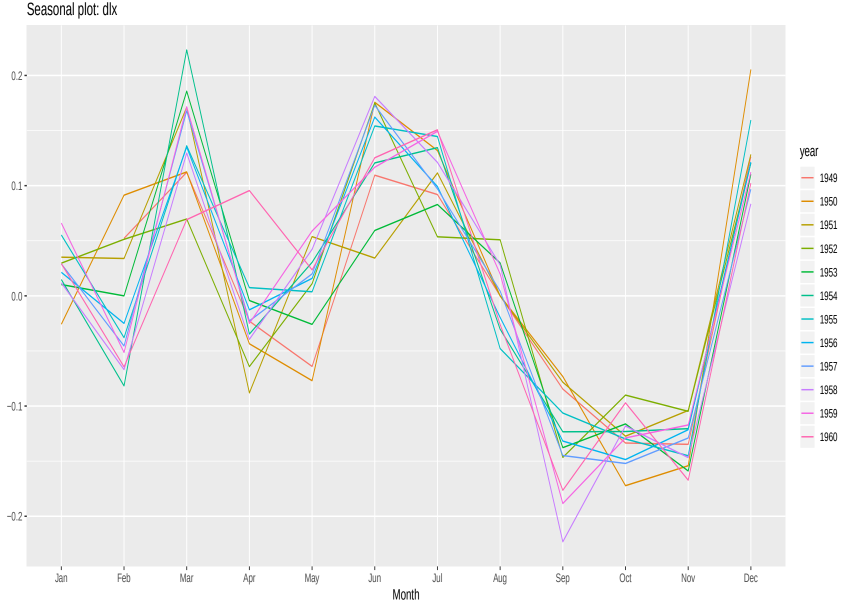


Multiplicative Seasonal ARIMA

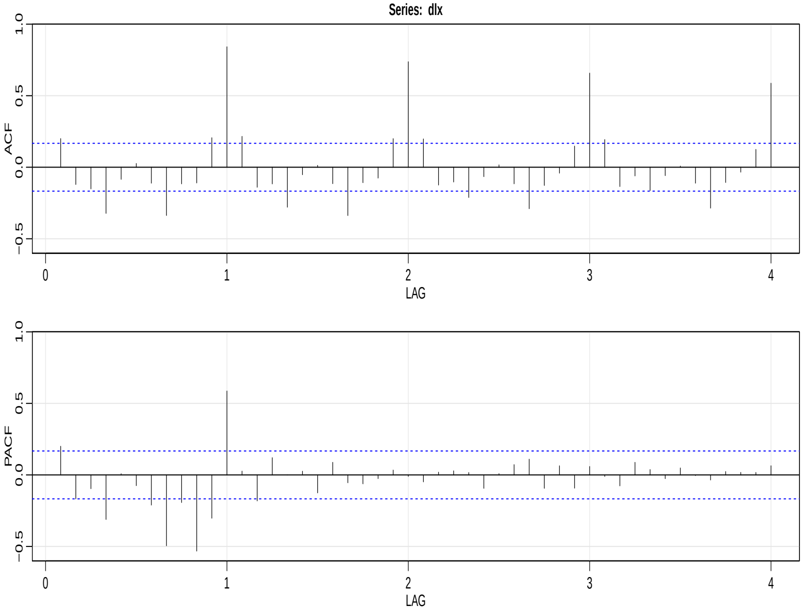
- **Air Passengers**: Monthly totals of international airline passengers from 1949-1960. Below are the original data x_t and the transformed data:lx=$\log(x_t)$,dlx=$\nabla \log x_t$,ddlx=$\nabla_{12} \nabla \log x_t$.



Multiplicative Seasonal ARIMA

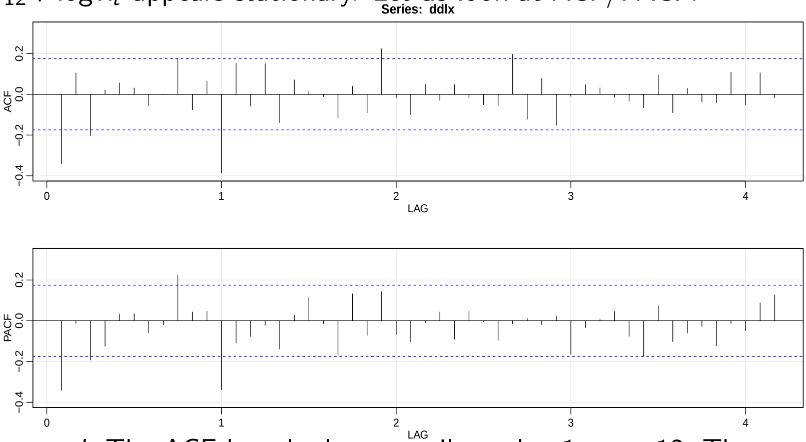


Multiplicative Seasonal ARIMA



Multiplicative Seasonal ARIMA

- $\nabla_{12}\nabla \log x_t$ appears stationary. Let us look at ACF/PACF.



- *Seasonal:* The ACF has the largest spike at lag $1s, s = 12$. The PACF is tailing off with persistent spikes at lag $ks, k = 1, 2, \ldots$ $\implies SMA(1), P = 0, Q = 1$.
- *Non-seasonal:* At lower lags, both are tailing off suggesting an ARMA(1,1).

Multiplicative Seasonal ARIMA

- Thus we can start with `Model 1`: ARIMA(1,1,1)$\times(0,1,1)_{12}$ on the log-transformed data. Using R, we obtain

```
> sarima(lx, 1,1,1, 0,1,1, 12)   # model 1
$ttable
     Estimate     SE t.value p.value
ar1    0.1960 0.2475  0.7921  0.4296
ma1   -0.5784 0.2132 -2.7127  0.0075
sma1  -0.5643 0.0747 -7.5544  0.0000
```

- The AR part is not significant so we can drop it and consider `Model 2`: ARIMA(0,1,1)$\times(0,1,1)_{12}$.

```
> sarima(lx, 0,1,1, 0,1,1, 12)
$ttable
     Estimate     SE t.value p.value
ma1   -0.4018 0.0896 -4.4825       0
sma1  -0.5569 0.0731 -7.6190       0
```

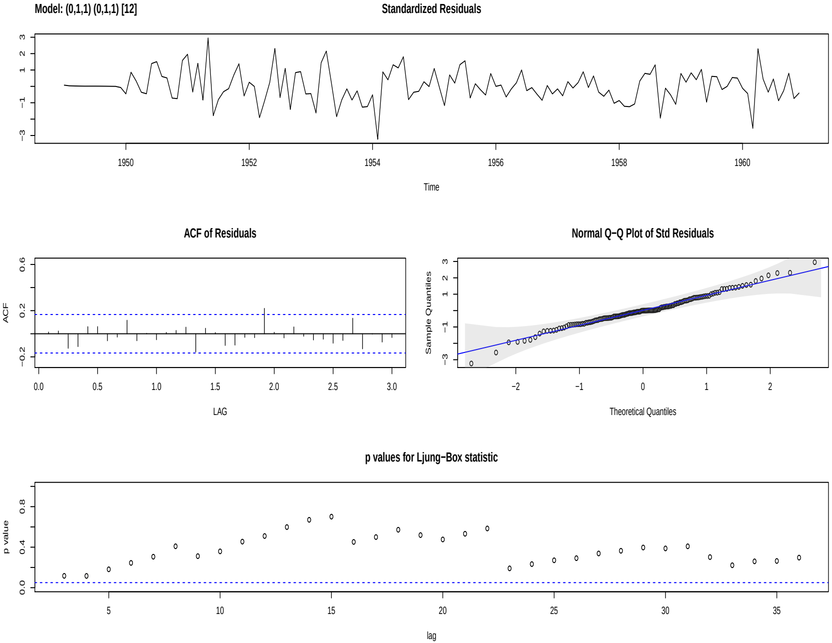
Multiplicative Seasonal ARIMA

- We can also try `Model 3`: ARIMA$(1, 1, 0) \times (0, 1, 1)_{12}$.

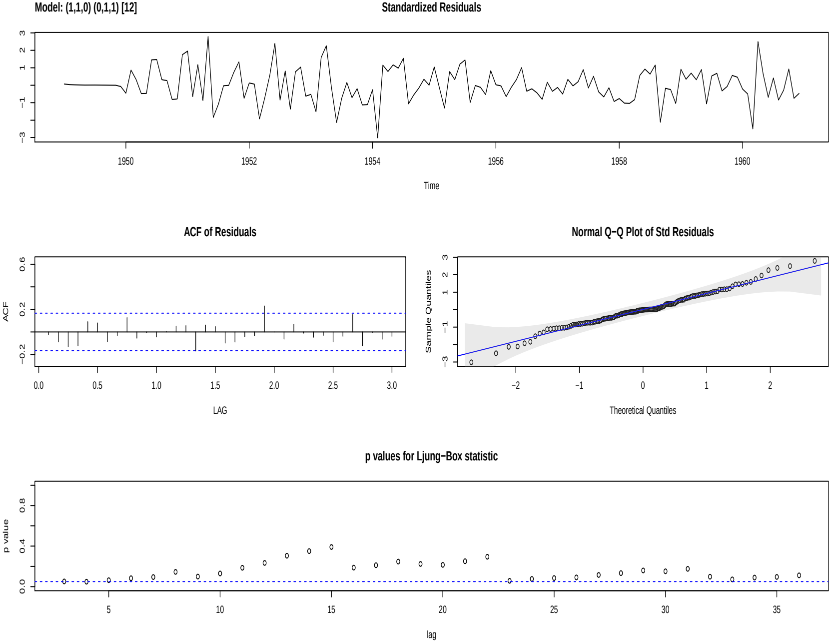
```
>sarima(lx, 1,1,0, 0,1,1, 12)
$ttable
     Estimate     SE t.value p.value
ar1   -0.3395 0.0822 -4.1295   1e-04
sma1  -0.5619 0.0748 -7.5109   0e+00
```

- Let us look at diagnostics.

Model 2: ARIMA $(0,1,1) \times (0,1,1)_{12}$



Model 3: ARIMA $(1,1,0) \times (0,1,1)_{12}$



More model comparisons

- More for Model 2:

```
> sarima(lx, 0,1,1, 0,1,1, 12)
$AIC
[1] -5.58133

$AICc
[1] -5.56625

$BIC
[1] -6.540082
```

- More for Model 3:

```
>sarima(lx, 1,1,0, 0,1,1, 12)
$AIC
[1] -5.567081

$AICc
[1] -5.552002

$BIC
[1] -6.525834
```

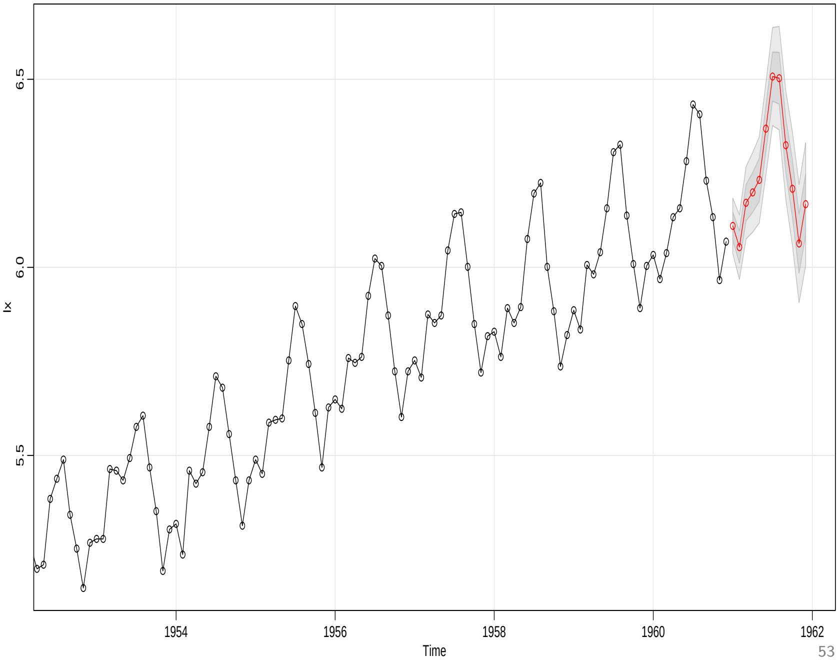
- And the winner is?
- Compare with `auto.arima()`.

Model 2 forecasts

- Forecast in the next 12 months:

```
> sarima.for(lx, 12, 0,1,1, 0,1,1,12)  # forecasts
$pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
1961 6.110186 6.053775 6.171715 6.199300 6.232556 6.368779 6.507294 6.502906
          Sep      Oct      Nov      Dec
1961 6.324698 6.209008 6.063487 6.168025

$se
            Jan        Feb        Mar        Apr        May        Jun
1961 0.03671562 0.04278290 0.04809071 0.05286829 0.05724854 0.06131668
            Jul        Aug        Sep        Oct        Nov        Dec
1961 0.06513121 0.06873437 0.07215784 0.07542608 0.07855847 0.08157066
```



6.5
6.0
5.5
lx
1954
1956
1958
1960
1962
Time

Lab for Today

- Fit a seasonal ARIMA model of your choice to the chicken price data in `chicken`. Check the fit and use the estimated model to forecast the next 12 months.